

Probabilistic Graphical Model Structure Learning: Application to Multi-Label Classification

Maxime Gasse

▶ **To cite this version:**

Maxime Gasse. Probabilistic Graphical Model Structure Learning: Application to Multi-Label Classification. Artificial Intelligence [cs.AI]. Université Lyon 1 - Claude Bernard, 2017. English. NNT : 2017LYSE1003 . tel-01442613v2

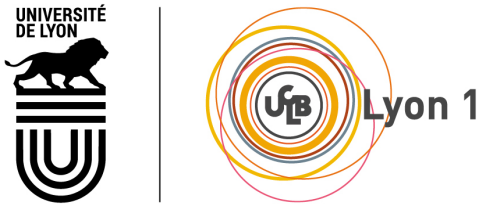
HAL Id: tel-01442613

https://hal.science/tel-01442613v2

Submitted on 1 Feb 2017 (v2), last revised 28 Aug 2017 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N° d'ordre NNT : 2017LYSE1003

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
l'Université Claude Bernard Lyon 1

École Doctorale ED512
Infomath

Spécialité de doctorat : Informatique

Soutenue publiquement le 13 janvier 2017, par :
Maxime Gasse

Apprentissage de Structure de Modèles Graphiques Probabilistes: Application à la Classification Multi-Label

---

Devant le jury composé de :

Céline Robardet, Professeur, INSA Lyon	Présidente
Christophe Gonzales, Professeur, Université Paris 6	Rapporteur
Jose M. Peña, Associate Professor, Linköping University	Rapporteur
Elisa Fromont, Maître de Conférence, Université Jean Monnet	Examinatrice
Willem Waegeman, Professor, Ghent University	Examinateur
Veronique Delcroix, Maître de Conférence, Université de Valenciennes	Examinatrice
Alexandre Aussem, Professeur, Université Lyon 1	Directeur de thèse
Haytham Elghazel, Maître de Conférence, Polytech Lyon	Co-directeur de thèse

UNIVERSITÉ CLAUDE BERNARD - LYON 1

Président de l'Université	**M. le Professeur Frédéric FLEURY**
Président du Conseil Académique	M. le Professeur Hamda BEN HADID
Vice-président du Conseil d'Administration	M. le Professeur Didier REVEL
Vice-président du Conseil Formation et Vie Universitaire	M. le Professeur Philippe CHEVALIER
Vice-président de la Commission Recherche	M. Fabrice VALLÉE
Directrice Générale des Services	Mme Dominique MARCHAND

COMPOSANTES SANTÉ

Faculté de Médecine Lyon Est - Claude Bernard	Directeur : M. le Professeur G.RODE
Faculté de Médecine et de Maïeutique Lyon Sud - Charles Mérieux	Directeur : Mme la Professeure C. BURILLON
Faculté d'Odontologie	Directeur : M. le Professeur D. BOURGEOIS
Institut des Sciences Pharmaceutiques et Biologiques	Directeur : Mme la Professeure C. VINCIGUERRA
Institut des Sciences et Techniques de la Réadaptation	Directeur : M. X. PERROT
Département de formation et Centre de Recherche en Biologie Humaine	Directeur : Mme la Professeure A-M. SCHOTT

COMPOSANTES ET DÉPARTEMENTS DE SCIENCES ET TECHNOLOGIE

Faculté des Sciences et Technologies	Directeur : M. F. DE MARCHI
Département Biologie	Directeur : M. le Professeur F. THEVENARD
Département Chimie Biochimie	Directeur : Mme C. FELIX
Département GEP	Directeur : M. Hassan HAMMOURI
Département Informatique	Directeur : M. le Professeur S. AKKOUCHE
Département Mathématiques	Directeur : M. le Professeur G. TOMANOV
Département Mécanique	Directeur : M. le Professeur H. BEN HADID
Département Physique	Directeur : M. le Professeur J-C. PLENET
UFR Sciences et Techniques des Activités Physiques et Sportives	Directeur : M. Y.VANPOULLE
Observatoire des Sciences de l'Université de Lyon	Directeur : M. B. GUIDERDONI
Polytech Lyon	Directeur : M. le Professeur E.PERRIN
Ecole Supérieure de Chimie Physique Electronique	Directeur : M. G. PIGNAULT
Institut Universitaire de Technologie de Lyon 1	Directeur : M. le Professeur C. VITON
Ecole Supérieure du Professorat et de l'Education	Directeur : M. le Professeur A. MOUGNIOTTE
Institut de Science Financière et d'Assurances	Directeur : M. N. LEBOISNE

Remerciements

En premier lieu, je tiens à exprimer toute ma gratitude aux membres du jury pour l'attention qu'ils ont portée à l'évaluation de mes travaux de thèse. En particulier, je voudrais remercier mes rapporteurs, le Professeur Jose M. Peña et le Professeur Christophe Gonzales, pour le travail considérable qu'ils ont consacré à la lecture assidue de ce long mémoire.

J'adresse de chaleureux remerciements envers mes encadrants, le Professeur Alexandre Aussem et le Docteur Haytham Elghazel, pour la confiance qu'ils m'ont accordée. Au delà des échanges scientifiques, ils ont sû me faire profiter de leur expérience précieuse du monde de la recherche, qui m'était jusqu'alors inconnu. Surtout, j'ai pu jouir tout au long de cette thèse d'une grande liberté intellectuelle pour laquelle je suis très reconnaissant.

Je remercie l'ensemble des personnels du laboratoire LIRIS et de l'université de Lyon, et en particulier le personnel administratif que j'ai pu côtoyer, Isabelle, Brigitte, Jean-Pierre, Catherine, pour son efficacité et sa bienveillance. Je remercie bien sûr mes collègues de bureau, malheureusement trop nombreux pour être cités ici, pour toutes ces discussions passionnantes le midi ou autour d'un café. Cette atmosphère chaleureuse et amicale, dans un environnement scientifique stimulant, ont largement contribué à mon épanouissement au cours de ces quatre années de recherches.

Il convient également de remercier l'union européenne ainsi que l'état français qui ont financé cette thèse, et m'ont permis de vivre décemment de mes activités de recherche durant quatre ans.

Je tiens à remercier mes proches, mes amis, ma famille, pour leur soutien ou simplement leur présence. Mener à bien une thèse de doctorat est une expérience passionnante, mais également laborieuse à bien des égards, ainsi on apprécie pouvoir se ressourcer ailleurs lors des passages à vide. Je remercie enfin Charlotte qui m'a supporté toutes ces années, et continue à le faire aujourd'hui.

Probabilistic Graphical Model Structure Learning: Application to Multi-Label Classification

---

Maxime Gasse, Ph.D thesis

Résumé

Dans cette thèse, nous nous intéressons au problème spécifique de l'apprentissage de structure de modèles graphiques probabilistes, c'est-à-dire trouver la structure la plus efficace pour représenter une distribution, à partir seulement d'un ensemble d'échantillons $\mathcal{D} \sim p(\mathbf{v})$. Dans une première partie, nous passons en revue les principaux modèles graphiques probabilistes de la littérature, des plus classiques (modèles dirigés, non-dirigés) aux plus avancés (modèles mixtes, cycliques etc.). Puis nous étudions particulièrement le problème d'apprentissage de structure de modèles dirigés (réseaux Bayésiens), et proposons une nouvelle méthode hybride pour l'apprentissage de structure, H2PC (*Hybrid Hybrid Parents and Children*), mêlant une approche à base de contraintes (tests statistiques d'indépendance) et une approche à base de score (probabilité postérieure de la structure).

Dans un second temps, nous étudions le problème de la classification multi-label, visant à prédire un ensemble de catégories (vecteur binaire $\mathbf{y} \in (0,1)^m$) pour un objet (vecteur $\mathbf{x} \in \mathbb{R}^d$). Dans ce contexte, l'utilisation de modèles graphiques probabilistes pour représenter la distribution conditionnelle des catégories prend tout son sens, particulièrement dans le but minimiser une fonction coût complexe. Nous passons en revue les principales approches utilisant un modèle graphique probabiliste pour la classification multi-label (*Probabilistic Classifier Chain*, *Conditional Dependency Network*, *Bayesian Network Classifier*, *Conditional Random Field*, *Sum-Product Network*), puis nous proposons une approche générique visant à identifier une factorisation de $p(\mathbf{y}|\mathbf{x})$ en distributions marginales disjointes, en s'inspirant des méthodes d'apprentissage de structure à base de contraintes. Nous démontrons plusieurs résultats théoriques, notamment l'unicité d'une décomposition minimale, ainsi que trois procédures quadratiques sous diverses hypothèses à propos de la distribution jointe $p(\mathbf{x}, \mathbf{y})$. Enfin, nous mettons en pratique ces résultats afin d'améliorer la classification multi-label avec les fonctions coût *F-loss* et *zero-one loss*.

Abstract

In this thesis, we address the specific problem of probabilistic graphical model structure learning, that is, finding the most efficient structure to represent a probability distribution, given only a sample set $\mathcal{D} \sim p(\mathbf{v})$. In the first part, we review the main families of probabilistic graphical models from the literature, from the most common (directed, undirected) to the most advanced ones (chained, mixed etc.). Then we study particularly the problem of learning the structure of directed graphs (Bayesian networks), and we propose a new hybrid structure learning method, H2PC (*Hybrid Hybrid Parents and Children*), which combines a constraint-based approach (statistical independence tests) with a score-based approach (posterior probability of the structure).

In the second part, we address the multi-label classification problem, which aims at assigning a set of categories (binary vector $\mathbf{y} \in (0,1)^m$) to a given object (vector $\mathbf{x} \in \mathbb{R}^d$). In this context, probabilistic graphical models provide convenient means of encoding $p(\mathbf{y}|\mathbf{x})$, particularly for the purpose of minimizing general loss functions. We review the main approaches based on PGMs for multi-label classification (*Probabilistic Classifier Chain*, *Conditional Dependency Network*, *Bayesian Network Classifier*, *Conditional Random Field*, *Sum-Product Network*), and propose a generic approach inspired from constraint-based structure learning methods to identify the unique partition of the label set into irreducible label factors (ILFs), that is, the irreducible factorization of $p(\mathbf{y}|\mathbf{x})$ into disjoint marginal distributions. We establish several theoretical results to characterize the ILFs based on the compositional graphoid axioms, and obtain three generic procedures under various assumptions about the conditional independence properties of the joint distribution $p(\mathbf{x}, \mathbf{y})$. Our conclusions are supported by carefully designed multi-label classification experiments, under the *F-loss* and the *zero-one loss* functions.

Contents

Introduction **1**

1 Background and notation **3**
1.1 Probability theory . . . 3
1.1.1 Probability spaces . . . 3
1.1.2 Random variables . . . 7
1.1.3 Independence models . . . 13
1.2 Graph theory . . . 16
1.2.1 Connectivity, paths, walks, cycles . . . 16
1.2.2 Classes of graphs . . . 17

2 Probabilistic graphical models **19**
2.1 Classical graphical models . . . 20
2.1.1 Undirected graphs . . . 20
2.1.2 Directed acyclic graphs . . . 27
2.2 Advanced graphical models . . . 41
2.2.1 Bi-directed graphs . . . 42
2.2.2 Chain graphs . . . 43
2.2.3 Mixed graphs . . . 54
2.3 Discussion . . . 62
2.3.1 Trends . . . 63
2.3.2 Limitations . . . 65

3 Bayesian network structure learning **71**
3.1 Motivation . . . 71
3.2 The score-based approach . . . 73
3.2.1 Bayesian scores . . . 74
3.2.2 Information-theoretic scores . . . 77
3.2.3 The optimization problem . . . 81
3.2.4 Meek's conjecture . . . 81
3.3 The constraint-based approach . . . 82
3.3.1 Conditional independence tests . . . 83
3.3.2 The faithfulness assumption . . . 84
3.3.3 Algorithms . . . 84

3.4 The hybrid approach . . . 90
3.4.1 Early works . . . 92
3.4.2 Max-Min Hill-Climbing (MMHC) . . . 92
3.5 Our contribution: a new hybrid algorithm . . . 93
3.5.1 The ideal skeleton . . . 94
3.5.2 Hybrid Hybrid Parents and Children (H2PC) . . . 94
3.5.3 Experimental validation . . . 95
3.5.4 Discussion . . . 100

4 Multi-label classification **103**
4.1 Supervised learning . . . 104
4.1.1 Risk minimization . . . 104
4.1.2 Multi-label loss functions . . . 107
4.1.3 Illustration . . . 111
4.2 Meta-learning approaches . . . 118
4.2.1 Binary Relevance . . . 118
4.2.2 Label Powerset . . . 119
4.2.3 Chaining . . . 120
4.2.4 Stacking . . . 121
4.2.5 Ensemble learning . . . 122
4.2.6 Discussion . . . 124
4.3 Plug-in approaches . . . 125
4.3.1 Probabilistic classifier chains . . . 126
4.3.2 Conditional dependency networks . . . 127
4.3.3 Bayesian network classifiers . . . 128
4.3.4 Conditional Random Fields . . . 129
4.3.5 Sum product networks . . . 132
4.3.6 Discussion . . . 138

5 Irreducible label factors **141**
5.1 Characterizations . . . 142
5.1.1 Preliminary materials . . . 142
5.1.2 Irreducible label factors . . . 144
5.1.3 Minimal feature subsets . . . 151
5.1.4 Algorithms . . . 153
5.2 Application to subset zero-one loss minimization . . . 160
5.2.1 Factorized LP . . . 160
5.2.2 Toy problem . . . 161
5.2.3 Real-world benchmark . . . 164
5.3 Application to F-measure maximization . . . 171
5.3.1 Factorized GFM . . . 172
5.3.2 Toy problem . . . 178

5.3.3 Real-world benchmark 181
5.4 Discussion 182

Conclusion and perspectives **185**

A Supplementary material **189**
A.1 Decomposition graphs 189
A.2 Additional benchmark measures 192

B Proofs **195**

Bibliography **205**

Author's publications **223**

Introduction

A probabilistic graphical model (PGM) allows for the compact representation of a multivariate probability distribution $p(\mathbf{v})$ by exploiting the independence structure between the variables, encoded in the form of a graph.

The first part of this thesis is dedicated to the problem of PGM structure learning, with a comprehensive review of the main PGM families present in the literature, and a narrow focus on the Bayesian network structure learning problem. This part culminates with a new hybrid algorithm for BN structure learning, the so-called *Hybrid Hybrid Parents and Children* (H2PC) algorithm, [GAE12; GAE14].

The second part of this thesis is dedicated to the problem multi-label classification (MLC), a natural application for probabilistic graphical models. We will discuss the main challenges of MLC, and review the main approaches proposed so far. Finally, we will present our generic approach based on the concept of *Irreducible Label Factors* (ILFs), accompanied by a series of theoretical and empirical results [GAE15; GA16a; GA16b].

This manuscript is intended to be self-contained, therefore advised readers may skip the preliminary materials in Chapter 1. A confident reader may also skip Chapters 2 and 4, which respectively present a comprehensive review of PGM families and MLC approaches. Personal contributions are contained within Chapters 3 and 5.

In Chapter 1 we introduce some basic preliminary concepts from probability theory and graph theory, such as independence relations, independence models, independence properties, and basic graph properties.

In Chapter 2 we present a comprehensive review of the main PGM families present in the literature, from the most common (directed, undirected) to the most advanced ones (chained, mixed etc.), with some discussions about their inherent limitations.

In Chapter 3 we introduce the specific problem of Bayesian network structure learning, review the main approaches proposed so far (score-based, constraint-based, hybrid), and present a new hybrid approach, MMHC, which improves over the state-of-the-art H2PC algorithm.

In Chapter 4 we present a comprehensive review of the MLC problem, its combinatorial challenges, and the main PGM approaches discussed so far in the literature (*Probabilistic Classifier Chain*, *Conditional Dependency Network*, *Bayesian Network Classifier*, *Conditional Random Field*, *Sum-Product Network*).

In Chapter 5 we propose a generic approach to help solving the MLC problem efficiently under any loss function, based on the concept of *Irreducible Label Factors* (ILFs). We present a series of theoretical results to identify the ILF partition of a multivariate conditional distribution, $p(\mathbf{y}|\mathbf{x})$, by adopting a constraint-based structure learning approach. Finally we demonstrate empirically the usefulness of our approach for multi-label classification under the subset zero-one loss and the F-loss functions.

1 Background and notation

> *Probability theory is nothing but common sense reduced to calculation.*
>
> — **Pierre-Simon Laplace**
> 1812

In this chapter we will introduce some important concepts and notations from probability theory and graph theory, which will be heavily used in the remainder of this work. Most of the material presented here can be found in the very good book from Koller and Friedman [KF09], which covers many aspects in much more detail. Another very accurate resource on probabilistic independence models is the book from Studeny [Stu05].

1.1 Probability theory

The probability of an event is intuitively defined as: how much do I believe this event will happen? Such a question is very subjective, as it implicitly refers to the future, to the unknown, and to our degree of uncertainty about the world. To give a more formal definition of the notion of probability, we will introduce a mathematical framework called probability theory.

1.1.1 Probability spaces

universe

To reason about uncertainty, we introduce first the set of all elementary events, also called the state space or the universe, denoted Ω. The universe may include all the possible states of the world, which would allow us to reason about everything. However, in that case Ω would be infinitely bulky and tedious to handle. To make reasoning easier, we usually restrict ourselves to a smaller system of interest. Suppose our state space consists in the possible outcomes of a six-sided die, then the universe

event

is $\Omega = \{1, 2, 3, 4, 5, 6\}$. An event E is a set of states, including the empty event $\emptyset$, the elementary events which correspond to a unique state, or any combination of them. Equivalently, we say that an event is a sub-space of the universe, i.e. $E \subseteq \Omega$. The set

event space

of all possible events is then called the event space, denoted S.

Probability theory requires that the event space satisfies three basic properties:

- S contains the empty event $\emptyset$, and the trivial event Ω.
- S is closed under union. That is, if $\alpha, \beta \in S$, then so is $\alpha \cup \beta$.
- S is closed under complementation. That is, if $\alpha \in S$, then so is $\Omega \setminus \alpha$.

The requirement that the event space is closed under union ($\cup$) and complementation implies that it is also closed under other Boolean operations, such as intersection ($\cap$) and set difference ($\setminus$).

We can now introduce P, a function defined over S which assigns to each event a certain value, the probability of that event. This value $P(E)$ gives a quantified answer to the question: how much is it plausible that, if I observe the universe, it will be in one of the states in E? By definition, P is a positive real-valued function, normalized over Ω.

probability distribution

Def. 1.1 *A probability distribution P defined over (Ω, S) is a mapping from events S to real values that satisfies the following conditions:*

1. *The probability of an event is positive: $P(E) \geq 0, \forall E \in S$.*
2. *The probability of the whole universe is one: $P(\Omega) = 1$.*
3. *Any pair of disjoint events α and β satisfies $P(\alpha \cup \beta) = P(\alpha) + P(\beta)$.*

The third axiom, also known as the additive rule, states that the probability that one of two mutually disjoint events will occur is the sum of the probabilities of each event. By definition, the elementary events (denoted e) are mutually exclusive, and the probability of any event E decomposes as

$$P(E) = \sum_{e \in E} P(e), \quad \forall E \in S.$$

The second and third axioms have many others implications. Of particular interest are $P(\emptyset) = 0$ (where $\emptyset$ denotes the empty set of states, or the null event), and $P(\alpha \cup \beta) = P(\alpha) + P(\beta) - P(\alpha \cap \beta)$.

Take again the example of a six-sided die. With a balanced die, the probability of obtaining an odd outcome after a roll is given by

$$P(\{1, 3, 5\}) = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{1}{2}.$$

Conditional probability

A conditional probability is the probability of an event, given some evidence. Conditional probabilities arise naturally when we reason about the world, especially if we have to make a decision. We usually want to take into account as many information we have, to adjust our belief in what will happen in the future and ensure we make the less risky choice. Suppose you are playing poker, then typically you want to know the probability of your opponent having a strong hand, for example a pair of Aces. If there is an Ace revealed on the table, then it is quite likely that he has one of the three remaining Aces in his hand. However, if you also have an Ace in your hand, then only two Aces are remaining, and it is less likely that one of them is in your opponent's hand. Of course, a good poker player will use a lot more information to adjust the probability of the different hands his opponent may have, such as his behaviour, the look on his face, and so on. In the end, a good part of playing poker resides in computing conditional probabilities, although in an informal (or unconscious) way. In a formal probabilistic framework, conditional probability is defined as follows

Def. 1.2 *conditional probability* *The conditional probability of an event α given an event β is:*

$$P(\alpha|\beta) = \frac{P(\alpha \cap \beta)}{P(\beta)}.$$

When $P(\beta) = 0$, $P(\alpha|\beta)$ is not defined[1].

If we go back to our die example, we may be interested in the probability of obtaining a 3, given that I already known that the outcome of the roll is an odd number (suppose someone looked at it and told you so). This conditional probability is given by

$$P(\{3\}|\{1,3,5\}) = \frac{1/6}{1/2} = \frac{1}{3}.$$

Chain rule and Bayes' rule

chain rule

From the definition of conditional probability, we immediately have that $P(\alpha \cap \beta) = P(\beta)P(\alpha|\beta)$. By extension we obtain the so-called chain rule of conditional probabilities. That is, for any number of events $\alpha_1, \ldots, \alpha_n$ we can write

$$P(\alpha_1 \cap \ldots \cap \alpha_n) = P(\alpha_1)P(\alpha_2|\alpha_1)\ldots P(\alpha_n|\alpha_1 \cap \ldots \cap \alpha_{n-1}).$$

[1] Note that conditional independence when conditioning on events of probability zero is possible within the context of full conditional probabilities. While not discussed here, interested readers are pointed to Cozman [Coz13].

In other words, the probability of a combination of events can be expressed in terms of the probability of the first, the probability of the second given the first, and so on. It is important to notice that this holds for any ordering of the events.

Bayes' rule

Another immediate consequence of the conditional probability definition is the following, known as Bayes' rule:

$$P(\alpha|\beta) = \frac{P(\alpha)P(\beta|\alpha)}{P(\beta)}.$$

Bayes' rule is important in that it allows us to compute the conditional probability $P(\alpha|\beta)$ from the "inverse" conditional probability $P(\beta|\alpha)$. A lot of approaches to statistics are derived from this simple rule, and form the so-called field of Bayesian statistics. The term Bayesian is often used in opposition to the frequentist statistics, which refer to a somewhat more classical view of statistics. However, the difference between the two approaches is merely philosophical, and the practical methods employed in both fields often end up doing quite the same thing [GM91; GS98].

Independence between events

The notions of independence and conditional independence constitute the cornerstone of probabilistic graphical models. It is essential to clearly understand them before to dig into such models. As previously mentioned, adjusting the probability of an event, $P(\alpha)$, by taking into account the fact that another event is true, i.e. $P(\alpha|\beta)$, is crucial for decision making. However, it may be that the event β does not change the probability of α, in which case we say that α is independent of β.

Def. 1.3 *indep. of events*

Two events α and β are independent, denoted $\alpha \perp\!\!\!\perp \beta$, when the following holds:

$$P(\alpha \cap \beta) = P(\alpha)P(\beta).$$

It follows immediately from the definition that both the null event $\emptyset$ and the sure event Ω are independent of any event E. An alternative definition is $P(\alpha) = P(\alpha|\beta)$ when $P(\beta) > 0$.

Note that two disjoint events with non-zero probability are always dependent. The converse, however, is not true. For example, if we go back to our single die example, the probability of obtaining an odd number does not change if we know that that number is at most equal to 4, i.e. $P(\{1,3\}) = P(\{1,3,5\})P(\{1,2,3,4\})$. Or, equivalently $P(\{1,3,5\}) = P(\{1,3,5\}|\{1,2,3,4\})$.

The definition of conditional independence is strictly the same, except for the addition of an evidence term, that is, $P(\alpha \cap \beta|\gamma) = P(\alpha|\gamma)P(\beta|\gamma)$. However, such a definition is problematic when $P(\gamma) = 0$, as in that case $P(\alpha|\gamma)$ is not defined. We follow Waal [Waa09] who uses a more general definition, which is strictly equivalent when $P(\gamma) > 0$.

conditional indep. of events

Def. 1.4 *Two events α and β are conditionally independent given a third event γ, denoted $\alpha \perp\!\!\!\perp \beta \mid \gamma$, when the following holds:*

$$P(\alpha \cap \beta \cap \gamma)P(\gamma) = P(\alpha \cap \gamma)P(\beta \cap \gamma).$$

Again, an alternative definition is $P(\alpha|\gamma) = P(\alpha|\beta \cap \gamma)$.

It is important to understand that conditional independence does not imply independence, nor independence does imply conditional independence. This will become clear in the next section with Example 1.1.

1.1.2 Random variables

We may now introduce the concept of random variable. In our die example, we may assign a random variable to the outcome of the die, denoted with a capital letter X. The domain of the random variable, i.e. the specific set of all the possible outcomes, is then denoted by a calligraphic letter $\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$. The outcome of a random variable is denoted with a lowercase letter, i.e. $x \in \mathcal{X}$. Because here X takes values within a finite set, we say that it is a discrete random variable. In that case we can introduce a probability mass function, $p_X(x)$, which maps every value of X to the probability of the event $X = x$, i.e. X takes the particular value x:

prob. mass function

$$p_X(x) = P(X = x).$$

Each possible value for X being mutually exclusive, we can consider each event $X = x$ as an elementary event of the universe $\Omega = \mathcal{X}$. Then, from the third axiom we can compute the probability of any event $E \subseteq \Omega$ by summing up over $p_X(x)$:

$$P(E) = \sum_{x \in E} p_X(x), \quad \forall E \subseteq \mathcal{X}.$$

For the sake of simplicity, most of the time we will consider discrete random variables. However, all the properties which we describe hereinafter apply to continuous random variables as well, by replacing the summation $\sum$ with an integration $\int$.

Suppose that X corresponds to the lifetime of an electric bulb, then it can take any positive real value, i.e. $\mathcal{X} = \mathbb{R}_{\geq 0}$. The set of all possible events being infinite, the probability of a particular elementary event is necessarily $P(X = x) = 0$. It makes no sense to consider a probability mass function in that case. Instead, we usually employ a probability density function, $p_X(x)$, which is a non-negative Lebesgue-integrable function corresponding to:

prob. density function

$$p_X(x) = \frac{\mathrm{d}P(X \leq x)}{\mathrm{d}x}.$$

Because of that property, we can use the third axiom to compute the probability of any event E by summing up (integrating) over $p_X(x)$:

$$P(E) = \int_{x \in E} p_X(x)\mathrm{d}x, \quad \forall E \subseteq \mathcal{X}.$$

Note that, because a density function corresponds to a derivative, in the continuous case it is very likely that $p_X(x)$ takes values greater than 1.

In the end, any probability mass or density function $p_X(x)$ defined over the state space $\mathcal{X}$ characterizes a probability distribution P defined over the corresponding event space. Therefore, in the remainder of this work we allow ourselves to use the term probability distribution, or simply distribution, when we refer to a mass or a density function.

Multivariate random variables

Suppose now that our universe is the outcome of two dice, then it can take 6×6 different states, denoted $\Omega = \{(1,1),(1,2),\ldots,(6,5),(6,6)\}$. It makes sense to consider the universe as a space in two dimensions, each one corresponding to the outcome of one of the dice. In that case we will use two random variables X and Y to represent the value of each die, and we will introduce a multivariate random variable denoted by a bold capital letter, $\mathbf{U} = \{X, Y\}$. Note that here we employed the letter $\mathbf{U}$ for universe, because our multivariate random variable characterizes the whole state space, $\Omega = \mathcal{X} \times \mathcal{Y}$. An elementary event is then denoted by a bold lowercase letter, $\mathbf{u} = (x, y)$, which is a vector in the state space of the random variable, a.k.a. a sample point.

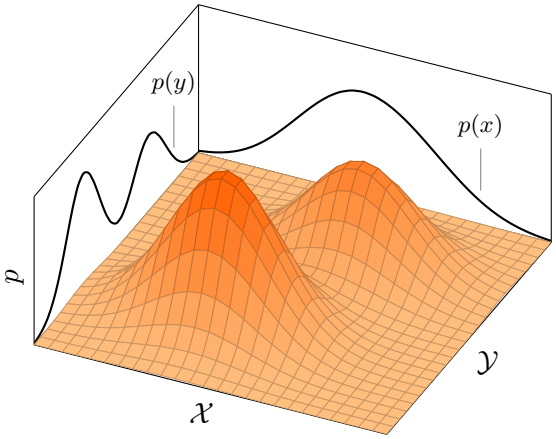


Fig. 1.1. Marginalization illustration.

Joint and marginal distributions

A joint probability distribution, like $p_{XY}(x, y)$, denotes a probability distribution defined over a multi-dimensional space. A marginal distribution, like $p_X(x)$, denotes a probability distribution defined over a reduced number of dimensions, relatively to a joint distribution. To derive a marginal distribution from a joint distribution, we employ the marginalization rule:

marginalization rule

$$p_X(x) = \sum_{y \in \mathcal{Y}} p_{XY}(x, y).$$

When the domain of a probability distribution is clear from the context, we will omit the under-script to gain in clarity, as in $p(x)$. Another shorthand in notation is that $\sum_x$ refers to $\sum_{x \in \mathcal{X}}$, a sum over all possible values that X can take. For example, the marginalization rule becomes: $p(x) = \sum_y p(x, y)$.

Figure 1.1 provides a visual illustration of marginalization in a two-dimensional space. In this example X and Y are two continuous random variables, and their joint distribution $p(x, y)$ is defined as a mixture of two multivariate Gaussian distributions. To marginalize out some random variables (say X), we project the distribution defined over a multi-dimensional space ($\mathcal{X} \times \mathcal{Y}$) onto a lower-dimensional space ($\mathcal{X}$), by summing up over the remaining dimensions ($\mathcal{Y}$).

Conditional distributions

By extension to conditional probabilities, a conditional distribution is a probability distribution re-defined over a sub-space of the universe, where the evidence is

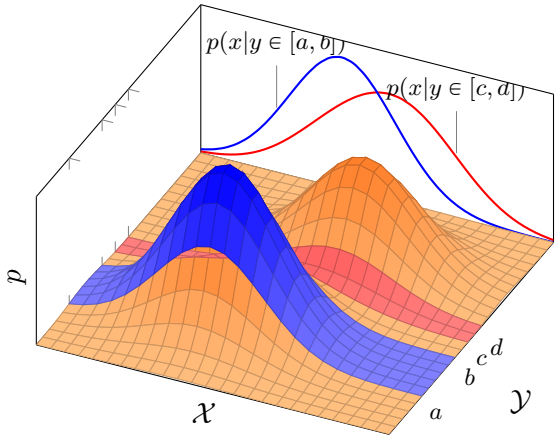


Fig. 1.2. Conditioning illustration.

true. Any conditional distribution can be derived from a marginal and a joint distribution:

$$p(x|y) = \frac{p(x,y)}{p(y)}.$$

Figure 1.2 provides a visual illustration of conditional distributions in a two-dimensional space. It consists in restricting the domain of p to a particular space where the evidence is true, such as the blue area where $y \in [a,b]$, and re-normalizing it accordingly. Then, a proper marginalization can be applied to obtain conditional marginal distributions, like $p(x|y \in [a,b])$.

Independence between random variables

The notion of independence applies to random variables as well. We give right away the definition of conditional independence for multivariate random variables[2].

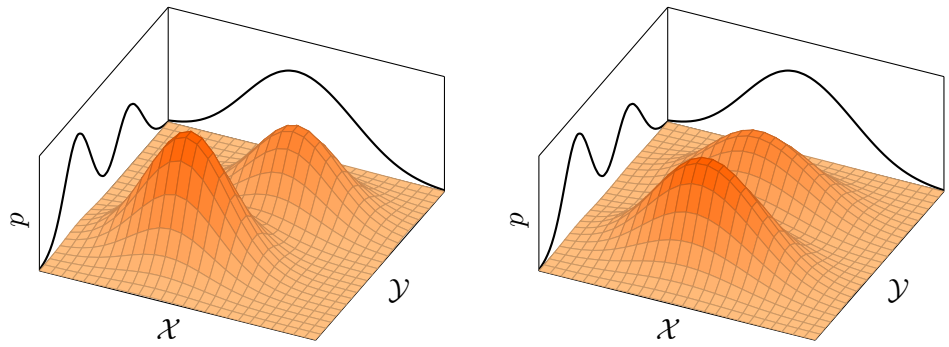
conditional indep.

Def. 1.5 *Two random variables* $\mathbf{X}$ *and* $\mathbf{Y}$ *are independent given a third random variable* $\mathbf{Z}$*, denoted* $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$*, when the following holds for all* $(\mathbf{x},\mathbf{y},\mathbf{z}) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$*:*

$$p(\mathbf{x},\mathbf{y},\mathbf{z})p(\mathbf{z}) = p(\mathbf{x},\mathbf{z})p(\mathbf{y},\mathbf{z}).$$

This definition includes univariate random variables as a particular case, for example $\mathbf{X} = \{X\}$. In that case we may write $X \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$ as a shorthand for $\{X\} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$ to alleviate our notation.

[2]Note that most definitions from the literature use the condition $p(\mathbf{x},\mathbf{y}|\mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{y}|\mathbf{z})$ or $p(\mathbf{x}|\mathbf{y},\mathbf{z}) = p(\mathbf{x}|\mathbf{z})$, however we prefer the definition below which does not rely on any positivity condition, i.e. $p(\mathbf{z}) > 0$ or $p(\mathbf{y},\mathbf{z}) > 0$.



(a) X and Y are dependent. **(b)** X and Y are independent.

Fig. 1.3. Independence between random variables in a two-dimensional space. The joint distributions in (a) and (b) share the same marginal distributions, however only the distribution in (b) factorizes according to its marginals, i.e. $p(x, y) = p(x)p(y)$.

Another random variable of interest is the empty set of variables. By definition, the empty random variable $\mathbf{X} = \emptyset$ (not to be confused with the empty event $E = \emptyset$) is constant, which implies that $P(\mathbf{X} = \mathbf{x}) = 1$ for every $\mathbf{x} \in \mathcal{X}$. If we set $\mathbf{Z} = \emptyset$ the evidence term above vanishes and we obtain the definition of unconditional independence, i.e. $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \iff \mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \emptyset$. Another implication, called the trivial independence, is that $\mathbf{X} \perp\!\!\!\perp \emptyset \mid \mathbf{Z}$ is always true.

trivial indep.

Figure 1.3 provides a visual illustration of independence between two continuous random variables.

Ex. 1.1 *car parking example*

Suppose that we observe a car parking, and that for each car we have access to three information: the fuel level indicator, the battery level indicator, and whether the engine starts when we turn the ignition key. Let us define three binary random variables: X which equals 0 if the car is out of fuel, 1 if not; Y which equals 0 if the car is out of battery, 1 if not; and Z which equals 1 if the engine starts, 0 if not. In this example we will consider the full joint probability distribution given in Table 1.1. The fuel level of a car does not give any information about its battery level, as shown in Table 1.2, and we have that $X \perp\!\!\!\perp Y$. However, if we know whether the engine starts or not, the battery level can give an information about the fuel level, and we have $X \not\perp\!\!\!\perp Y \mid Z$. For example if the engine does not start when we turn the ignition key, but we have the information that the car is not out of battery, then the car must probably be out of fuel. This is shown in Table 1.3.

We may now give an alternative definition of conditional independence between random variables, which is mathematically more convenient.

Thm. 1.1 *$\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$ if and only if there exists two functions f and g such that*

$$p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{x}, \mathbf{z})g(\mathbf{y}, \mathbf{z}).$$

Tab. 1.1. The joint probability distribution of X, Y and Z in our car parking example, i.e. $p(x, y, z)$.

Z (ignition)	X (fuel)	Y (battery) 0	1
0	0	.06	.23
	1	.13	.06
1	0	.00	.01
	1	.01	.50

Tab. 1.2. The joint and marginal probability distributions of X and Y in our car parking example, i.e. $p(x, y)$, $p(x)$ and $p(y)$.

		Y (battery) 0	1	
X (fuel)	0	.06	.24	.30
	1	.14	.56	.70
		.20	.80	

Tab. 1.3. The joint and marginal conditional probability distributions of X and Y given Z in our car parking example, i.e. $p(x, y|z)$, $p(x|z)$ and $p(y|z)$. Exact probabilities are rounded for clarity.

(a)

Z (ignition)		Y (battery) 0	1	
	0			
X (fuel)	0	.125	.479	.604
	1	.271	.125	.396
		.396	.604	

(b)

Z (ignition)		Y (battery) 0	1	
	1			
X (fuel)	0	.000	.019	.019
	1	.019	.962	.981
		.019	.981	

Proof. The first implication $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \implies \exists(f, g)$ holds trivially. From Definition 1.5, we readily obtain $p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{x}, \mathbf{z})g(\mathbf{y}, \mathbf{z})$, with $f(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}, \mathbf{z})$ and $g(\mathbf{y}, \mathbf{z}) = p(\mathbf{y}|\mathbf{z})$ when $p(\mathbf{z}) > 0$, any positive value otherwise. To show the converse, let us express $p(\mathbf{x}, \mathbf{y}, \mathbf{z})p(\mathbf{z})$ and $p(\mathbf{x}, \mathbf{z})p(\mathbf{y}, \mathbf{z})$ in terms of the f and g functions. We obtain

$$p(\mathbf{x}, \mathbf{y}, \mathbf{z})p(\mathbf{z}) = f(\mathbf{x}, \mathbf{z})g(\mathbf{y}, \mathbf{z}) \sum_{\mathbf{x}', \mathbf{y}'} f(\mathbf{x}', \mathbf{z})g(\mathbf{y}', \mathbf{z}),$$

as well as

$$p(\mathbf{x}, \mathbf{z})p(\mathbf{y}, \mathbf{z}) = f(\mathbf{x}, \mathbf{z}) \left(\sum_{\mathbf{y}'} g(\mathbf{y}', \mathbf{z}) \right) g(\mathbf{y}, \mathbf{z}) \left(\sum_{\mathbf{x}'} f(\mathbf{x}', \mathbf{z}) \right),$$

which is equivalent. □

1.1.3 Independence models

Suppose we are given some conditional (in)dependence statements between random variables, then it would be very convenient if we could derive other (in)dependencies analytically. As we will see, reasoning about conditional independence will prove very useful to learn the structure of probabilistic graphical models from data efficiently, by means of statistical independence tests. To this end, we introduce the notion of independence model, along with an axiomatic characterization of the properties of conditional independence which will provide a formal deductive system.

indep. model

Def. 1.6 *An independence model I over a set $\mathbf{V}$ consists in a set of triples $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$, called independence relations, where $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ are disjoint subsets of $\mathbf{V}$ and $\langle \mathbf{X}, \emptyset \mid \mathbf{Z} \rangle$ and $\langle \emptyset, \mathbf{Y} \mid \mathbf{Z} \rangle$ always belong to I.*

I-map
D-map
P-map

Consider two independence models I_1 and I_2, defined over the same set $\mathbf{V}$. We say that I_1 is an independence map for I_2 (I-map for short), if all the independence relations in I_1 hold in a I_2 ($I_1 \subseteq I_2$). Equivalently, we say that I_1 is a dependence map for I_2 (D-map), when all the dependence relations in I_1 hold in a I_2 ($I_2 \subseteq I_1$). Finally, we say that I_1 is a perfect map for I_2 (P-map), when I_1 is both an I-map and a D-map for I_2 ($I_1 = I_2$).

faithfulness

Def. 1.7 *A probability distribution p defined over $\mathbf{V}$ is said faithful to an independence model I when all and only the independence relations in I hold in p, that is,*

$$\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \in I \iff \mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \quad \textit{w.r.t. } p.$$

An independence model I is said probabilistic, if there exists a probability distribution p that is faithful to it.

Conditional independence properties

The study of conditional independence properties goes back to the late seventies with Dawid and Spohn. In seminal theoretical developments, Dawid [Daw79; Daw80] proposes a first axiomatization of conditional independence properties, which unifies a variety of topics within probability and statistics under the same framework. At the same period, Spohn [Spo80] derives similar properties in his work on causal independence. These properties were then studied in great detail by Pearl and colleagues on their work on probabilistic graphical models, work that is presented in [Pea89].

Consider four mutually disjoint random variables, $\mathbf{W}$, $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$. We first introduce the following properties, which hold in any probability distribution ($\wedge$ and $\vee$ respectively denote the logical AND and OR operators):

- Symmetry: $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \iff \langle \mathbf{Y}, \mathbf{X} \mid \mathbf{Z} \rangle$.
- Decomposition: $\langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle \implies \langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$.
- Weak Union: $\langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle \implies \langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W} \rangle$.
- Contraction: $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \wedge \langle \mathbf{X}, \mathbf{W} \mid \mathbf{Z} \cup \mathbf{Y} \rangle \implies \langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle$.

semi-graphoid

Any independence model that respects these four properties is called a semi-graphoid [PV87].

A fifth property holds in strictly positive distributions, that is when $p > 0$:

- Intersection: $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W} \rangle \wedge \langle \mathbf{X}, \mathbf{W} \mid \mathbf{Z} \cup \mathbf{Y} \rangle \implies \langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle$.

graphoid

Any independence model that respects these five properties is called a graphoid. The term "graphoid" comes from Pearl and Paz [PP86], who noticed that these properties had striking similarities with vertex separation in graphs.

Finally, a sixth property will be of particular interest in our work:

- Composition: $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \wedge \langle \mathbf{X}, \mathbf{W} \mid \mathbf{Z} \rangle \implies \langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle$.

compositional graphoid

Any independence model that respects these six properties is called a compositional graphoid [SL14]. Similarly, any semi-graphoid which respects the composition property is called a compositional semi-graphoid. The composition property holds

in particular probability distributions, such as the regular multivariate Gaussian distribution, or, say, the symmetric binary distributions used in [WMC09].

The characterization problem

As stated previously, any probabilistic independence model satisfies the semi-graphoid properties. Because of that, the semi-graphoid properties provide a necessary condition to characterize probabilistic independence models. It is then possible to detect contradictory conditional independence relations, by checking if they respect the semi-graphoid properties. For example, different experts of the same domain may give different assumptions of conditional independence, which may not correspond together to any probability distribution.

The question of the converse implication arises naturally. Do the semi-graphoid properties provide a sufficient condition to characterize a probabilistic independence model?

A famous conjecture from Pearl [Pea89], known as Pearl's completeness conjecture, was that the graphoid axioms were a sufficient condition to characterize probabilistic independence models. Unfortunately, a counter-example was given by Studeny [Stu89], who found a set of conditional independence relations that respects the graphoid axioms, yet does not correspond to any faithful probability distribution. It consists in the following relations, plus the symmetric and trivial ones:

$$\langle A, B \mid \{C, D\}\rangle \wedge \langle C, D \mid A\rangle \wedge \langle C, D \mid B\rangle \wedge \langle A, B \mid \emptyset\rangle.$$

This counter-example suffices to disprove the conjecture. Moreover, we can see that this example satisfies also the semi-graphoid, the compositional graphoid, and the compositional semi-graphoid axioms, so neither of these sets provides a sufficient condition to characterize probabilistic independence models.

Afterwards, Studeny [Stu92] showed that, in the general case, there exists no finite set of conditional independence properties that is both a sufficient and necessary condition to characterize probabilistic independence models. Or, equivalently, probabilistic independence models have no finite complete axiomatic characterization. Later on, Sullivant [Sul09] showed that, even in the restricted case of probabilistic independence models over regular Gaussian distributions, no such characterization exists.

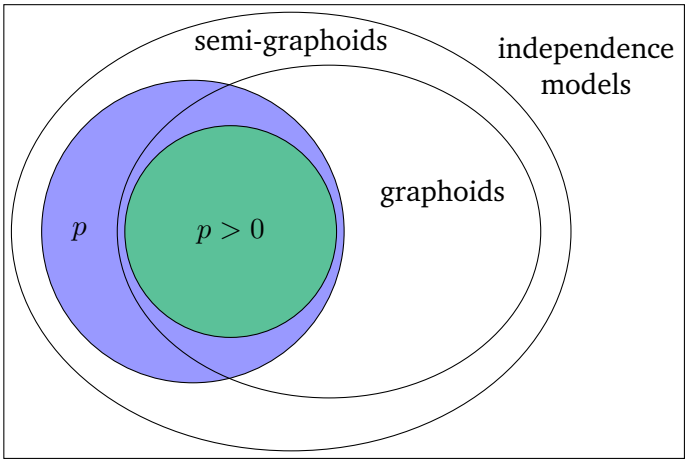


Fig. 1.4. Overlapping between different classes of independence models. Here p denotes models for which there exists a faithful probability distribution (probabilistic independence models), while $p > 0$ denotes models for which there exists a faithful strictly positive probability distribution. The compositional classes are omitted for clarity.

As we will see in the Chapter 2, probabilistic graphical models represent independence models in the form of a graph. In the next section we introduce some basic concepts and notations from graph theory which are common to graphical models.

1.2 Graph theory

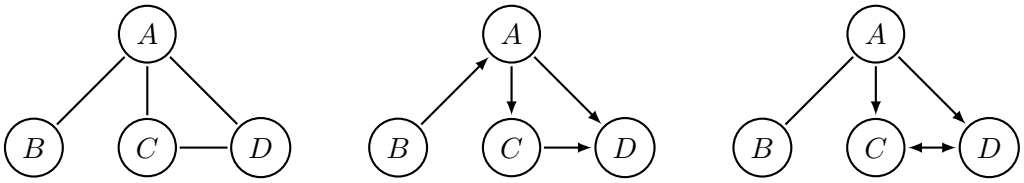
The following definitions are adapted from [SL14; Peñ14]. Formally, a graph $\mathcal{G}$ is
node defined as an ordered pair of sets $(\mathbf{V}, \mathcal{E})$. The first set, $\mathbf{V} = \{V_1, \ldots, V_n\}$, represents
edge the *nodes* (or vertices) of the graph, while the second set $\mathcal{E}$ represents its *edges*. An edge always associates two nodes (not necessarily distinct), called its *endpoints*.

Notice that under this notation our graphs are labeled, that is, every node is considered as a different object. Hence, for example, graph $A - B - C$ is not equal to $B - A - C$.

subgraph A *subgraph* of $\mathcal{G} = (\mathbf{V}, \mathcal{E})$ is any graph $\mathcal{G}' = (\mathbf{V}', \mathcal{E}')$ such that $\mathbf{V}' \subseteq \mathbf{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$. An *induced subgraph* is any such subgraph that contains all and only the edges that are present in $\mathcal{G}$ between pairs of nodes in $\mathbf{V}'$.

1.2.1 Connectivity, paths, walks, cycles

Two nodes V_i and V_j which are endpoints of the same edge are called *adjacent*. The adjacents of a set of nodes $\mathbf{X}$ in $\mathcal{G}$ is the set $\mathbf{AD}^{\mathcal{G}}_{\mathbf{X}} = \{V_1 | V_1$ is adjacent to V_2 in $\mathcal{G}, V_1 \notin$
clique $\mathbf{X}$ and $V_2 \in \mathbf{X}\}$. A *clique* is a set of nodes such that each node is adjacent to every



(a) Undirected. **(b)** Directed. **(c)** Mixed.

Fig. 1.5. Three simple graphs.

other node in the set. A *maximal clique* is a clique that does not accept any other clique as a proper superset.

walk
path
cycle

A *walk* between two nodes V_1 and V_k is a sequence of nodes in the form $V_1, \ldots, V_k$, $k \geq 1$, such that V_i is adjacent to V_{i+1} for all $1 \leq i < k$. Note that nodes in a walk are not necessarily distinct, i.e. the same node may appear several times. A walk with only distinct nodes is called a *path*. A walk with only distinct nodes except $V_1 = V_k$ is called a *cycle*. Within a walk (resp. path) $V_1, \ldots, V_k$, a subwalk (resp. subpath) is any sequence $V_i, \ldots, V_j$, $1 \leq i \leq j \leq k$, whose consecutive members appear consecutively in the walk.

connected set

Two nodes that accept a path between them are said *connected*. A *connected set* is a set of nodes $\mathbf{C}$ such that there is a path between each pair of nodes in $\mathbf{C}$ with all intermediate nodes in $\mathbf{C}$. A *maximal connected set* is a connected that does not accept any other connected set as a proper superset.

1.2.2 Classes of graphs

loopless
simple

A *loop* is an edge with the same endpoints, and *multiple edges* are edges with the same pair of endpoints. A *loopless graph* is a graph that has no loops. A *simple graph* is a graph that has neither loops nor multiple edges.

complete
connected
chordal

An *empty graph* is a graph that has no edges. A *complete graph* is a graph that has only one maximal clique. A *connected graph* is a graph that has only one maximal connected set. A *chordal graph* is a graph in which every cycle with more than 3 distinct nodes admits a smaller cycle as a proper subset.

Classical graphical models are restricted to simple graphs that contain only one type of edge: undirected, in the form $X - Y$, or directed, in the form $X \to Y$. Several attempts have been made to unify and increase the expressiveness of these models, by mixing directed and undirected edges, and by introducing new types of edges such as bi-directed ones in the form $X \leftrightarrow Y$. Figure 1.5 provides an illustration of such graphs.

2 Probabilistic graphical models

> *Probability does not exist.*
>
> — **Bruno De Finetti**
> 1974

Probabilistic graphical models belong the the family of probabilistic models, in the sense that they are able to represent a probability distribution p defined over a set of random variables. A PGM always consists in a set of parameters Θ and a graphical structure $\mathcal{G}$. The graphical structure encodes a set of conditional independence relations between nodes in the graph by the presence and absence of edges, and induces an independence model denoted $I(\mathcal{G})$. By definition, p satisfies every independence relations in $I(\mathcal{G})$, that is,

$$\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \in I(\mathcal{G}) \implies \mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} \quad \text{w.r.t. } p.$$

Equivalently, we say that $I(\mathcal{G})$ is an I-map for p. Note that the reverse implication is not required, that is, $I(\mathcal{G})$ is not necessarily a D-map for p.

Because p supports the conditional independence relations encoded in the graph, an interesting property of PGMs is that p factorizes according to $\mathcal{G}$. Once this graphical structure is known, the factorization supposedly makes the learning process easier (i.e. choose the best parameters Θ to estimate p from a finite number of samples), as well as the inference process (i.e. answer probabilistic queries from the model). When the graphical structure is unknown, it may also be learned from data samples. The whole process of learning a PGM that estimates a probability distribution p from a finite number of samples is usually split into two distinct problems, namely the structure learning problem (learn $\mathcal{G}$), and the parameter learning problem (learn Θ).

generative vs discriminative

Probabilistic models, among which are PGMs, may be further divided into two categories: generative models, which encode a probability distribution $p(\mathbf{v})$ over a set of random variables $\mathbf{V}$, and discriminative models, which encode a conditional probability distribution $p(\mathbf{y}|\mathbf{x})$ over two disjoint sets of random variables $\mathbf{X}$ and $\mathbf{Y}$. This chapter will focus on the former family of generative models, as discriminative models may be seen as a particular case of these.

2.1 Classical graphical models

We will now review the most popular models based on undirected graphs (Markov networks) and acyclic directed graphs (Bayesian networks). We will extend the discussion to advanced graphical models in the next section. Note that the literature abounds with different families of graphical models, and the list we discuss here is by far non-exhaustive.

2.1.1 Undirected graphs

The most popular probabilistic models based on undirected graphs are the Markov networks (MNs), also called Markov random fields, which emerged from different fields in the literature. Historically, the theory of Markov fields traces back to the Ising model [Isi25] from statistical physics, where undirected graphs were used to model geometric arrangements in space. Several types of Markov conditions were later introduced (see Lauritzen [Lau96] for an overview) in order to associate these graphs with independence models.

Undirected graphs as probabilistic models

Def. 2.1 *Markov network*

A Markov network consists in a set of random variables $\mathbf{V} = \{V_1, \ldots, V_n\}$, a simple undirected graph $\mathcal{G} = (\mathbf{V}, \mathcal{E})$, and a set of parameters Θ. Together, $\mathcal{G}$ and Θ define a probability distribution p over $\mathbf{V}$ which factorizes as

$$p(\mathbf{v}) = \prod_{\mathbf{C}_i \in \mathcal{C}l_{\mathcal{G}}} \phi_i(\mathbf{c}_i),$$

where $\mathcal{C}l_{\mathcal{G}}$ is the set of all cliques in $\mathcal{G}$.

Each ϕ_i function is called a factor, a potential function, or a clique potential. Note that, without loss of generality, the factorization of $p(\mathbf{v})$ may also be defined over the maximal cliques in $\mathcal{G}$.

Ex. 2.1 *Consider the undirected graphs in Figure 2.1. From the above definition, the corresponding factorizations over the maximal cliques are $p(\mathbf{v}) = \phi_1(a, b)\phi_2(b, c)\phi_3(c, d)\phi_4(d, a)$ in (a), and $p(\mathbf{v}) = \phi_1(a, b, d)\phi_2(d, b, c)$ in (b). Suppose that A, B, C and D are 4 binary variables, then Tables 2.1 and 2.2 respectively define valid clique potentials for each of these Markov network structures. These potentials are valid because their product defines a valid probability distribution, that is, $\sum_{a,b,c,d} p(a, b, c, d) = 1$. Note however that individual clique potentials do not necessarily normalize to 1, and therefore do not necessarily correspond to marginal or conditional probability distributions.*

Because of this, potential functions in Markov network are not subject to an intuitive probabilistic interpretation, and one must always go through a factorization to obtain proper probability measures.

Undirected graphs as independence models

Every undirected graph $\mathcal{G}$ induces a formal independence model $I(\mathcal{G})$ over $\mathbf{V}$, by means of a graphical separation criterion, called u-separation.

Def. 2.2 *UG ind. model* *Given an undirected graph $\mathcal{G}$, $I(\mathcal{G})$ is the independence model such that a disjoint triplet $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$ belongs to $I(\mathcal{G})$* iff $\mathbf{Z}$ *u-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$, that is, every path between a node in $\mathbf{X}$ and a node in $\mathbf{Y}$ contains a node in $\mathbf{Z}$.*

Note that the trivial relations $\langle \mathbf{X}, \emptyset \mid \mathbf{Z} \rangle$ and $\langle \emptyset, \mathbf{Y} \mid \mathbf{Z} \rangle$ are included in $I(\mathcal{G})$. From the above definition, the two extreme independence models correspond to the empty graph (without edges), where $I(\mathcal{G})$ contains every possible triplet, and the clique graph where $I(\mathcal{G})$ contains only trivial relations. Indeed, in undirected graphical models the addition of edges only creates dependence relations, while their removal creates independence relations.

Ex. 2.2 *Consider again the undirected graphs in Figure 2.1. In (a) the independence model is $\langle \{A\}, \{C\} \mid \{D, B\} \rangle \wedge \langle \{D\}, \{B\} \mid \{A, C\} \rangle$, while in (b) it reduces to $\langle \{A\}, \{C\} \mid \{D, B\} \rangle$ by the addition of a single edge.*

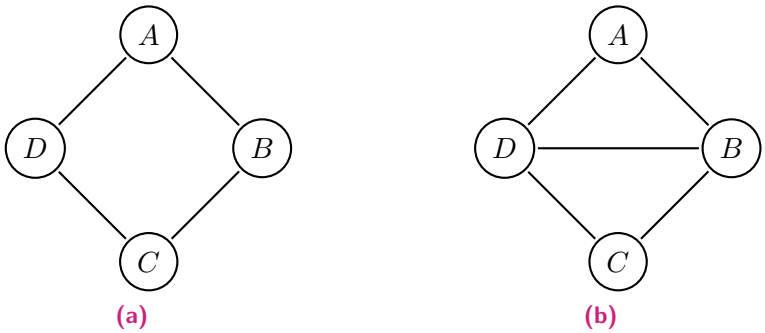
The next example illustrates u-separation in more complicated cases.

Ex. 2.3 *Consider the undirected graph in Figure 2.2. Here the independence model contains $\langle \{A\}, \{B\} \mid \{C\} \rangle$ because every path between A and B contains C, as well as $\langle \{A\}, \{B, F\} \mid \{C, E\} \rangle$ because every path between A and B or between A and F contains C. However, the relation $\langle \{A\}, \{B, F\} \mid \{D, E\} \rangle$ is not in $I(\mathcal{G})$ because there is a path between A and B which contains neither D nor E.*

Soundness of Markov networks

A Markov network structure always defines an I-map of the underlying probability distribution. Consider $\mathcal{G}$ an undirected graph over the variables $\mathbf{V}$, and p a probability distribution over the same set.

Thm. 2.1 *$I(\mathcal{G})$ is an I-map for p if p factorizes into a product of potentials over the cliques in $\mathcal{G}$.*



(a) (b)

Fig. 2.1. Two undirected graphs (UGs).

Tab. 2.1. A set of clique potentials (i.e. parameters Θ) that define a valid probability distribution according to the Markov network structure in Figure 2.1a.

(a) $\phi_1(a, b)$

		B	
		0	1
A	0	2/3	3/3
	1	1/3	1/3

(b) $\phi_2(b, c)$

		C	
		0	1
B	0	1/2	2/2
	1	1/2	3/2

(c) $\phi_3(c, d)$

		D	
		0	1
C	0	3/3	2/3
	1	2/3	1/3

(d) $\phi_4(d, a)$

		A	
		0	1
D	0	3/10	1/10
	1	1/10	2/10

Tab. 2.2. A set of clique potentials (i.e. parameters Θ) that define a valid probability distribution according to the Markov network structure in Figure 2.1b.

(a) $\phi_1(a, b, d)$

			B	
A	D		0	1
0	0		1/8	3/8
	1		1/8	1/8
1	0		2/8	1/8
	1		2/8	2/8

(b) $\phi_2(b, c, d)$

			B	
C	D		0	1
0	0		1/6	4/6
	1		1/6	1/6
1	0		1/6	2/6
	1		2/6	3/6

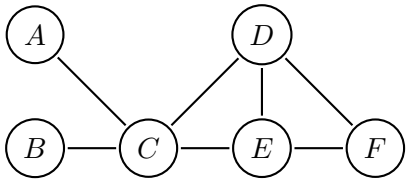


Fig. 2.2. An undirected graph to illustrate u-separation.

Proof. Let $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ be any three disjoint subsets of $\mathbf{V}$ such that $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \in I(\mathcal{G})$. We start by considering the case where $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z} = \mathbf{V}$. As $\mathbf{Z}$ u-separates $\mathbf{X}$ and $\mathbf{Y}$, there are no direct edges between $\mathbf{X}$ and $\mathbf{Y}$. Hence, any clique in $\mathcal{G}$ is fully contained either in $\mathbf{X} \cup \mathbf{Z}$ or in $\mathbf{Y} \cup \mathbf{Z}$. So we may re-write the factorization of p as

$$p(\mathbf{v}) = \prod_i \phi_i(\mathbf{c}_i) \cdot \prod_j \phi_j(\mathbf{c}_j),$$

where i indexes of the set of cliques that are contained in $\mathbf{X} \cup \mathbf{Z}$, while j indexes the remaining cliques. As discussed, none of the factors in the first product involves any variable in $\mathbf{Y}$, and none in the second product involves a variable in $\mathbf{X}$. Hence we can rewrite this product as $p(\mathbf{v}) = f(\mathbf{x}, \mathbf{z})g(\mathbf{y}, \mathbf{z})$. The desired independence relation follows immediately (Theorem 1.1), that is, $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$. In the case where $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z} \subset \mathbf{V}$, let us to consider the remaining set of variables $\mathbf{W} = \mathbf{V} \setminus (\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z})$ as follows. Necessarily, one can find a partition $\{\mathbf{W}_1, \mathbf{W}_2\}$ of $\mathbf{W}$ such that $\mathbf{Z}$ u-separates $\mathbf{X} \cup \mathbf{W}_1$ and $\mathbf{Y} \cup \mathbf{W}_2$ in $\mathcal{G}$. Using our precedent argument, we have that $\mathbf{X} \cup \mathbf{W}_1 \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W}_2 \mid \mathbf{Z}$. Using the decomposition property (from the semi-graphoid axioms), we obtain the desired result $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$. □

The converse implication was shown to be true only for strictly positive distributions[1]. This result is known as the Hammersley-Clifford's theorem [HC71].

Hammersley Clifford

Thm. 2.2 *Suppose that $p > 0$. Then, p factorizes into a product of potentials over the cliques in $\mathcal{G}$* iff *$I(\mathcal{G})$ is an I-map for p.*

Interestingly, Hammersley and Clifford found the positivity condition $p > 0$ unnatural, and postponed their publication in hope of relaxing it. Thereby, they were preceded by Besag [Bes74] in publishing the theorem. The condition was shown to be necessary shortly after by Moussouris [Mou74], who provided a simple counter-example with only 4 variables, which we present below.

Ex. 2.4 *Consider 4 binary variables A, B, C, D, and the probability distribution in Table 2.3. The undirected graph in Figure 2.1a is an I-map for p, because $A \perp\!\!\!\perp C \mid \{B, D\}$ and $B \perp\!\!\!\perp D \mid \{A, C\}$. However, one may observe that each combination of $\{A, B\}$, $\{B, C\}$, $\{C, D\}$ or $\{D, A\}$ has a positive probability, while some joint combinations of $\{A, B, C, D\}$ have zero probabilities. The only way to obtain a zero probability for a particular joint combination would be to set one of the clique potentials $\phi_1(\{a, b\})$, $\phi_2(\{b, c\})$, $\phi_3(\{c, d\})$ or $\phi_4(\{d, a\})$ to zero, which would immediately result in a zero probability for the corresponding pairwise combination. Thus, p cannot be encoded as a product of pairwise potentials.*

[1] Note that, in the discrete case, Geiger et al. [GMS02] give a necessary and sufficient condition that encompasses the positivity condition from the Hammersley-Clifford's theorem.

Tab. 2.3. Moussouris' counter-example of a probability distribution that satisfies the independence relations in a grid graph (Figure 2.1) but cannot be encoded as a product of pairwise potentials. Any $\{a, b, c, d\}$ combination that is not displayed has probability 0. This distribution does not satisfy the positivity condition.

A	B	C	D	$p(a, b, c, d)$
0	0	0	0	1/8
0	0	0	1	1/8
0	0	1	1	1/8
0	1	1	1	1/8
1	1	1	1	1/8
1	1	1	0	1/8
1	1	0	0	1/8
1	0	0	0	1/8

The Hammersley-Clifford's theorem has a practical application for Markov network structure learning. Assuming $p > 0$, learning a Markov network that estimates p can be done in two steps: 1) find a structure $\mathcal{G}$ that is an I-map for p; and 2) learn clique potentials ϕ_i that express p. Without the positivity condition, it is not guaranteed that the graph recovered after phase 1) will be able to correctly encode p.

Conditional independence properties of undirected graphs

Any independence model that can be expressed by u-separation over an undirected graph, i.e. for which there exists an undirected graph $\mathcal{G}$ that is a perfect map, is said UG-faithful. An interesting property of such independence models is that they are characterized by a finite set of conditional independence properties, as shown by Pearl and Paz [PP86].

Thm. 2.3 *Consider an independence model I defined over $\mathbf{V}$. A necessary and sufficient condition for I is to be UG-faithful is that it satisfies the following properties:*

- *Symmetry:* $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \iff \langle \mathbf{Y}, \mathbf{X} \mid \mathbf{Z} \rangle$.
- *Decomposition:* $\langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle \implies \langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$.
- *Intersection:* $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W} \rangle \wedge \langle \mathbf{X}, \mathbf{W} \mid \mathbf{Z} \cup \mathbf{Y} \rangle \implies \langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle$.
- *Strong union:* $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \implies \langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W} \rangle$.
- *Transitivity,* $\forall W \in \mathbf{W}$*:* $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \implies \langle \mathbf{X}, W \mid \mathbf{Z} \rangle \vee \langle W, \mathbf{Y} \mid \mathbf{Z} \rangle$.

Note that we can easily derive the weak union, contraction and composition properties from this set of axioms, and as a consequence independence models based on undirected graphs are compositional graphoids. First, weak union may be derived as follows: $\langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle$ implies $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$ due to the decomposition property, which in turn implies $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W} \rangle$ due to the strong union property. Second, contraction is derived as follows: $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W} \rangle \wedge \langle \mathbf{X}, \mathbf{W} \mid \mathbf{Z} \rangle$ implies $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W} \rangle \wedge \langle \mathbf{X}, \mathbf{W} \mid \mathbf{Z} \cup \mathbf{Y} \rangle$ due to the strong union, which in turn implies $\langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle$ due to the intersection. Finally, composition is derived as follows: $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \wedge \langle \mathbf{X}, \mathbf{W} \mid \mathbf{Z} \rangle$ implies $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W} \rangle \wedge \langle \mathbf{X}, \mathbf{W} \mid \mathbf{Z} \cup \mathbf{Y} \rangle$ due to the strong union, which in turn implies $\langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle$ due to the intersection.

A second important property of undirected graphs is that they always produce a probabilistic independence model. Indeed, it was shown by Geiger and Pearl [GP90] that for every independence model I that is UG-faithful, there exists a probability distribution p that satisfies all and only the independence relations in I. However, the converse does not necessarily hold, that is, not every probability distribution p is UG-faithful. Because of the axiomatic characterization given above, all and only the distributions that satisfy the intersection, the strong union and the transitivity property are UG-faithful.

Discussion

We will now emphasize some important points and caveats about Markov networks. First, any probability distribution can be encoded in a Markov network. It suffices to consider the complete graph $\mathcal{G}$, which does not impose any constraint on the expression of p, that is, $p(\mathbf{v}) = \phi(\mathbf{v})$. With a proper parameterization, ϕ may encode any probability distribution, even a non-strictly positive one such as in Example 2.4.

However, it must be kept in mind that the benefit of modeling a probability distribution with a Markov network comes from the sparseness of structure. The key idea of probabilistic graphical models is the factorization of p, which makes the learning and inference problems easier. This factorization comes from the independence relations encoded in the model, which for Markov networks directly result from the sparseness of the structure (recall that the absence of an edge only creates independence relations). The tractability of graphical models is often measured in terms of the *tree-width* of the model, which for undirected graphical models is given by the size of the largest clique, after the graph has been made chordal with added edges. Another way of assessing the tractability of a model is by measuring the number of degrees of freedom in its parameterization. Indeed, the more the

structure of a model imposes restrictions on p, the less free parameters remain to express p, and the easier will be the learning and inference problems.

Ex. 2.5 *Consider 3 binary random variables A, B and C. Their joint distribution can be represented as a contingency table in 3 dimensions, resulting in 2^3 parameters. Obviously one of these parameters is not free, as it can be deduced from the others due to the normalization $\sum_{a,b,c} p(a,b,c) = 1$. Still, we have $2^3 - 1 = 7$ free parameters to express p. If p supports some independence relations, such as $A \perp\!\!\!\perp C \mid B$, then it factorizes as $p(a,b,c) = p(a,b)p(c|b)$, in which case the number of free parameters is reduced to 5 ($2^2 - 1$ for $p(a,b)$ plus $2 \times (2^1 - 1)$ for $p(c|b)$). This corresponds to the undirected graph $A - B - C$ with $p(a,b,c) = \phi_1(a,b)\phi_2(b,c)$. In the extreme case where every single variable is independent of the others, we have $p(a,b,c) = p(a)p(b)p(c)$, in which case the number of free parameters is reduced to 3. This corresponds to the empty graph $A\,B\,C$ (without edges), with $p(a,b,c) = \phi_1(a)\phi_2(b)\phi_3(c)$.*

To summarize, for a graphical model to be interesting it should include as many of the independence relations in p as possible. However, the model must remain an I-map of p, that is, all the independence relations encoded in the structure must hold in p. In the case where p is UG-faithful, then there exists an "optimal" Markov network in some sense, that is, an undirected graphical model that perfectly encodes all and only the independence relations in p. However, and this is the main flaw of undirected graphical models, not every probability distribution is UG-faithful. That is, there are some probability distributions for which it is impossible to include all the independence relations in an undirected graph without violating a dependence relation as well.

Ex. 2.6 *Consider again the car parking example from Example 1.1. It can be verified that the only non-trivial independence relation is $X \perp\!\!\!\perp Y$. Then, p factorizes as $p(x,y,z) = p(x)p(y)p(z|x,y)$, which corresponds to 6 free parameters. Unfortunately, this factorization can not be exploited within an undirected graphical model. Indeed, any undirected model that encodes the relation $\langle X, Y \mid \emptyset \rangle$ necessarily implies $\langle X, Y \mid Z \rangle$ as well, due to the strong union property (Theorem 2.3). As a result, the only undirected graph that is an I-map for p is the complete graph, which necessarily results in 7 free parameters. In such a situation Markov networks do not appear to be well suited models to encode p.*

In the next section we will introduce directed acyclic graphical models, which have admittedly a higher expressive power that undirected models. However, directed acyclic models do not subsume undirected ones, as they suffer from different flaws and in some cases an undirected independence model is still better suited than a directed one.

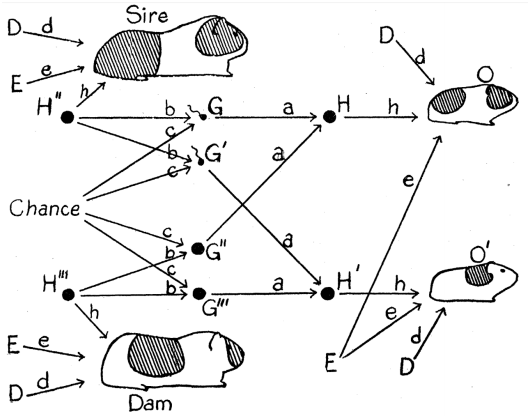


Fig. 2.3. A path diagram from Wright [Wri20], showing how the fur pattern of the litter guinea pigs is determined by various genetic and environmental factors.

2.1.2 Directed acyclic graphs

Within classical probabilistic graphical models, the directed counterpart of Markov networks are the Bayesian networks (BNs), which rely on directed acyclic graphs (DAGs). The modern definition of Bayesian networks as probabilistic independence models originates from the mid-1980s in a series of papers from Pearl and his colleagues [Pea85; VP88; GP88; GVP89; GVP90], work that is presented in great detail in Pearl [Pea89]. However, the idea of using directed acyclic graphs to encode general probability distributions goes back to the mid-1970s with influence diagrams in the context of decision analysis [HM81]. Furthermore, a first use of directed graphs to represent relations between random variables can be traced back to the 1920s with Wright [Wri20; Wri34] who introduced the notion of path diagrams in the context of genetics analysis (Figure 2.3).

Before talking about Bayesian networks, we must introduce some notions from graph theory that are specific to directed graphs. Recall that a directed graph $\mathcal{G} = (\mathbf{V}, \mathcal{E})$ contains only edges in the form $V_i \to V_j$. The *parents* of a set of nodes $\mathbf{X}$ is the set $\mathbf{PA}_{\mathbf{X}}^{\mathcal{G}} = \{V_1 | V_1 \to V_2 \text{ is in } \mathcal{G}, V_1 \notin \mathbf{X} \text{ and } V_2 \in \mathbf{X}\}$. Likewise, the *children* of $\mathbf{X}$ is the set $\mathbf{CH}_{\mathbf{X}}^{\mathcal{G}} = \{V_1 | V_1 \leftarrow V_2 \text{ is in } \mathcal{G}, V_1 \notin \mathbf{X} \text{ and } V_2 \in \mathbf{X}\}$. When clear from the context we may omit the superscript and just write $\mathbf{PA}_{\mathbf{X}}$ and $\mathbf{CH}_{\mathbf{X}}$.

parent

child

A *directed walk* from V_1 to V_k is a walk $V_1, \ldots, V_k$ such that $V_i \in \mathbf{PA}_{V_{i+1}}$ for all $1 \leq i < k$. Likewise, the definition of *directed path* and *directed cycle* follows. The *ancestors* of $\mathbf{X}$ is the set $\mathbf{AN}_{\mathbf{X}} = \{V_1 | V_1, \ldots, V_k \text{ is a directed path in } \mathcal{G}, V_1 \notin \mathbf{X} \text{ and } V_k \in \mathbf{X}\}$. Likewise, the *descendants* of $\mathbf{X}$ is the set $\mathbf{DE}_{\mathbf{X}} = \{V_1 | V_k, \ldots, V_1 \text{ is a directed path in } \mathcal{G}, V_1 \notin \mathbf{X} \text{ and } V_k \in \mathbf{X}\}$, and the non-descendants of $\mathbf{X}$ is the set $\mathbf{ND}_{\mathbf{X}} = \mathbf{V} \setminus (\mathbf{X} \cup \mathbf{DE}_{\mathbf{X}})$.

ancestor

descendant

directed acyclic graph

A *directed acyclic graph* (DAG for short) is a directed graph that contains no directed cycle. An equivalent characterization is a directed graph in which no node is both an ancestor and a descendant of the same node, that is, $\mathbf{AN}_V \cap \mathbf{DE}_V = \emptyset$ for every $V \in \mathbf{V}$. Some authors point that the phrase directed acyclic graph is ambiguous, and prefer to refer to acyclic directed graphs (ADG), or acyclic digraphs [Stu05, p. 46]. In this work we will follow the common practice in the field of Bayesian networks and refer to DAGs.

Directed acyclic graphs as probabilistic model

Def. 2.3 *Bayesian network*

A Bayesian network consists in a set of random variables $\mathbf{V} = \{V_1, \ldots, V_n\}$, a simple directed acyclic graph $\mathcal{G} = (\mathbf{V}, \mathcal{E})$, and a set of parameters Θ. Together, $\mathcal{G}$ and Θ define a probability distribution p over $\mathbf{V}$ which factorizes as

$$p(\mathbf{v}) = \prod_{V_i \in \mathbf{V}} p(v_i | \mathbf{pa}_{V_i}).$$

chain rule for BNs

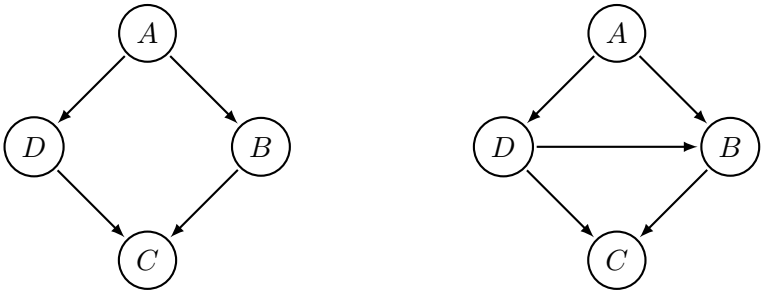
This factorization according to a DAG is called *recursive factorization*, or *chain rule for BNs*. Each of the factors $p(v_i|\mathbf{pa}_{V_i})$ can be seen as a potential function $\phi_i(v_i, \mathbf{pa}_{V_i})$, similarly to a clique potential in Markov networks. However, in Bayesian networks each factor must define a valid conditional probability distribution for V_i, and thus respects the normalization constraint $\sum_{v_i} \phi_i(v_i, \mathbf{pa}_{V_i}) = 1$.

Ex. 2.7 *Consider the DAGs in Figure 2.4. From the above definition, the corresponding factorizations are $p(\mathbf{v}) = p(a)p(d|a)p(b|a)p(c|b,d)$ in (a) and $p(\mathbf{v}) = p(a)p(d|a)p(b|a,d)p(c|b,d)$ in (b). Suppose that A, B, C and D are 4 binary variables, then Tables 2.4 and 2.5 respectively define valid conditional probability distributions for each of the Bayesian networks. In the context of discrete random variables, such tables are called conditional probability tables (CPTs). Note that each of these conditional probability distribution is normalized and can be intuitively interpreted. For example, from Table 2.4b we have that the event $D = 0$ is more likely to happen if we already know that $A = 0$ (probability 0.6), than when we know that $A = 1$ (probability 0.5).*

Directed acyclic graphs as independence models

Every DAG $\mathcal{G}$ induces a formal independence model $I(\mathcal{G})$ over $\mathbf{V}$, by means of a graphical separation criterion called d-separation [GVP90]. In order to define d-separation, we must first introduce the notion of a *collider* node. Within a path $V_1, \ldots, V_k$, an intermediate node V_i is said to be a collider *iff* it is in the form $V_{i-1} \rightarrow V_i \leftarrow V_{i+1}$.

collider



(a) (b)

Fig. 2.4. Two directed acyclic graphs (DAGs).

Tab. 2.4. A set of conditional probability tables that define a valid set of parameters Θ for the Bayesian network structure in Figure 2.4a.

(a) $p(a)$

A	
0	1
0.4	0.6

(b) $p(d|a)$

		D	
		0	1
A	0	0.6	0.4
	1	0.5	0.5

(c) $p(b|a)$

		B	
		0	1
A	0	0.3	0.7
	1	0.1	0.9

(d) $p(c|b,d)$

			C	
B	D		0	1
0	0		0.8	0.2
	1		0.7	0.3
1	0		0.5	0.5
	1		0.7	0.3

Tab. 2.5. The conditional probability table $p(b|a,d)$ that, along with Tables 2.4a, 2.4d and 2.4b, define a valid set of parameters Θ for the Bayesian network structure in Figure 2.4b.

		B	
A	D	0	1
0	0	0.2	0.8
	1	0.5	0.5
1	0	0.0	1.0
	1	0.2	0.8

Def. 2.4 *DAG ind. model* *Given an DAG $\mathcal{G}$, $I(\mathcal{G})$ is the independence model such that a disjoint triplet $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$ belongs to $I(\mathcal{G})$* iff $\mathbf{Z}$ *d-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$, that is, every path between a node in $\mathbf{X}$ and a node in $\mathbf{Y}$ contains a non-collider node that belongs to $\mathbf{Z}$, or a collider node that does not belong to $\mathbf{Z} \cup \mathbf{AN}_{\mathbf{Z}}$.*

Again, the trivial relations $\langle \mathbf{X}, \emptyset \mid \mathbf{Z} \rangle$ and $\langle \emptyset, \mathbf{Y} \mid \mathbf{Z} \rangle$ are included in $I(\mathcal{G})$. Note that some authors propose a different definition of $I(\mathcal{G})$ with the moralization criterion, which was shown to be equivalent to the d-separation criterion [Lau+90]. From the above definition, d-separation is equivalent to u-separation when $\mathcal{G}$ contains no *v-structure*, that is, no pattern in the form $V_1 \rightarrow V_2 \leftarrow V_3$ with V_1 and V_3 non-adjacent.

v-structure

Ex. 2.8 *Consider again the DAGs in Figure 2.4. In (a) the induced independence model is $\langle \{A\}, \{C\} \mid \{D, B\} \rangle \wedge \langle \{D\}, \{B\} \mid \{A\} \rangle$. This is different from the independence model induced from the undirected graph in Figure 2.1a, due to the v-structure $D \rightarrow C \leftarrow B$. In (b) the independence model is $\langle \{A\}, \{C\} \mid \{D, B\} \rangle$, which is equivalent to the one from the undirected graph in Figure 2.1b since here the DAG contains no v-structure.*

Similarly to the undirected case, the two extreme independence models correspond to the empty graph (without edges), where $I(\mathcal{G})$ contains every possible triplet, and a clique graph (without directed cycle), where $I(\mathcal{G})$ contains only trivial relations. Indeed, the addition of edges in a DAG only creates dependence relations, while their removal creates independence relations.

A friendly interpretation of d-separation is the following. Consider a path between two random variables X and Y, and a conditioning set $\mathbf{Z}$. The path represents an information flow. When $\mathbf{Z}$ is empty, each intermediate node that is not a collider is open, that is, it lets the flow go through. Conversely, each intermediate node that is a collider is closed, and blocks the flow. By adding some observed variables to $\mathbf{Z}$, one can only change the state of non-collider nodes from open to closed, and collider nodes from closed to open. When a non-collider node along the path is known (i.e. added to $\mathbf{Z}$) it becomes closed, and when a collider node or one of its descendants is known it becomes open. One then simply has to check if, given $\mathbf{Z}$, all the nodes along the path are open to determine if there is an information flow between X and Y. In that case the path is said active, and $\mathbf{Z}$ does not d-separate X and Y.

Ex. 2.9 *Consider the DAG in Figure 2.5. While in this graph the adjacencies are the same as in the undirected graph in Figure 2.2, the independence model is different due to the v-structure $A \rightarrow C \leftarrow D$. Here the independence model contains $\langle \{A\}, \{B\} \mid \{C\} \rangle$ because the only path $A \rightarrow C \rightarrow B$ is closed by the non-collider C that is observed. However, it does not contain $\langle \{A\}, \{B, F\} \mid \{C, E\} \rangle$ because in the path $A \rightarrow C \leftarrow D \rightarrow F$ the non-collider D is open, as well as the collider C that is observed. Other interesting relations induced by $\mathcal{G}$ are $\langle \{A\}, \{F\} \mid \emptyset \rangle \notin I(\mathcal{G})$, because of the open path*

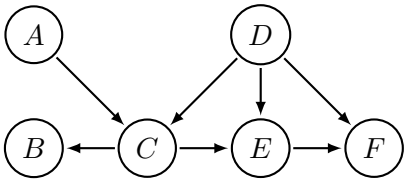


Fig. 2.5. A directed acyclic graph to illustrate d-separation.

$A \to C \to E \to F$. Conditioning on E does not d-separate A and F either, it closes the previous path but opens a new one with $A \to C \to E \leftarrow D$. We may then add C to the conditioning set, which closes the two previous paths but opens yet a new one with $A \to C \leftarrow D \to F$. We may now add D, which ultimately closes every path, and finally notice that conditioning on $\{C, D\}$ is sufficient and that E it no longer necessary in the conditioning set.

Soundness of Bayesian networks

As with Markov networks, a Bayesian network structure always defines an I-map of the underlying probability distribution. Moreover, the converse implication is true (recall that for MNs the converse holds only when $p > 0$). Consider $\mathcal{G}$ a directed acyclic graph graph over the variables $\mathbf{V}$, and p a probability distribution over the same set.

Thm. 2.4 *With $\mathcal{G}$ a DAG, $I(\mathcal{G})$ is an I-map for p* iff *p factorizes recursively over $\mathcal{G}$.*

Proof. We first prove the implication I-map $\implies$ factorization. Because $\mathcal{G}$ is a DAG, we may arrange its nodes in a topological ordering $V_1, \ldots, V_n$ according to $\mathcal{G}$, that is, $i < j$ if $V_i \to V_j$ is in $\mathcal{G}$. From the chain rule of probabilities, we can write p as

$$p(\mathbf{v}) = \prod_{i=1}^{n} p(v_i | v_1, \ldots, v_{i-1}).$$

Now, consider one of the factors $p(v_i|v_1, \ldots, v_{i-1})$. From the d-separation criterion, every node is independent of its non-descendants given its parents (a.k.a. local Markov property), that is, $V_i \perp\!\!\!\perp \mathbf{ND}_{V_i} \setminus \mathbf{PA}_{V_i} \mid \mathbf{PA}_{V_i}$. Because of our ordering of the nodes, all of V_i's parents are necessarily in the set $\{V_1, \ldots, V_{i-1}\}$, while none of its descendants can possibly be in the set. Then, we can write $\{V_1, \ldots, V_{i-1}\} = \mathbf{W} \cup \mathbf{PA}_{V_i}$, with $\mathbf{W} \subseteq (\mathbf{ND}_{V_i} \setminus \mathbf{PA}_{V_i})$. From the decomposition property we obtain $V_i \perp\!\!\!\perp \mathbf{W} \mid \mathbf{PA}_{V_i}$, which implies

$$p(v_i|v_1, \ldots, v_{i-1}) = p(v_i|\mathbf{pa}_{V_i}).$$

By applying this transformation to every such factor, we obtain the desired recursive factorization according to $\mathcal{G}$.

Second, we prove the converse. Let $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ be any three disjoint subsets of $\mathbf{V}$ such that $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \in I(\mathcal{G})$. As $\mathbf{Z}$ d-separates $\mathbf{X}$ and $\mathbf{Y}$, there are no direct edges between $\mathbf{X}$ and $\mathbf{Y}$. Hence, the variables in $\mathbf{X}$ have no parent in $\mathbf{Y}$ and the variables in $\mathbf{Y}$ have no parent in $\mathbf{X}$. Moreover, there are no pattern in the form $X \rightarrow Z \leftarrow Y$ with $X \in \mathbf{X}$, $Z \in \mathbf{Z}$ and $Y \in \mathbf{Y}$, so the variables in $\mathbf{Z}$ have either no parent in $\mathbf{X}$ or no parent in $\mathbf{Y}$. Let us now introduce $\mathbf{C}$, the set of all the remaining variables that have no descendant in $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$. In the case where $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z} \cup \mathbf{C} = \mathbf{V}$, we may re-write the factorization of p as

$$p(\mathbf{v}) = \prod_{V_i \in \mathbf{X} \cup (\mathbf{Z} \cap \mathbf{CH}_{\mathbf{X}})} p(v_i|\mathbf{pa}_{V_i}) \cdot \prod_{V_j \in \mathbf{Y} \cup (\mathbf{Z} \setminus \mathbf{CH}_{\mathbf{X}})} p(v_j|\mathbf{pa}_{V_j}) \cdot \prod_{V_l \in \mathbf{C}} p(v_l|\mathbf{pa}_{V_l}).$$

As discussed, none of the factors in the first product involves any variable in $\mathbf{Y} \cup \mathbf{C}$, and none in the second product involves any variable in $\mathbf{X} \cup \mathbf{C}$. Let us now marginalize out $\mathbf{C}$, we obtain

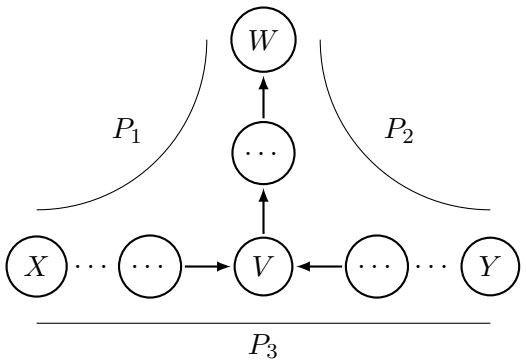
$$p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{x}, \mathbf{z}) g(\mathbf{y}, \mathbf{z}) \cdot \sum_{\mathbf{c}} \prod_{V_l \in \mathbf{C}} p(v_l|\mathbf{pa}_{V_l}).$$

Consider $C_1, \ldots, C_k$ an arrangement of the nodes in $\mathbf{C}$ in a topological ordering according to $\mathcal{G}$. Then, the third term transforms to a recursive sum

$$\sum_{c_1} p(c_1|\mathbf{pa}_{C_1}) \cdot \cdots \cdot \sum_{c_k} p(c_k|\mathbf{pa}_{C_k}).$$

The right-most term sums to 1 because of the normalization constraint, and vanishes. By summing each remaining term from right to left, the whole expression equals 1 and we obtain $p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{x}, \mathbf{z}) g(\mathbf{y}, \mathbf{z})$. The desired independence follows immediately (Theorem 1.1), that is, $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$. In the case where $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z} \cup \mathbf{C} \subset \mathbf{V}$, let us to consider the remaining set of variables $\mathbf{W} = \mathbf{V} \setminus (\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z} \cup \mathbf{C})$ as follows. Necessarily, one can find a partition $\{\mathbf{W}_1, \mathbf{W}_2\}$ of $\mathbf{W}$ such that $\mathbf{Z}$ d-separates $\mathbf{X} \cup \mathbf{W}_1$ and $\mathbf{Y} \cup \mathbf{W}_2$ in $\mathcal{G}$. Suppose it is not the case, then there is a node $W \in \mathbf{W}$ such that $\langle \mathbf{X}, \{W\} \mid \mathbf{Z} \rangle \notin I(\mathcal{G})$ and $\langle \{W\}, \mathbf{Y} \mid \mathbf{Z} \rangle \notin I(\mathcal{G})$. Equivalently, there is an open path $P_1 = (X, \ldots, V, \ldots, W)$ between a node $X \in \mathbf{X}$ and W, and an open path $P_2 = (Y, \ldots, V, \ldots, W)$ between a node $Y \in \mathbf{Y}$ and W, with $V, \ldots, W$ the shared sequence between P_1 and P_2 (of size 1 at least in the case where $V = W$). Because $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \in I(\mathcal{G})$, the path $P_3 = (X, \ldots, V, \ldots, Y)$ is closed, so V is a collider node in P_3 that is not in $\mathbf{Z}$ nor has any descendant in $\mathbf{Z}$. This implies that V has no descendant in $\mathbf{X}$ or $\mathbf{Y}$ either, otherwise this would result in an open path between $\mathbf{X}$ and $\mathbf{Y}$. Hence, V belongs to $\mathbf{C}$. This also implies that V is not a collider in P_1,

otherwise the path would be closed. Moreover, there can be no collider node along the path $V, \dots, W$, otherwise it would be a descendant of V without a descendant in $\mathbf{Z}$, and the path P_1 would be closed. So either W is a descendant of V, or $W = V$. In both cases W also belongs to $\mathbf{C}$, which violates our initial assumption $W \in \mathbf{W}$.



Using our precedent argument we have that $\mathbf{X} \cup \mathbf{W}_1 \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W}_2 \mid \mathbf{Z}$. Using the decomposition property we obtain the desired result $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$. □

Conditional independence properties of directed acyclic graphs

Any independence model that can be expressed by d-separation over a directed acyclic graph (i.e. for which there exists a DAG $\mathcal{G}$ that is a perfect map) is said to be DAG-faithful. Unlike in the case of undirected graphs, it was shown that such independence models can not be characterized by a finite set of conditional independence properties [Gei87; Li08; WWL02]. However, such a finite set of axioms can provide a necessary condition for an independence model to be faithful to a DAG. Pearl [Pea89] gives such a partial characterization.

Thm. 2.5 *Consider an independence model I defined over $\mathbf{V}$. A necessary condition for I to be DAG-faithful is that it satisfies the following properties:*

- *Symmetry:* $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \iff \langle \mathbf{Y}, \mathbf{X} \mid \mathbf{Z} \rangle$.
- *Decomposition:* $\langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle \implies \langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$.
- *Weak Union:* $\langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle \implies \langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W} \rangle$.
- *Contraction:* $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \wedge \langle \mathbf{X}, \mathbf{W} \mid \mathbf{Z} \cup \mathbf{Y} \rangle \implies \langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle$.
- *Intersection:* $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W} \rangle \wedge \langle \mathbf{X}, \mathbf{W} \mid \mathbf{Z} \cup \mathbf{Y} \rangle \implies \langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle$.

- *Composition:* $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \wedge \langle \mathbf{X}, \mathbf{W} \mid \mathbf{Z} \rangle \implies \langle \mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{Z} \rangle$.

- *Weak transitivity,* $\forall W \in \mathbf{W}$*:*
  $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \wedge \langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \cup W \rangle \implies \langle \mathbf{X}, W \mid \mathbf{Z} \rangle \vee \langle W, \mathbf{Y} \mid \mathbf{Z} \rangle$.

- *Chordality,* $\forall (X, Y, Z, W) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{Z} \times \mathbf{W}$*:*
  $\langle X, Y \mid \{Z, W\} \rangle \wedge \langle Z, W \mid \{X, Y\} \rangle \implies \langle X, Y \mid W \rangle \vee \langle X, Y \mid Z \rangle$.

As a direct consequence, DAG-faithful independence models are compositional graphoids.

Moreover, as in the undirected case, an important property of directed acyclic graphs is that they always produce a probabilistic independence model. Indeed, it was shown by Geiger and Pearl [GP88] that for every independence model I that is DAG-faithful, there exists a probability distribution that satisfies all and only the independence relations in I. However, here again the converse does not necessarily hold, that is, not every probability distribution is DAG-faithful.

A note on causal networks

We now briefly discuss the idea of causality and causal networks, although this is not required for the understanding of our work. We find necessary to add some clarifications to highlight the difference between a Bayesian network and a causal network, since many people do not distinguish between the two notions and tend to interpret every DAG as a causal graph.

Causal networks are basically Bayesian networks whose structure can be given a causal interpretation, that is, each directed edge represents a direct causal influence between a cause and an effect. Causal networks offer a powerful tool for reasoning about the causal influences among a set of random variables. A well-known paradox in statistics that is well explained with a causal DAG is the so-called Simpson's paradox, first described in Simpson [Sim51].

Simpson's paradox (1/2)

Ex. 2.10 *Consider a population of people who have the same disease, and among which some took a treatment, and some recovered the disease. Take three binary random variables G, T and R that respectively correspond to three indicators for each individual: his gender $g \in \{male, female\}$, he took a treatment $t \in \{yes, no\}$, and whether or not he recovered from the disease $r \in \{yes, no\}$. Let us observe 200 people randomly, 100 male and 100 female, which results in the probability distribution given in Table 2.6. We ask the following question: is the treatment efficient? In the general case, people who took the treatment tend to recover more: $p(R = yes|T = yes) > p(R = yes|T = no)$.*

Tab. 2.6. Illustration of Simpson's paradox. This table represents the joint probability distribution of G (gender), T (treatment) and R (recovery) on a population of 200 people (100 male and 100 female). $\#(g,t,r)$ counts the number of people that have characteristics (g,t,r) in the population, while $p(r|g,t)$ and $p(r|t)$ the corresponding conditional probability distributions.

T (treatment)	G (gender)	R (recovery) no	yes	no	yes	no	yes
no	male	42	28	.60	.40	.48	.52
	female	06	24	.20	.80		
yes	male	24	06	.80	.20	.45	.55
	female	21	49	.30	.70		
		$\#(g,t,r)$		$p(r|g,t)$		$p(r|t)$	

However, it also appears that women who took the treatment tend to recover less: $p(R = yes|T = yes, G = female) < p(R = yes|T = no, G = female)$. *And, very surprisingly, the same phenomenon appears in the population of men:* $p(R = yes|T = yes, G = male) < p(R = yes|T = no, G = male)$. *All these statements appear to be true according to* p*, but are rather counter-intuitive, hence the paradox. Let us to give it an explanation with a causal DAG.*

Suppose the causal DAG in Figure 2.6a represents the causal mechanisms between our three random variables. Admittedly, the causal directions in the DAG are plausible: the gender is determined when people are born thus can not be an effect of taking the treatment or recovering the disease, and the treatment is taken before the recovery in time thus can not be an effect of it. We may now re-formulate our initial question more explicitly, that is: does the action of taking the treatment has a positive causal effect on the recovery of the disease?

When measuring the probability $p(r|t)$*, two information paths are open in the graph (in the sense of* d*-separation): the direct causal path* $T \to R$ *and the indirect non-causal path* $T \leftarrow G \to R$ *due to the common cause* G*. Suppose the direct influence* $T \to R$ *is negative (taking the medicine has a negative effect on the recovery), then the statistical relationship carried in the first path is negative. At the same time, suppose the direct influences* $G \to T$ *and* $G \to R$ *are both positive (women tend to consume more medicine than men, and women naturally tend to recover the disease better than men), then the statistical relationship carried in the second path is positive as well (people who have taken the treatment tend to be women, who tend to recover better, so overall people who have taken medicine tend to recover better). When measuring the conditional probability* $p(r|t)$*, the statistical relationships carried in the two paths are mixed up, which results in a negative or a positive relationship depending on which path is the strongest. In the present situation the overall statistical relationship is positive. However, it can not be given any causal meaning since it carries information from the non-causal*

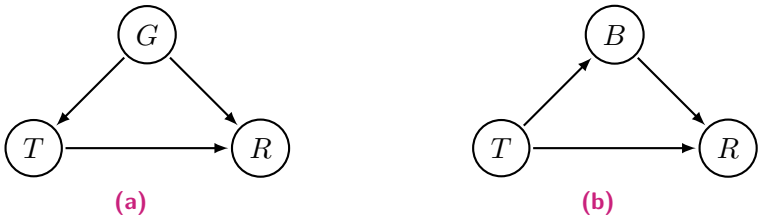


Fig. 2.6. Two causal DAGs illustrating Simpson's paradox.

path $T \leftarrow G \rightarrow R$. The second measurement $p(r|t,g)$ adds G to the conditioning set, which closes the non-causal path $T \leftarrow G \rightarrow R$ owing to the d-separation criterion. Because of that, only the causal path $T \rightarrow R$ remains, and the resulting statistical measurements can be interpreted as a causal relationships. So, with this causal DAG, it is the second set of measurements (consider men and women separately) that answers our initial question. The answer is then no, the treatment does not have a positive causal effect on the disease recovery.

By modeling Simpson's paradox with a causal DAG, we can measure unambiguously the causal influence between two random variables, by choosing a proper conditioning set $\mathbf{Z}$ that closes every non-causal path in the DAG and keeps every causal path open between the variables of interest. This measurement is known as the *do-calculus* [Pea95; Pea12], which is nowadays a well established tool for causal inference. It consists in measuring the probability of observing an event given that an action is performed, denoted $p(x|do(y))$. This is different from $p(x|y)$, i.e. the probability of observing an event given that another event is observed. In do-calculus, the structure of a causal DAG tells us how to compute $p(x|do(y))$. In Example 2.10, we have $p(r|do(t)) = \sum_g p(g)p(r|g,t)$ because of the common cause G. In that case G is commonly called a confounding variable. We can now measure the causal influence from T to R unambiguously, which gives $p(R = yes|do(T = no)) = 0.60$ and $p(R = yes|do(T = yes)) = 0.45$. We end up with the same conclusion we had reached in Example 2.10.

do-calculus

In Example 2.10, the direction of the edges was justified only by introducing some external knowledge, a.k.a. the expert knowledge. Without an expert to validate that the DAG structure indeed corresponds to a plausible causal system, no causal interpretation can be given to the measured probabilities. When someone tries to infer a causal explanation from observational data, that person always implicitly enforces an underlying causal system. If that causal system is wrong, such interpretations can lead to perilous conclusions.

Ex. 2.11 *Simpson's paradox (2/2)*

Consider the same probability distribution as in Example 2.10, but suppose that the variable G is replaced with the variable B, $b \in \{low, high\}$ representing the blood pressure level of each individual. The probability distribution is exactly the same,

however now the causal DAG could arguably be the one in Figure 2.6b. From our external knowledge, the blood pressure is more likely to be a consequence than a cause of taking the treatment. If we apply the do-calculus again to assess the efficiency of the treatment, then we have this time $p(r|do(t)) = p(r|t)$ *due to the causal structure of the DAG. This gives* $p(R = yes|do(T = no)) = 0.52$ *and* $p(R = yes|do(T = yes)) = 0.55$*, and we conclude that the treatment has a positive causal effect on the recovery of the disease. Same observational data, different causal system, different causal interpretation.*

It is important to keep in mind that it is impossible to infer causal mechanisms from observational data. It is not because a Bayesian network correctly encodes a probability distribution p, even faithfully, that the corresponding DAG necessarily represents causal mechanism that generated the data. Indeed, the two DAGs in Figure 2.6 are able to faithfully encode the probability distribution from Table 2.6. Without any additional information, both DAGs are indistinguishable with respect to the observed data, and represent equally plausible causal systems. However, performing do-calculus with these two DAGs does not lead to the same conclusions.

In the general case, it is impossible to infer causal relationships from a probability distribution based on observational data. The only way to infer such relationships is by collecting experimental data, that is, performing an action on the system and observe how it affects the probability distribution. This is well-known in the domain of clinical studies, as one can assess the effectiveness of a treatment only through double-blind trials (a.k.a. placebo-controlled studies). Readers interested in causality may consult Dawid [Daw10], or the very comprehensive book from Pearl [Pea09].

Discussion

We will now compare directed acyclic and undirected graphical models, namely Bayesian networks and Markov networks. We first discuss how Bayesian networks are similar to Markov networks, and finally we emphasize on their differences.

First, just like Markov networks, any probability distribution can be encoded in a Bayesian network. Consider an arbitrary ordering $V_1, \dots, V_n$ of the nodes, and the complete DAG such that $V_i \to V_j$ for every $j > i$. Then, the recursive factorization according to the DAG is $p(\mathbf{v}) = p(v_1)p(v_2|v_1) \dots p(v_n|v_1, \dots, v_{n-1})$, which respects the chain rule of probability and does not impose any constraint on p. This holds for any ordering of the nodes, and thus any complete DAG.

However, just like Markov networks, the structure of a Bayesian network should be as sparse as possible. This is true for probabilistic graphical models in general, and neither Markov networks or Bayesian networks are an exception to this rule. Only a small difference arises with Bayesian networks: while in undirected graphs the independence model results solely from the absence of edges, in DAGs it also results from the presence of v-structures. It is still true, however, that removing an edge in a DAG only creates independence relations, while adding an edge only creates dependence relations, thus the notion of sparseness for DAGs remains relevant.

The main difference between BNs and MNs comes from their expressive power, that is, their capacity to express independence models. Admittedly, the separation criterion is more complex for DAGs than from UGs, and in that sense it is less intuitive to interpret the structure of a BN than the structure of a MN. However, this higher complexity comes with an increased expressive power for DAGs. For the same number of nodes one can express more independence models with a DAG than with an UG. With 3 random variables, there are 8 distinct UG models and 11 DAG models. With 4 random variables, there are 64 UG models and 185 DAG models, and so on. However, DAG models do not subsume UG models, as there are probabilistic independence models that UG-faithful, but not DAG-faithful.

Ex. 2.12 *First, let us consider again the car parking example from Example 1.1, where the only non-trivial independence relation is $X \perp\!\!\!\perp Y$. In the last section, we showed that there exists no undirected graphical model that encodes only this relation. However, it is possible to encode this relation in the DAG $X \rightarrow Z \leftarrow Y$. In this example p is DAG-faithful, but not UG-faithful.*

Ex. 2.13 *Second, consider the Markov network structure in Figure 2.1a, along with the distribution encoded in Table 2.1. Clearly, the only non-trivial independence relations are $A \perp\!\!\!\perp C \mid B, D$ and $B \perp\!\!\!\perp D \mid A, C$, and the undirected graph is a perfect map for p. However, we can show that there exists no DAG model that encodes these two relations only. Suppose such a DAG exists, then due to the chordality property it also encodes either $\langle A, B \mid C \rangle$ or $\langle A, B \mid D \rangle$, which are not supported by p. Thus, in this example p is UG-faithful, but not DAG-faithful.*

Ex. 2.14 *Finally, consider a noisy exclusive OR (XOR) relationship between three random variables A, B and C, with the probability distribution represented in Table 2.7. Here p supports only three non-trivial independence relations $A \perp\!\!\!\perp B$, $B \perp\!\!\!\perp C$ and $C \perp\!\!\!\perp A$. Unfortunately, there exists no undirected graph that can encode all and only these independence relations, that is, p is not UG-faithful. Indeed, suppose such a graph exists, then due to the strong union property it also induces an independence relation that is not in p ($A \perp\!\!\!\perp B \implies A \perp\!\!\!\perp B \mid C$, and so on). Even worse, it also means that there exists no undirected graph that can encode any of the independence relations in p, so at best a Markov network model requires 7 free parameters to encode p, with a complete graph. What about Bayesian networks? We can show that p is not DAG-faithful either. Due to*

the composition property, any DAG that encodes two of the independence relations in p necessarily breaks a dependence relation as well ($A \perp\!\!\!\perp B \wedge A \perp\!\!\!\perp C \implies A \perp\!\!\!\perp \{B, C\}$, and so on). However, there are some structures that are able to encode only one of the independence relations, such as the DAG $A \rightarrow C \leftarrow B$. This BN structure results in the factorization $p(a, b, c) = p(a)p(b)p(c|a, b)$, which expresses p with 6 free parameters. In this example p is neither UG-faithful nor DAG-faithful, so both Markov networks and Bayesian networks are not well-suited models to encode p efficiently.

And yet, we can show that any distribution of binary random variables that supports the independence relations in p can be encoded efficiently with only 4 parameters. Due to the independence relation $A \perp\!\!\!\perp B$, p can be factorized as $p(a, b, c) = p(a)p(b)p(c|a, b)$. This reduces the number of free parameters to 6, for example: $p(\alpha)$, $p(\beta)$, $p(\gamma|\alpha, \beta)$, $p(\gamma|\alpha, \bar{\beta})$, $p(\gamma|\bar{\alpha}, \beta)$ and $p(\gamma|\bar{\alpha}, \bar{\beta})$. We will now show that, due to $B \perp\!\!\!\perp C$ and $C \perp\!\!\!\perp A$, two of these parameters are not free to vary once the other ones are known. Because $B \perp\!\!\!\perp C$ we can write

$$p(\bar{\beta}, \gamma) = p(\bar{\beta})p(\gamma).$$

We substitute $p(\bar{\beta}, \gamma)$ by $\sum_a p(a, \bar{\beta}, \gamma)$, and $p(\gamma)$ by $\sum_{a,b} p(a, b, \gamma)$ to obtain

$$p(\alpha, \bar{\beta}, \gamma) + p(\bar{\alpha}, \bar{\beta}, \gamma) = p(\bar{\beta}) \left[p(\alpha, \beta, \gamma) + p(\alpha, \bar{\beta}, \gamma) + p(\bar{\alpha}, \beta, \gamma) + p(\bar{\alpha}, \bar{\beta}, \gamma) \right].$$

Then, we regroup the $p(\bar{\alpha}, \bar{\beta}, \gamma)$ and $p(\alpha, \bar{\beta}, \gamma)$ terms on one side to obtain

$$p(\beta)p(\bar{\alpha}, \bar{\beta}, \gamma) = p(\bar{\beta}) \left[p(\alpha, \beta, \gamma) + p(\bar{\alpha}, \beta, \gamma) \right] - p(\beta)p(\alpha, \bar{\beta}, \gamma).$$

Finally, because $A \perp\!\!\!\perp B$ we replace each $p(a, b, c)$ term by $p(a)p(b)p(c|a, b)$. We readily obtain

$$p(\gamma|\bar{\alpha}, \bar{\beta}) = \frac{p(\alpha)}{1 - p(\alpha)} \left[p(\gamma|\alpha, \beta) - p(\gamma|\alpha, \bar{\beta}) \right] + p(\gamma|\bar{\alpha}, \beta).$$

Thus, the parameter $p(\gamma|\bar{\alpha}, \bar{\beta})$ is induced by 4 other parameters. Notice that we used only two independence relations $A \perp\!\!\!\perp B$ and $B \perp\!\!\!\perp C$. We may follow the same reasoning with $A \perp\!\!\!\perp B$ and $A \perp\!\!\!\perp C$ to derive a similar result, that is,

$$p(\gamma|\bar{\alpha}, \bar{\beta}) = \frac{p(\beta)}{1 - p(\beta)} \left[p(\gamma|\alpha, \beta) - p(\gamma|\bar{\alpha}, \beta) \right] + p(\gamma|\alpha, \bar{\beta}).$$

By combining these two statements we readily obtain that the parameter $p(\beta)$ is also induced by the same set of 4 parameters, that is,

$$p(\beta) = \left(\left[\frac{\frac{p(\alpha)}{1-p(\alpha)} \left[p(\gamma|\alpha, \beta) - p(\gamma|\alpha, \bar{\beta}) \right] + p(\gamma|\bar{\alpha}, \beta) - p(\gamma|\alpha, \bar{\beta})}{p(\gamma|\alpha, \beta) - p(\gamma|\bar{\alpha}, \beta)} \right]^{-1} + 1 \right)^{-1}.$$

In the end, the two parameters $p(\beta)$ and $p(\gamma|\bar{\alpha},\bar{\beta})$ are not free to vary once $p(\alpha)$, $p(\gamma|\alpha,\beta)$, $p(\gamma|\alpha,\bar{\beta})$ and $p(\gamma|\bar{\alpha},\beta)$ are fixed, which reduces the number of free parameters that encode p to 4.

Tab. 2.7. A probability distribution $p(a,b,c)$ of three binary random variables that corresponds to the noisy XOR relationship $P(A = B \oplus C) = 1 - \epsilon$ (exclusive OR). Here p supports the positivity condition ($p > 0$) for any $\epsilon \in]0, 1/2[\cup]1/2, 0[$.

		C	
A	B	$\bar{\gamma}$	γ
$\bar{\alpha}$	$\bar{\beta}$	$(1-\epsilon)/4$	$\epsilon/4$
	β	$\epsilon/4$	$(1-\epsilon)/4$
α	$\bar{\beta}$	$\epsilon/4$	$(1-\epsilon)/4$
	β	$(1-\epsilon)/4$	$\epsilon/4$

What can we conclude about the difference between undirected graphical models (Markov networks) and directed acyclic graphical models (Bayesian networks)? First, none of these models is superior to the other, as there are probability distributions for which a DAG is better-suited than an UG to capture the underlying independence model, as well as cases where the contrary stands. The two models are not complementary either, as there are probability distributions that are neither DAG-faithful nor UG-faithful. This situation is pictured in Figure 2.7. An interesting class of probabilistic graphical models are the *decomposable models*, which correspond to independence models that are both UG-faithful and DAG-faithful, that is, chordal UGs and DAGs without v-structure. de Campos [de 96] shows that such models are characterized by a finite set of conditional independence properties, that is, those in Theorem 2.3 plus the *strong chordality* axiom:

decomposable models

$$\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{W} \rangle \wedge \langle \mathbf{Z}, \mathbf{W} \mid \mathbf{X} \cup \mathbf{Y} \rangle \implies \langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \vee \langle \mathbf{X}, \mathbf{Y} \mid \mathbf{W} \rangle.$$

A decomposable model is also characterized by the existence of a junction tree over its cliques, which allows for efficient learning and inference methods [Cow+99]. Still, decomposable models are weaker than general UGs and DAGs as independence models, in the sense that they are much less expressive.

Despite the inherent limitation of UG and DAG models, they are still very frequently used in the machine learning community due to their simple structure which is rather intuitive to interpret, and also because the factorization remains simple and there exists plenty of tools available to solve both the learning and inference problems with these models. An interesting question to ask is if we could find a graphical independence model that subsumes both DAG and UG models, maybe by allowing a mixture of undirected and directed edges in the graph? This idea of mixing directed and undirected graphical models is rather old, as it can be traced back to Verma and Pearl [VP88; VP90] who introduced the idea of hybrid graphs. In the next section

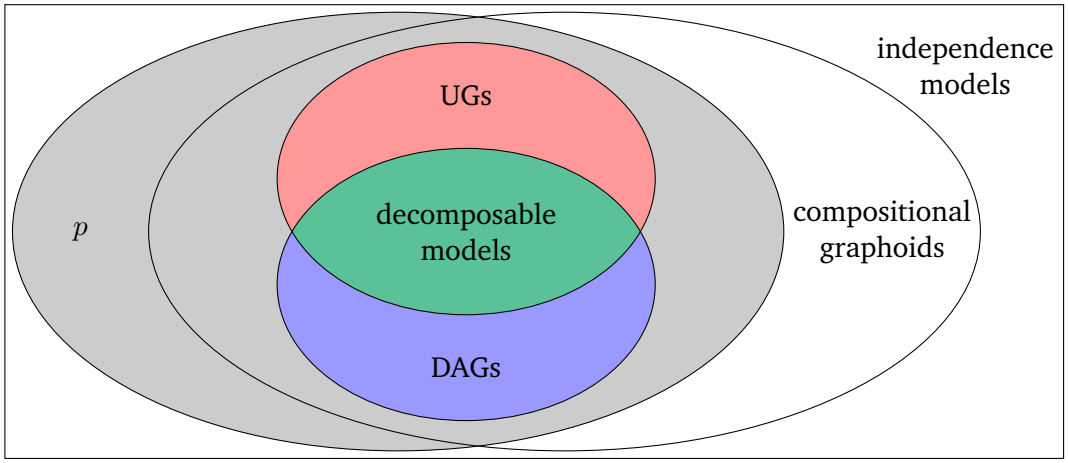


Fig. 2.7. Overlapping between probabilistic independence models (p), independence models based on u-separation (UG-faithful), and d-separation (DAG-faithful).

we will try to give a brief summary of the recent advances in advanced graphical models.

2.2 Advanced graphical models

We will now briefly review the state-of-the-art and recent developments in the field of advanced probabilistic graphical models. As discussed in the previous section, classical PGMs based on undirected and directed acyclic graphs suffer from limitations inherent to their restricted expressive power as independence models. As we will see next, over the years many alternative graphical models have been proposed to overcome these limitations, by extending and unifying the expressive power of UGs and DAGs. The study of advanced graphical models is still an active area of research today, and a lot of different families of models and interpretations can be found in the literature. In this section we will try to adopt an epistemological approach, by following the development of probabilistic graphical models chronologically.

We will use the notation from Sadeghi and Lauritzen [SL15] who unify most of the families of advanced graphical models with graphs made of four types of edges: directed edges denoted by arrows $\rightarrow$, and three types of undirected edges denoted by lines $-$, arcs $\leftrightarrow$ and dashed arcs $\dashleftarrow\dashrightarrow$.

The definitions of parents, children, ancestors and descendants introduced in the context of directed graphs remain valid in this context. Additionally, the *neighbours* of a set of nodes $\mathbf{X}$ is the set $\mathbf{NE_X} = \{V_1 | V_1 - V_2$ is in $\mathcal{G}, V_1 \notin \mathbf{X}, V_2 \in \mathbf{X}\}$. The *spouses* of a set of nodes $\mathbf{X}$ is the set $\mathbf{SP_X} = \{V_1 | V_1 \leftrightarrow V_2$ is in $\mathcal{G}, V_1 \notin \mathbf{X}, V_2 \in \mathbf{X}\}$.

neighbour

spouse

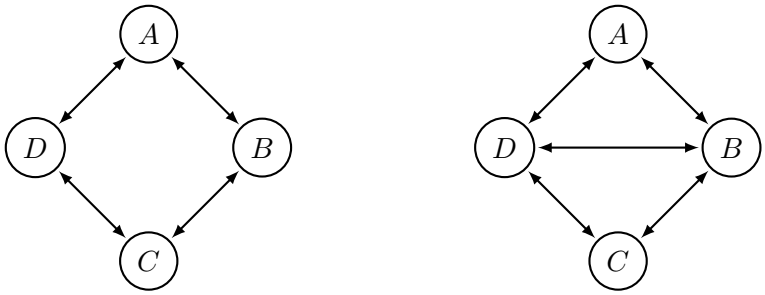


Fig. 2.8. Two bi-directed graphs (BGs). The non-trivial independence relations induced by b-separation are $\langle A, C \mid \emptyset \rangle$ and $\langle B, D \mid \emptyset \rangle$ for the first graph, and only $\langle A, C \mid \emptyset \rangle$ for the second.

partner

The *partners* of a set of nodes $\mathbf{X}$ is the set $\mathbf{PT}_{\mathbf{X}} = \{V_1 | V_1 \dashleftarrow\dashrightarrow V_2$ is in $\mathcal{G}, V_1 \notin \mathbf{X}, V_2 \in \mathbf{X}\}$.

semi-dir. cycle

A *semi-directed cycle* is a cycle $V_1, \ldots, V_k$ such that $V_i \in \mathbf{PA}_{V_{i+1}}$ for at least one $1 \leq i < k$, and $V_i \notin \mathbf{CH}_{V_{i+1}}$ for all $1 \leq i < k$. In other words, a semi-directed cycle is composed only of lines $-$, arcs $\dashleftarrow\dashrightarrow$ or arrows $\rightarrow$ pointing from left to right, and contains at least one arrow (it may contain only arrows).

2.2.1 Bi-directed graphs

In Speed and Kiiveri [SK86] appears the idea of representing zero hypotheses on the covariance matrix of normally distributed variables in the form of a graph. The idea is further developed by [CW93; Kau96], and results in a new probabilistic graphical model called a *covariance graph*, whose structure is a simple undirected graph with dashed edges to distinguish it from classical undirected graphical models, a.k.a. Markov networks. The interpretation of covariance graphs in terms of independence model appears to be dual to the interpretation of classical undirected graphs, which in the case of normally distributed variables represent zero hypotheses on the inverse covariance matrix. To be consistent with recent practice, we represent covariance graphs with bi-directed edges $\dashleftarrow\dashrightarrow$.

The separation criterion for bi-directed graphs (BGs for short) is given by Drton and Richardson [DR08]. We conveniently name it b-separation.

Def. 2.5 *BG ind. model*

Given a bi-directed graph $\mathcal{G}$, $I(\mathcal{G})$ is the independence model such that a disjoint triplet $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$ belongs to $I(\mathcal{G})$ iff $\mathbf{Z}$ b-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$, that is, every path between a node in $\mathbf{X}$ and a node in $\mathbf{Y}$ contains a node not in $\mathbf{Z}$.

Interestingly, it appears that the BG interpretation of an undirected graph is the dual of the UG interpretation using u-separation. Equivalently, BG graphs may be interpreted with the d-separation criterion if we re-define the pattern of a collider node as $\leftrightarrow V_i \leftrightarrow$. As for conditional independence properties, it can be found in [RS02][Theorem 7.5] that every BG independence model is probabilistic.

The general factorization associated with a BG model is also given by Drton and Richardson [DR08].

Def. 2.6 *BG prob. model*

An BG model consists in a set of random variables $\mathbf{V} = \{V_1, \ldots, V_n\}$, a simple bi-directed graph $\mathcal{G} = (\mathbf{V}, \mathcal{E})$, and a set of parameters Θ. Together, $\mathcal{G}$ and Θ define a probability distribution p over $\mathbf{V}$ which factorizes, for every subset $\mathbf{S} \subseteq \mathbf{V}$, as

$$p(\mathbf{s}) = \prod_{\mathbf{C}_i \in \mathcal{C}m_{\mathcal{G}_\mathbf{S}}} p(\mathbf{c}_i)$$

where $\mathcal{C}m_{\mathcal{G}_\mathbf{S}}$ is the set of all maximal connected sets in $\mathcal{G}_\mathbf{S}$, the induced subgraph of $\mathcal{G}$ over $\mathbf{S}$.

Obviously, the parameterization of a BG model appears less trivial than that of an UG model, since the constraints on p are expressed in the form of several factorizations on marginal distributions. Nevertheless, Drton and Richardson [DR08] give practical solutions to parameterize BG models in the discrete case, by expressing p in terms of its *saturated Möbius parameters*. As for soundness, it was also shown by Drton and Richardson [DR08] that, with $\mathcal{G}$ a BG, $I(\mathcal{G})$ is an I-map for p *iff* p factorizes according to $\mathcal{G}$.

2.2.2 Chain graphs

The idea of combining directed and undirected edges to form hybrid graphical independence model can be traced back to Verma and Pearl [VP88], who briefly emits a separation criterion for so-called hybrid graphs. Shortly after, the concept of chain graph appears in the literature, that is, a graph with two types of edges, directed and undirected, that accepts no semi-directed cycle. We will see that, under a particular interpretation, each chain graph $\mathcal{G}$ defines a probabilistic independence model $I(\mathcal{G})$ and a corresponding factorization of p which is a necessary condition for $I(\mathcal{G})$ to be an I-map of p. As of now, three different interpretation of chain graphs as probabilistic graphical models have been studied, namely the LWF, AMP and MVR interpretations. A fourth possible interpretation exists according to Drton [Drt09], and appears to be the dual of the AMP interpretation [SL14]. Depending on which interpretation is applied, for the same chain graph $\mathcal{G}$ both $I(\mathcal{G})$ and the induced factorization may differ. Therefore, each of these chain graph interpretations defines

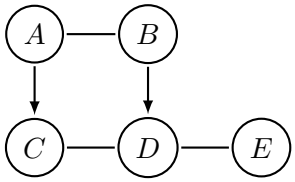


Fig. 2.9. A chain graph (CG), whose chain components are $\{A, B\}$ and $\{C, D, E\}$.

a new family of probabilistic graphical models on its own. We will now introduce the common concepts shared by these interpretations.

Common definitions

chain graph

A *chain graph* is a simple graph that admits two types of edges, directed $\rightarrow$ and undirected $-$, and accepts no semi-directed cycle.

chain component

A *chain component* of a chain graph $\mathcal{G}$ is a maximal undirected connected set, that is, a maximal connected set in the subgraph over $\mathbf{V}$ that contains only the undirected edges in $\mathcal{G}$. Thus, each chain graph decomposes uniquely into chain components.

section

A *section* of a walk $V_1, \ldots, V_k$ is an intermediate subwalk $V_i, \ldots, V_j$, $1 < i \leq j < k$ that is undirected and maximal, that is, which does not accept any other such intermediate undirected subwalk as a proper superset. Thus, any walk decomposes uniquely into sections.

LWF chain graphs

Lauritzen and Wermuth [LW89] propose the first probabilistic graphical model whose structure is a *chain graph*. This new interpretation of chain graphs is later completed by Frydenberg [Fry90], and is now referred to as the LWF-CG model (Lauritzen, Wermuth, Frydenberg).

collider section

The separation criterion for LWF chain graphs is called c-separation [SB98], and is based on the notion of collider sections. In an LWF chain graph, a *collider section* is any section $\rho = (V_i, \ldots, V_j)$ that follows the following pattern in the walk: $\rightarrow \rho \leftarrow$. Collider sections in LWF-CGs can be seen as an extension of colliders in DAGs, in which all sections are single nodes.

Def. 2.7 *LWF-CG ind. model*

Given an LWF-CG $\mathcal{G}$, $I(\mathcal{G})$ is the independence model such that a disjoint triplet $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$ belongs to $I(\mathcal{G})$ iff *$\mathbf{Z}$ c-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$, that is, every walk between*

a node in $\mathbf{X}$ *and a node in* $\mathbf{Y}$ *contains a non-collider section that has a node in* $\mathbf{Z}$*, or a collider section that has no node in* $\mathbf{Z}$*.*

It is clear that c-separation for LWF chain graphs reduces to u-separation in the case of an UG, and d-separation in the case of a DAG. Note that the use of walks in the c-separation criterion seems problematic at first sight, as the number of walks in a graph is potentially infinite. This problem is alleviated by Studený [Stu98], who shows that an efficient local algorithm exists to check for c-separation. Moreover, one can argue that the formulation of separation in DAGs is made simpler and somehow more elegant with the c-separation criterion than with the d-separation criterion, as it is no longer necessary to refer to the descendants of colliders. As for conditional independence properties, it was shown by Studený [Stu97] that every LWF-CG independence model is probabilistic.

closure graph

The factorization of p according to an LWF-CG is given by [Fry90][Theorem 4.1], and relies on the notion of *closure graph*. Given an LWF chain graph $\mathcal{G}$ and a chain component $\mathbf{K}$, the closure graph $\mathcal{H}(\mathcal{G}, \mathbf{K})$ is obtained as follows: i) take $\mathcal{H}$ the induced subgraph of $\mathcal{G}$ over the node set $\mathbf{K} \cup \mathbf{PA_K}$; ii) add an edge between each pair of nodes in $\mathbf{PA_K}$; iii) make each directed edge undirected.

Def. 2.8 *LWF-CG prob. model*

An LWF-CG model consists in a set of random variables $\mathbf{V} = \{V_1, \ldots, V_n\}$*, a chain graph* $\mathcal{G} = (\mathbf{V}, \mathcal{E})$*, and a set of parameters* Θ*. Together,* $\mathcal{G}$ *and* Θ *define a probability distribution* p *over* $\mathbf{V}$ *which factorizes as*

$$p(\mathbf{v}) = \prod_{\mathbf{K}_i \in \mathcal{C}c_\mathcal{G}} p(\mathbf{k}_i | \mathbf{pa}_{\mathbf{K}_i})$$

where $\mathcal{C}c_\mathcal{G}$ *is the set of all chain components in* $\mathcal{G}$*, and each* $p(\mathbf{k}_i|\mathbf{pa}_{\mathbf{K}_i})$ *term further factorizes as*

$$p(\mathbf{k}_i | \mathbf{pa}_{\mathbf{K}_i}) = \prod_{\mathbf{C}_j \in \mathcal{C}l_{\mathcal{H}(\mathcal{G}, \mathbf{K}_i)}} \phi_j(\mathbf{c}_j)$$

where $\mathcal{C}l_{\mathcal{H}(\mathcal{G},\mathbf{K}_i)}$ *is the set of all cliques in the closure graph* $\mathcal{H}(\mathcal{G}, \mathbf{K}_i)$*, and* ϕ_j *is a positive function.*

It is clear that the above factorization reduces to the clique factorization of a Markov network in the case of a fully undirected graph, and the recursive factorization of a Bayesian network in the case of a fully directed graph. As for soundness, it was shown by Frydenberg [Fry90] that, with $\mathcal{G}$ an LWF-CG, $I(\mathcal{G})$ is an I-map for p if p factorizes according to $\mathcal{G}$, and the converse holds if $p > 0$.

As LWF-CGs generalize both UGs and DAGs, they appear to be superior in every point to classical probabilistic graphical models. However, LWF-CGs do not subsume BGs.

AMP chain graphs

Shortly after LWF chain graph models were introduced, Andersson et al. [AMP96] propose an alternate interpretation of chain graphs as probabilistic independence models, which is now referred to as the AMP-CG models (Andersson, Madigan, Perlman, or conveniently Alternative Markov Property in the original paper).

collider node

The new separation criterion for a AMP chain graphs is called p-separation [LPM01], and extends the notion of collider nodes from DAGs (whereas c-separation was based on collider sections). In an AMP chain graph, a *collider node* in a walk is any intermediate node V_i that follows one of the following patterns: $\rightarrow V_i \leftarrow$ or $-V_i \leftarrow$. The pattern $\rightarrow V_1 -$ may be excluded without loss of generality, since p-separation is symmetric. Colliders in AMP-CGs can be seen as an extension of colliders in DAGs, in which only the pattern $\rightarrow V_i \leftarrow$ appears.

AMP-CG ind. model

Def. 2.9 *Given an AMP-CG $\mathcal{G}$, $I(\mathcal{G})$ is the independence model such that a disjoint triplet $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$ belongs to $I(\mathcal{G})$* iff *$\mathbf{Z}$ p-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$, that is, every walk between a node in $\mathbf{X}$ and a node in $\mathbf{Y}$ contains a non-collider node in $\mathbf{Z}$, or a collider node that is not in $\mathbf{Z}$.*

Here again, p-separation for AMP-CGs reduces to u-separation in the case of an UG, and d-separation in the case of a DAG. Furthermore, the use of walks in p-separation is not problematic, as Levitz et al. [LPM01] shows that the induced independence model can be recovered in linear time with respect to the number of nodes and edges. As for conditional independence properties, it was shown by Levitz et al. [LPM01] that every AMP-CG independence model is probabilistic.

cond. closure graph

The factorization of p according to an AMP-CG model was recently given by Peña [Peñ15][addendum Theorem 1], and relies on a specific notion which we call *conditional closure graph*. Given an AMP chain graph $\mathcal{G}$ and a chain component $\mathbf{K}$, the conditional closure graph $\mathcal{L}(\mathcal{G}, \mathbf{S})$ of a subset $\mathbf{S} \subseteq \mathbf{K}$ is obtained as follows: i) take $\mathcal{H}$ the induced subgraph of $\mathcal{G}$ over the node set $\mathbf{S}$; ii) add an undirected edge between each pair of nodes in $\mathbf{S}$ if they accept a path in $\mathcal{G}$ made only of intermediate nodes in $\mathbf{K} \setminus \mathbf{S}$.

AMP-CG prob. model

Def. 2.10 *An AMP-CG model consists in a set of random variables $\mathbf{V} = \{V_1, \ldots, V_n\}$, a chain graph $\mathcal{G} = (\mathbf{V}, \mathcal{E})$, and a set of parameters Θ. Together, $\mathcal{G}$ and Θ define a probability distribution p over $\mathbf{V}$ which factorizes as*

$$p(\mathbf{v}) = \prod_{\mathbf{K}_i \in \mathcal{C}c_{\mathcal{G}}} p(\mathbf{k}_i | \mathbf{pa}_{\mathbf{K}_i})$$

where $\mathcal{C}c_{\mathcal{G}}$ is the set of all chain components in $\mathcal{G}$, and for every subset $\mathbf{S} \subseteq \mathbf{K}_i$, the conditional probability $p(\mathbf{s}|\mathbf{pa}_{\mathbf{K}_i})$ factorizes as

$$p(\mathbf{s}|\mathbf{pa}_{\mathbf{K}_i}) = \prod_{\mathbf{C}_j \in \mathcal{C}l_{\mathcal{L}(\mathcal{G},\mathbf{S})}} \phi_j(\mathbf{c}_j, \mathbf{pa}_{\mathbf{C}_j})$$

where $\mathcal{C}l_{\mathcal{L}(\mathcal{G},\mathbf{S})}$ is the set of all cliques in the conditional closure graph $\mathcal{L}(\mathcal{G}, \mathbf{S})$, and ϕ_j is a positive function.

The above factorization appears more complex than that of LWF-CG models, since it involves several factorizations over marginals of $p(\mathbf{k}_i|\mathbf{pa}_{\mathbf{K}_i})$, for each chain component. Still, this factorization reduces to the clique factorization of a Markov network in the case of a fully undirected graph, and the recursive factorization of a Bayesian network in the case of a fully directed graph. As for soundness, it was shown by Peña [Peñ15] that, with $\mathcal{G}$ an AMP-CG, $I(\mathcal{G})$ is an I-map for p if p factorizes according to $\mathcal{G}$, and the converse holds if $p > 0$.

AMP-CGs, just like LWF-CGs, generalize both UGs and DAGs, but not BGs. However, both CG interpretations produce a different family of independence models, and no interpretation subsumes the other.

MVR chain graphs

A third interpretation of chain graphs as independence models comes from Cox and Wermuth [CW93; CW96]. Such models are commonly called MVR-CG models, as they were initially proposed as models of multivariate regression.

The new separation criterion for a MVR chain graphs is called m-separation [RS02], and is based on the notion of collider nodes. In an MVR chain graph, a *collider node* in a walk is any intermediate node V_i that follows one of the following patterns: $\rightarrow V_i \leftarrow$, $\rightarrow V_i -$ or $-V_i -$. Note that this definition may also include symmetric patterns as m-separation is symmetric. Similarly to AMP-CGs, the notion of collider nodes in MVR-CGs can be seen as an extension of colliders in DAGs.

collider node

Def. 2.11 *MVR-CG ind. model* *Given an MVR-CG $\mathcal{G}$, $I(\mathcal{G})$ is the independence model such that a disjoint triplet $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$ belongs to $I(\mathcal{G})$* iff $\mathbf{Z}$ *m-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$, that is, every walk between a node in $\mathbf{X}$ and a node in $\mathbf{Y}$ contains a non-collider node in $\mathbf{Z}$, or a collider node that is not in $\mathbf{Z}$.*

Unlike LWF and AMP chain graphs, m-separation for MVR-CGs does not reduce to u-separation in the case of an UG. However, it reduces to d-separation in the case of a DAG, and b-separation in the case of a BG. As for conditional independence

properties, it can be found in Richardson and Spirtes [RS02][Theorem 7.5] that every MVR-CG independence model is probabilistic.

The factorization of p according to an MVR-CG model is given by Drton [Drt09][Theorem 8].

Def. 2.12 *MVR-CG prob. model*

An MVR-CG model consists in a set of random variables $\mathbf{V} = \{V_1, \ldots, V_n\}$, a chain graph $\mathcal{G} = (\mathbf{V}, \mathcal{E})$, and a set of parameters Θ. Together, $\mathcal{G}$ and Θ define a probability distribution p over $\mathbf{V}$ which factorizes as

$$p(\mathbf{v}) = \prod_{\mathbf{K}_i \in \mathcal{C}c_{\mathcal{G}}} p(\mathbf{k}_i | \mathbf{pa}_{\mathbf{K}_i})$$

where $\mathcal{C}c_{\mathcal{G}}$ is the set of all chain components in $\mathcal{G}$, and for every subset $\mathbf{S} \subseteq \mathbf{K}_i$, the conditional probability $p(\mathbf{s}|\mathbf{pa}_{\mathbf{K}_i})$ factorizes as

$$p(\mathbf{s}|\mathbf{pa}_{\mathbf{K}_i}) = \prod_{\mathbf{C}_j \in \mathcal{C}m_{\mathcal{G}_{\mathbf{S}}}} p(\mathbf{c}_j | \mathbf{pa}_{\mathbf{C}_j})$$

where $\mathcal{C}m_{\mathcal{G}_{\mathbf{S}}}$ indexes the set of maximal connected sets in $\mathcal{G}_{\mathbf{S}}$ the induced subgraph of $\mathcal{G}$ over $\mathbf{S}$.

Interestingly, the above factorization for MVR chain graphs reduces to the factorization of a bi-directed graph in the case of a fully undirected graph. Not surprisingly, it also reduces to the recursive factorization of a Bayesian network in the case of a fully directed graph. As for soundness, it was shown by Drton [Drt09] that, with $\mathcal{G}$ an MVR-CG, $I(\mathcal{G})$ is an I-map for p *iff* p factorizes according to $\mathcal{G}$. Interestingly, under the MVR interpretation, the positivity condition $p > 0$ of LWF and AMP interpretations is not required for the converse implication to hold.

As MVR-CGs generalize both BGs and DAGs but not UGs, it is common practice to represent the undirected edges in an MVR-CG with arcs $\leftrightarrow$ instead of lines $-$. Again, MVR chain graphs produce a different set of independence models than AMP and LWF chain graphs, and no interpretation subsumes the other.

A unified view

We will now attempt to give a unified definition of the factorization of a probability distribution p according to a chain graph $\mathcal{G}$, which generalizes the LWF, AMP and MVR factorizations. This re-formulation extends the notion of closure graph that was introduced for LWF chain graphs to AMP and MVR chain graphs. As we will see, the difference between each factorization seems to come only from the differences in the structure of these closure graphs, which in turn comes only from differences

in the independence model $I(\mathcal{G})$. Note that we give no formal proof of our claim, and therefore the results we give here are presented as conjectures.

closure graph

Given a chain graph $\mathcal{G}$ and a chain component $\mathbf{K}$, every subset $\mathbf{S} \subseteq \mathbf{K}$ defines a *closure graph* $\mathcal{H}(\mathcal{G}, \mathbf{S})$, which is obtained as follows: i) take $\mathcal{G}_{\mathbf{K}}$ the induced subgraph of $\mathcal{G}$ over $\mathbf{K} \cup \mathbf{PA}_{\mathbf{K}}$; ii) remove any edge in $\mathcal{G}_{\mathbf{K}}$ between each pair of nodes in $\mathbf{PA}_{\mathbf{K}}$; iii) take $\mathcal{H}$ the undirected graph over $\mathbf{S} \cup \mathbf{PA}_{\mathbf{K}}$ that contains an edge between two nodes V_i, V_j *iff* $\langle V_i, V_j \mid (\mathbf{S} \cup \mathbf{PA}_{\mathbf{K}}) \setminus \{V_i, V_j\} \rangle \notin I(\mathcal{G}_{\mathbf{K}})$; iv) remove any node in $\mathcal{H}$ that is not connected to a node in $\mathbf{S}$.

Def. 2.13 *CG prob. model* *An CG model consists in a set of random variables $\mathbf{V} = \{V_1, \ldots, V_n\}$, a chain graph $\mathcal{G} = (\mathbf{V}, \mathcal{E})$, and a set of parameters Θ. Together, $\mathcal{G}$ and Θ define a probability distribution p over $\mathbf{V}$ which factorizes as*

$$p(\mathbf{v}) = \prod_{\mathbf{K}_i \in \mathcal{C}c_{\mathcal{G}}} p(\mathbf{k}_i | \mathbf{pa}_{\mathbf{K}_i}),$$

where $\mathcal{C}c_{\mathcal{G}}$ is the set of all chain components in $\mathcal{G}$, and for every subset $\mathbf{S} \subseteq \mathbf{K}_i$, the conditional probability $p(\mathbf{s}|\mathbf{pa}_{\mathbf{K}_i})$ factorizes as

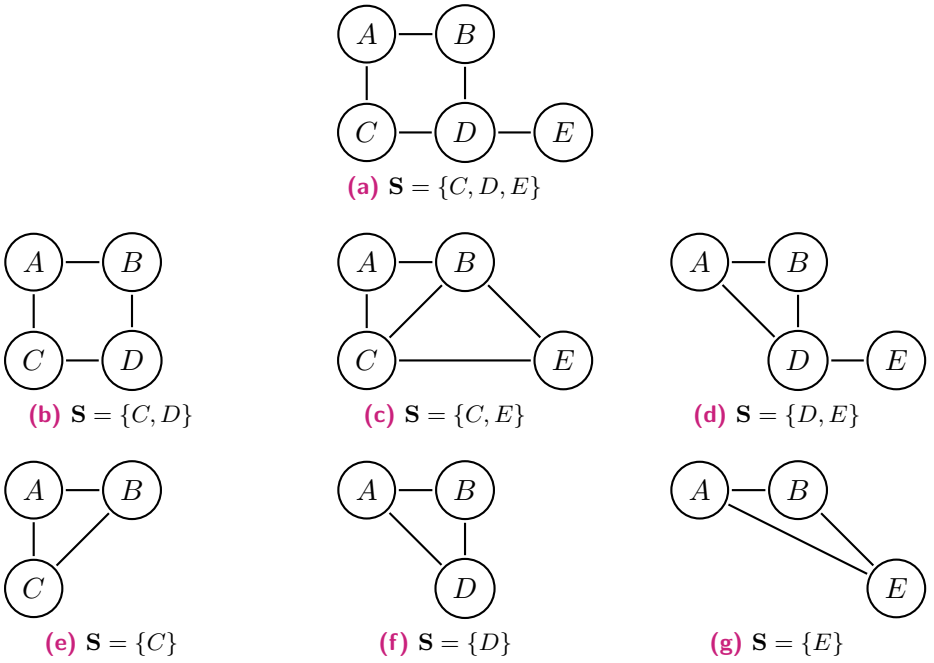
$$p(\mathbf{s}|\mathbf{pa}_{\mathbf{K}_i}) = \prod_{\mathbf{C}_j \in \mathcal{C}l_{\mathcal{H}(\mathcal{G},\mathbf{S})}} \phi_j(\mathbf{c}_j),$$

where $\mathcal{C}l_{\mathcal{H}(\mathcal{G}, \mathbf{S} \cup \mathbf{PA}_{\mathbf{K}_i})}$ is the set of all cliques in the closure graph $\mathcal{H}(\mathcal{G}, \mathbf{S})$, and ϕ_j is a positive function.

The first factorization is common to all chain graph interpretations, and can be seen as a consequence of a recursive factorization over the chain components, similarly to that of DAGs. The second factorization imposes a set of constraints on the conditional probability distribution of each chain component given its parents, which depends only on the structure of the closure graph $\mathcal{H}(\mathcal{G}, \mathbf{S})$. This is where arises the difference between each chain graph interpretation, as the structure of the closure graph is given by the independence model $I(\mathcal{G})$. We believe that the three following conjectures hold. We do not have any formal proof for these, however we will provide some motivational arguments.

Conj. 2.6 *With $\mathcal{G}$ an LWF chain graph, Definition 2.13 reduces to Definition 2.8.*

From the c-separation criterion, the closure graph $\mathcal{H}(\mathcal{G}, \mathbf{S})$ is obtained as follows: i) take $\mathcal{H}$ the induced subgraph of $\mathcal{G}$ over the node set $\mathbf{S} \cup \mathbf{PA}_{\mathbf{K}}$; ii) add an edge between each pair of nodes in $\mathbf{PA}_{\mathbf{K}}$; iii) add an edge between each pair of nodes that accepts a path in $\mathcal{G}$ made only of intermediate nodes in $\mathbf{K} \setminus \mathbf{S}$; iv) make each edge undirected.



(a) $\mathbf{S} = \{C, D, E\}$ (b) $\mathbf{S} = \{C, D\}$ (c) $\mathbf{S} = \{C, E\}$ (d) $\mathbf{S} = \{D, E\}$ (e) $\mathbf{S} = \{C\}$ (f) $\mathbf{S} = \{D\}$ (g) $\mathbf{S} = \{E\}$

Fig. 2.10. The closure graphs under the LWF interpretation, for the chain component $\mathbf{K} = \{C, D, E\}$ from Figure 2.9.

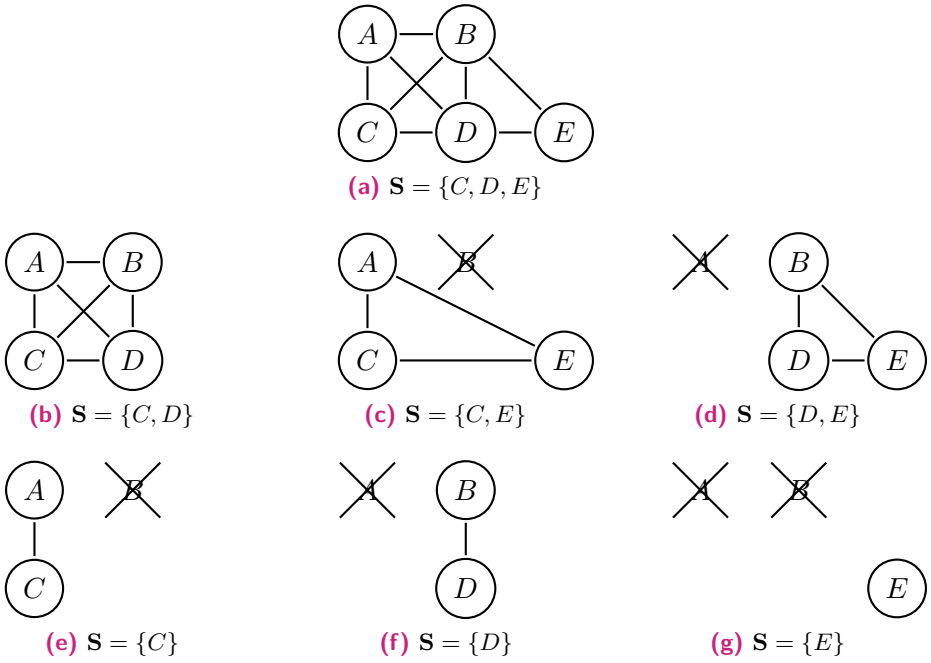
When $\mathbf{S} = \mathbf{K}$, the closure graph defined above results in the closure graph from Definition 2.8. When $\mathbf{S} \subset \mathbf{K}$ it appears that no additional constraint arises from the factorization, so only the subset $\mathbf{S} = \mathbf{K}$ may be considered. In the end we obtain the LWF-CG factorization from Definition 2.8.

Conj. 2.7 *With $\mathcal{G}$ an AMP chain graph, Definition 2.13 reduces to Definition 2.10.*

From the p-separation criterion, the closure graph $\mathcal{H}(\mathcal{G}, \mathbf{S})$ is obtained as follows: i) take $\mathcal{H}$ the induced subgraph of $\mathcal{G}$ over the node set $\mathbf{S} \cup \mathbf{PA_S}$; ii) add an edge between each pair of nodes in $\mathbf{S}$ if they accept a path in $\mathcal{G}$ made only of intermediate nodes in $\mathbf{K} \setminus \mathbf{S}$; iii) for each clique $\mathbf{C} \subseteq \mathbf{S}$ in $\mathcal{H}$, add an edge between every pair of nodes in $\mathbf{PA_C}$, and between every node in $\mathbf{C}$ and every node in $\mathbf{PA_C}$; iv) make each edge undirected.

It appears that the conditional closure graph from Definition 2.10 matches the induced subgraph over $\mathbf{S}$ of the closure graph $\mathcal{H}(\mathcal{G}, \mathbf{S})$ (step ii). Moreover, for every clique $\mathbf{C} \in \mathbf{S}$ in $\mathcal{H}$, it appears that $\mathbf{C} \cup \mathbf{PA_C}$ is also a clique (step iii), and for every clique $\mathbf{C} \in \mathbf{PA_S}$ it appears that $\mathbf{C} \cup \mathbf{CH_C}$ is also a clique. Without loss of generality, let us re-define the chain component factorization in Definition 2.13 with potentials over maximal cliques, we obtain the AMP-CG factorization from Definition 2.10.

Conj. 2.8 *With $\mathcal{G}$ an MVR chain graph, Definition 2.13 reduces to Definition 2.12.*



(a) $\mathbf{S} = \{C, D, E\}$ (b) $\mathbf{S} = \{C, D\}$ (c) $\mathbf{S} = \{C, E\}$ (d) $\mathbf{S} = \{D, E\}$ (e) $\mathbf{S} = \{C\}$ (f) $\mathbf{S} = \{D\}$ (g) $\mathbf{S} = \{E\}$

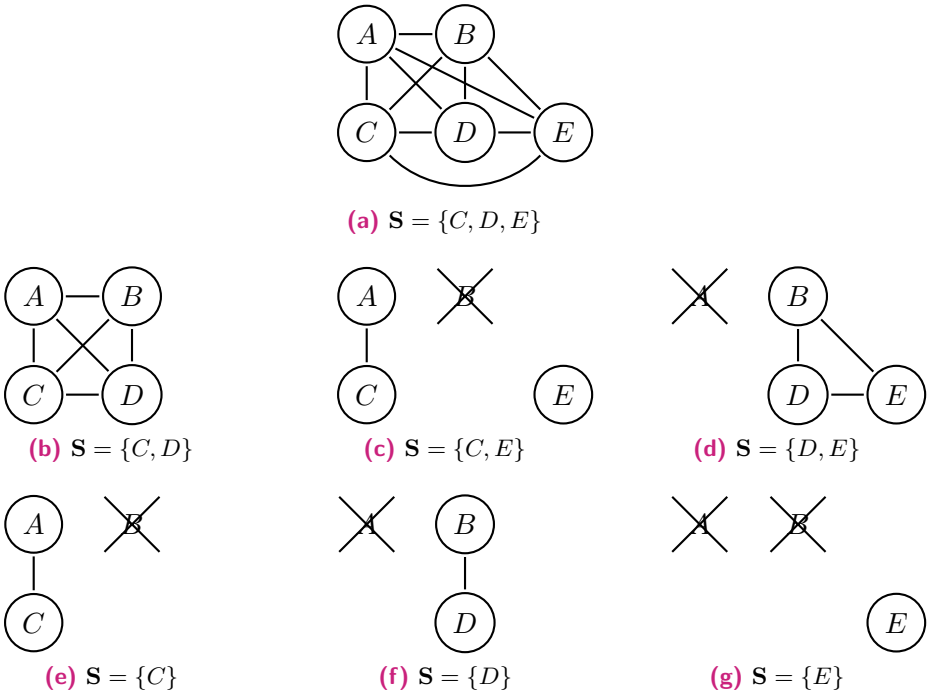
Fig. 2.11. The closure graphs under the AMP interpretation, for the chain component $\mathbf{K} = \{C, D, E\}$ from Figure 2.9.

From the m-separation criterion, the closure graph $\mathcal{H}(\mathcal{G}, \mathbf{S})$ is obtained as follows: i) take $\mathcal{H}$ the induced subgraph of $\mathcal{G}$ over the node set $\mathbf{S} \cup \mathbf{PA_S}$; ii) add an edge between each pair of nodes in $\mathbf{S}$ if they accept a path in $\mathcal{G}$ made only of intermediate nodes in $\mathbf{S}$; iii) for each clique $\mathbf{C} \in \mathbf{S}$ in $\mathcal{H}$, add an edge between every pair of nodes in $\mathbf{PA_C}$, and between every node in $\mathbf{C}$ and every node in $\mathbf{PA_C}$; iv) make each edge undirected.

Let us define $\mathcal{H}_\mathbf{S}$ the induced subgraph over $\mathbf{S}$ of the closure graph $\mathcal{H}(\mathcal{G}, \mathbf{S})$. It appears that the cliques in $\mathcal{H}_\mathbf{S}$ correspond to the maximal connected sets in the induced subgraph $\mathcal{G}_\mathbf{S}$ from Definition 2.10 (step ii). Moreover, for each of the cliques $\mathbf{C} \in \mathbf{S}$ in $\mathcal{H}$, it appears that $\mathbf{C} \cup \mathbf{PA_C}$ is also a clique (step iii), and for every clique $\mathbf{C} \in \mathbf{PA_S}$ it appears that $\mathbf{C} \cup \mathbf{CH_C}$ is also a clique. Without loss of generality, let us re-define the chain component factorization in Definition 2.13 with potentials over maximal cliques. It appears that all these maximal cliques are mutually disjoint. Because of that the clique potentials can be expressed as conditional probabilities, and we obtain the MVR-CG factorization from Definition 2.12.

A fourth chain graph interpretation ?

According to Drton [Drt09], a fourth interpretation of chain graphs as independence models exists, which is the dual of the AMP interpretation. To the best of our knowledge, this fourth interpretation as not been studied yet in the literature. However, we



(a) $\mathbf{S} = \{C, D, E\}$ (b) $\mathbf{S} = \{C, D\}$ (c) $\mathbf{S} = \{C, E\}$ (d) $\mathbf{S} = \{D, E\}$ (e) $\mathbf{S} = \{C\}$ (f) $\mathbf{S} = \{D\}$ (g) $\mathbf{S} = \{E\}$

Fig. 2.12. The closure graphs under the MVR interpretation, for the chain component $\mathbf{K} = \{C, D, E\}$ from Figure 2.9.

will give a possible separation criterion and factorization under this interpretation, and conjecture that it is complete (the independence model is probabilistic, and is a sufficient and necessary condition for the factorization). Let us call this new family of models the Type 3 chain graph models (T3-CGs), in agreement with the categorization in [Drt09].

d-collider section

u-collider section

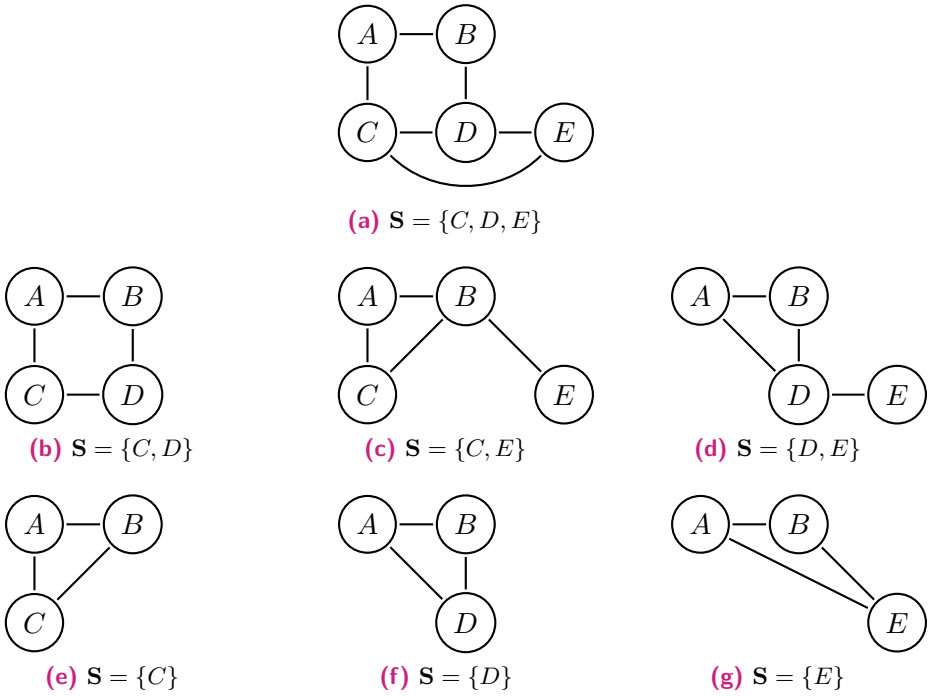
The separation criterion for a T3 chain graph, which we call $c3$-separation, is based on the notions of d-collider sections, which is that of collider sections in LWF-CGs, and u-collider sections, which somewhat extends that of collider nodes in BGs. In a T3 chain graph, a *d-collider section* is any section $\rho = (V_i, \ldots, V_j)$ that follows the following pattern in the walk: $\rightarrow \rho \leftarrow$. Likewise, a *$u$-collider section* is any section that follows the following pattern: $-\rho-$. [2]

Def. 2.14 *T3-CG ind. model*

Given a T3-CG $\mathcal{G}$, $I(\mathcal{G})$ is the independence model such that a disjoint triplet $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$ belongs to $I(\mathcal{G})$ iff *$\mathbf{Z}$ $c3$-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$, that is, every walk between a node in $\mathbf{X}$ and a node in $\mathbf{Y}$ contains a non-collider section that has a node in $\mathbf{Z}$, or a d-collider section that has no node in $\mathbf{Z}$, or a u-collider section that has a node not in $\mathbf{Z}$.*

From the $c3$-separation criterion, given a T3 chain graph $\mathcal{G}$, a chain component $\mathbf{K}$ and a subset $\mathbf{S} \subseteq \mathbf{K}$, the closure graph $\mathcal{H}(\mathcal{G}, \mathbf{S})$ is obtained as follows: i) take $\mathcal{H}$ the induced subgraph of $\mathcal{G}$ over the node set $\mathbf{S} \cup \mathbf{PA_K}$; ii) add an edge between each

[2] Note that since walk sections are maximal undirected sequences, a u-collider section can only appear in a completely undirected walk, and consists in the entire walk expect for the first and last nodes.



(a) $\mathbf{S} = \{C, D, E\}$ (b) $\mathbf{S} = \{C, D\}$ (c) $\mathbf{S} = \{C, E\}$ (d) $\mathbf{S} = \{D, E\}$ (e) $\mathbf{S} = \{C\}$ (f) $\mathbf{S} = \{D\}$ (g) $\mathbf{S} = \{E\}$

Fig. 2.13. The closure graphs under our T3 interpretation, for the chain component $\mathbf{K} = \{C, D, E\}$ from Figure 2.9.

pair of nodes in $\mathbf{PA_K}$; iii) add an edge between each pair of nodes in $\mathbf{S}$ that accepts a path in $\mathcal{G}$ made only of intermediate nodes in $\mathbf{S}$; iv) add an edge between each node in $\mathbf{S}$ and each node in $\mathbf{PA_K}$ that accepts a path in $\mathcal{G}$ made only of intermediate nodes in $\mathbf{K} \setminus \mathbf{S}$; v) make each edge undirected.

The factorization of p according to a T3-CG follows from Definition 2.13. Again, we give no formal proof that this new separation criterion for chain graphs corresponds to the Type 3 interpretation from Drton [Drt09]. However, we found that it was interesting to mention it, as it seems to complete nicely the family of chain graph models. As for soundness, we propose the two following conjectures.

Conj. 2.9 *With $\mathcal{G}$ a T3-CG, $I(\mathcal{G})$ is a probabilistic independence model.*

Conj. 2.10 *With $\mathcal{G}$ a T3-CG, $I(\mathcal{G})$ is an I-map for p* iff *p factorizes according to $\mathcal{G}$.*

We plan to answer Conjectures 2.9 and 2.10 in the near future. If these are answered positively, then the above separation criterion could define a new family of probabilistic graphical models based on chain graphs.

Discussion

To summarize, chain graphs were originally proposed as a new probabilistic graphical model to unify and extend directed and undirected graphical models. However, it

appears that three consistent chain graph interpretations exist, maybe even four. The LWF and AMP chain graphs subsume both DAG and UG models, but not BG models. On the other hand, MVR (and T3?) chain graphs subsume both DAG and BG models, but not UG models. Moreover, no chain graph interpretation subsumes another [SP15]. In the end, the so-called *hybrid graphs* suggested by Verma and Pearl [VP88] appeared to yield richer but also much more complex models than what was originally suggested.

To the best of our knowledge, no axiomatic characterization of chain graph models in terms of conditional independence properties exists in the literature. However, according to Sadeghi and Lauritzen [SL15] the independence model of an LWF, AMP or MVR chain graph is always a compositional graphoid.

2.2.3 Mixed graphs

In this section, we dig further into advanced graphical models and briefly review some important families of graphical models combining more than two types of edges, i.e. mixed graph models.

To understand the motivation behind the models discussed in this section, let us introduce the notions of marginal and conditional independence model, as defined by Sadeghi [Sad12].

Marginal and conditional models

Def. 2.15 *Consider an independence model I over a set $\mathbf{V}$. For any subset $\mathbf{M} \subseteq \mathbf{V}$, the* independence model after marginalization *over $\mathbf{M}$, denoted by $I^{\emptyset}_{\mathbf{M}}$, is the subset of I whose triples do not contain members of $\mathbf{M}$, i.e.*

marg. ind. model

$$I^{\emptyset}_{\mathbf{M}} = \{\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \in I \mid (\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}) \cap \mathbf{M} = \emptyset\}.$$

cond. ind. model *For any subset $\mathbf{C} \subseteq \mathbf{V}$, the* independence model after conditioning *on $\mathbf{C}$, denoted by $I^{\mathbf{C}}_{\emptyset}$, is*

$$I^{\mathbf{C}}_{\emptyset} = \{\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \mid \langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{C} \rangle \in I \text{ and } (\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}) \cap \mathbf{C} = \emptyset\}.$$

Combining these definitions, for disjoint subsets $\mathbf{M}$ and $\mathbf{C}$ of $\mathbf{V}$, the independence model after marginalization over $\mathbf{M}$ and conditioning on $\mathbf{C}$ is

$$I^{\mathbf{C}}_{\mathbf{M}} = \{\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle \mid \langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \cup \mathbf{C} \rangle \in I \text{ and } (\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}) \cap (\mathbf{C} \cup \mathbf{M}) = \emptyset\}.$$

One can observe that $I^{\mathbf{C}}_{\mathbf{M}}$ is an independence model over $\mathbf{V} \setminus (\mathbf{C} \cup \mathbf{M})$.

Let us now consider marginalization and conditioning for a probability distribution p defined over $\mathbf{V}$. The process of marginalizing out and conditioning on respectively two distinct subsets $\mathbf{M}$ and $\mathbf{C}$ yields a new probability distribution p' defined over $\mathbf{W} = \mathbf{V} \setminus (\mathbf{M} \cup \mathbf{C})$. The idea of marginalizing out $\mathbf{M}$ is quite straightforward. When $\mathbf{C} = \emptyset$, the resulting distribution is simply

$$p'(\mathbf{w}) = \sum_{\mathbf{m}} p(\mathbf{w}, \mathbf{m}).$$

latent variables

In this new distribution the variables in $\mathbf{M}$ are not observed, and are called *latent variables*. The idea of conditioning on $\mathbf{C}$ appears a little bit less intuitive. It consists in conditioning p on a certain event that depends on $\mathbf{C}$, then marginalizing out $\mathbf{C}$. When $\mathbf{M} = \emptyset$, the resulting distribution can be expressed as

$$p'(\mathbf{w}) = \sum_{\mathbf{c}} p(\mathbf{w}|\mathbf{c}) p^s(\mathbf{c}),$$

selection bias

with p^s any probability distribution over $\mathbf{C}$, called the *selection bias*. The variables in $\mathbf{C}$ are also called latent variables since they are not observed in p'. Typically, when the selection bias is $p^s(\mathbf{c}) = 1$ for a particular value $\mathbf{c}^s$ and 0 otherwise, the resulting distribution is simply $p'(\mathbf{w}) = p(\mathbf{w}|\mathbf{c} = \mathbf{c}^s)$. When combining marginalization and conditioning, the resulting distribution is

$$p'(\mathbf{w}) = \sum_{\mathbf{m}} \sum_{\mathbf{c}} p(\mathbf{w}, \mathbf{m}|\mathbf{c}) p^s(\mathbf{c}).$$

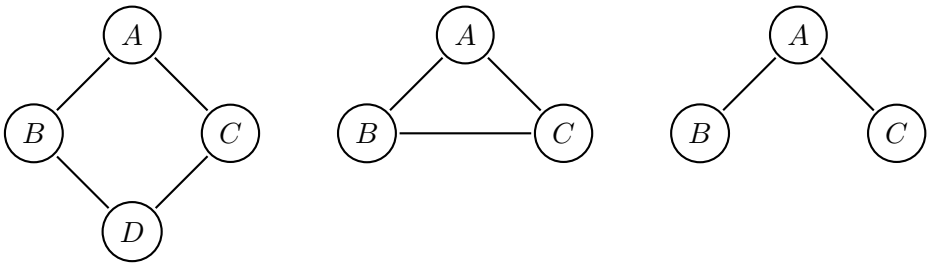
It can be shown that if an independence model I is an I-map for a distribution p, then $I^{\mathbf{C}}_{\mathbf{M}}$ is an I-map for the distribution p' resulting from marginalizing out $\mathbf{M}$ and conditioning on $\mathbf{C}$ under any selection bias.

Stability of graphical models

A family of graphical models is said *stable* under marginalization and conditioning *iff* for every graph $\mathcal{G}$ of this family and every distinct subsets $\mathbf{M}$ and $\mathbf{C}$ of $\mathbf{V}$, there exists a graph $\mathcal{G}'$ of the same family such that $I(\mathcal{G})^{\mathbf{C}}_{\mathbf{M}} = I(\mathcal{G}')$.

To illustrate the notion of stability, let us consider the families of graphical models we have encountered so far. The family of UGs is known to be stable under both marginalization and conditioning. For any UG $\mathcal{G}$, it suffices to add an edge between

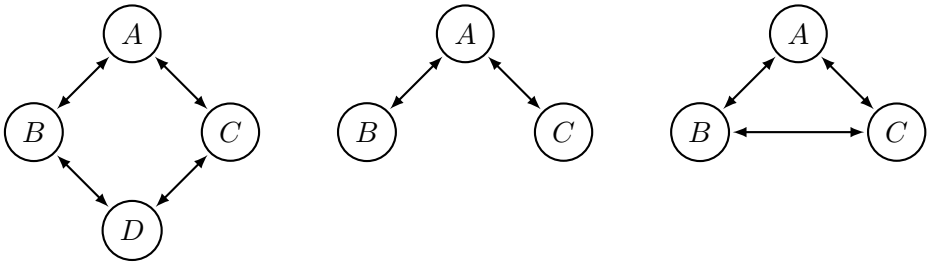
every pair of nodes that are adjacent to a common node in $\mathbf{M}$, and then remove the nodes in $\mathbf{M} \cup \mathbf{C}$ to obtain the desired graph $\mathcal{G}'$ (see Figure 2.14).



(a) $\mathcal{G}_1$ **(b)** $\mathcal{G}_2$, D marginalized out. **(c)** $\mathcal{G}_3$, D conditioned on.

Fig. 2.14. Three UGs $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ such that $I(\mathcal{G}_2) = I(\mathcal{G}_1)^{\emptyset}_{D}$ and $I(\mathcal{G}_3) = I(\mathcal{G}_1)^{D}_{\emptyset}$.

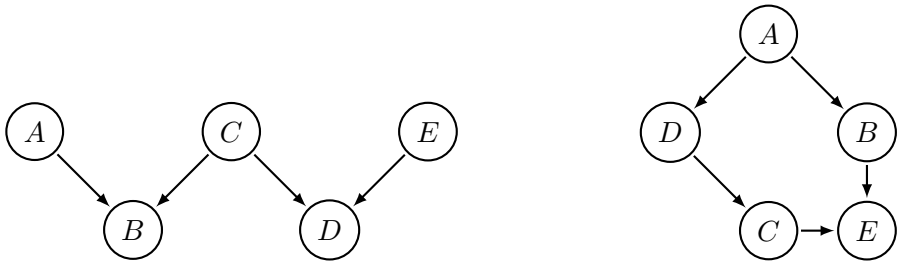
The family of BGs is also stable under marginalization and conditioning. For any BG $\mathcal{G}$, one can simply add an edge between every pair of nodes that are adjacent to a common node in $\mathbf{C}$, then remove the nodes in $\mathbf{M} \cup \mathbf{C}$ (see Figure 2.15).



(a) $\mathcal{G}_1$ **(b)** $\mathcal{G}_2$, D marginalized out. **(c)** $\mathcal{G}_3$, D conditioned on.

Fig. 2.15. Three BGs $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ such that $I(\mathcal{G}_2) = I(\mathcal{G}_1)^{\emptyset}_{D}$ and $I(\mathcal{G}_3) = I(\mathcal{G}_1)^{D}_{\emptyset}$.

The family of DAGs however, is stable neither under marginalization nor conditioning. Consider the two DAGs in Figure 2.16, the independence model of the first DAG after marginalizing out C can not be represented by any DAG, and neither can be the independence model of the second DAG after conditioning on E.



(a) $I(\mathcal{G})^{\emptyset}_{C}$ is not DAG-faithful. **(b)** $I(\mathcal{G})^{E}_{\emptyset}$ is not DAG-faithful.

Fig. 2.16. Two DAGs which are not stable under conditioning and marginalization.

Regarding chain graphs, it is known that the family of LWF CGs is stable under conditioning but not under marginalization [Sad16], while the family of MVR CGs is

stable under marginalization but not under conditioning. As for the family of AMP CGs, it seems that these are stable neither under marginalization nor conditioning.

Since several families of graphical models appear to be unstable under marginalization or conditioning, a question arises naturally: can we find a "super"-family of graphical model that captures these marginal and conditional independence models? As we will see, this question was answered positively for DAGs, LWF CGs, and possibly AMP CGs.

Ancestral graphs

According to [RS02], the problem of constructing graphical representations for the independence structure of DAGs under marginalization and conditioning was originally posed by Nanny Vermuth in 1994 in a lecture at CMU[3] (Carnegie-Mellon University). This led to the development of the so-called summary graphs (SGs) by Wermuth et al. [WCP94] and Wermuth [Wer11], the ancestral graphs (AGs) by Richardson and Spirtes [RS02] and the MC-graphs (MCGs) by Koster [Kos02]. All three families are based on mixed graphs with three types of edges: directed $\rightarrow$, undirected $-$ and bi-directed $\leftrightarrow$, and are stable under marginalization and conditioning. Roughly speaking, in such graphs the undirected edges are created by conditioning upon collider nodes, whereas bi-directed edges are created by marginalizing out non-collider nodes.

Interestingly, the separation criterion for the three families is the same [RS02]. It relies on the notion of collider nodes, which is actually the same as the one for MVR chain graphs if we consider the undirected edges in MVR-CGs as arcs $\leftrightarrow$ instead of lines $-$. Within the context of MCGs, SGs and AGs, a *collider node* in a walk is any intermediate node V_i that follows one of the following patterns: $\rightarrow V_i \leftarrow$, $\rightarrow V_i \leftrightarrow$ or $\leftrightarrow V_i \leftrightarrow$. The separation criterion for summary, ancestral and MC graphs is then also called m-separation.

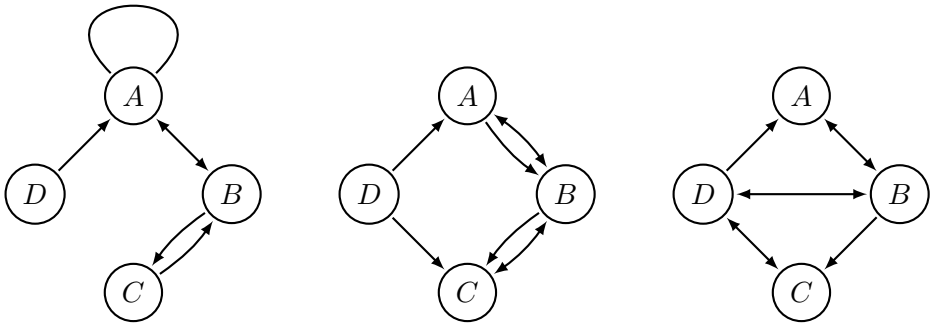
collider node

Def. 2.16 *MCG-SG-AG ind. model*

Given a MCG, SG or AG $\mathcal{G}$, $I(\mathcal{G})$ is the independence model such that a disjoint triplet $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$ belongs to $I(\mathcal{G})$ iff $\mathbf{Z}$ *m-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$, that is, every walk between a node in $\mathbf{X}$ and a node in $\mathbf{Y}$ contains a non-collider node in $\mathbf{Z}$, or a collider node that is not in $\mathbf{Z}$.*

Because they share the same separation criterion, the only difference between MC, summary and ancestral graph models comes from restrictions on the structure of the graph $\mathcal{G}$.

[3] Note that a similar idea is discussed in [VP90].



(a) An MC graph. **(b)** A summary graph. **(c)** An ancestral graph.

Fig. 2.17. Three mixed graphs which represent exactly the same independence model according to the m-separation criterion.

MC graph

An *MC graph* is a mixed graph with three types of edges ($-$, $\rightarrow$, $\leftrightarrow$), without any additional constraints. An MC-graph may then contain loops, directed cycles, and multiple edges. However, due to the m-separation criterion, only loops consisting of undirected edges $V_1 - V_1$ matter in the graph, as well as only multiple edges consisting in distinct edges (i.e. at most four edges $V_1 \leftarrow V_2$, $V_1 \rightarrow V_2$, $V_1 - V_2$ and $V_1 \leftrightarrow V_2$ between each pair of nodes).

summary graph

A *summary graph* is actually an MC graph with three additional constraints: i) no loop; ii) no directed cycle; iii) the endpoints of an undirected edge $-$ have no parent or spouse node. Notice that, due to this last constraint, summary graphs can only have multiple edges made of one directed and one bi-directed edge.

ancestral graph

Finally, an *ancestral graph* is a summary graph with one additional constraint: the endpoints of a bi-directed edge $\leftrightarrow$ accept no directed path between them. As a consequence, multiple edges are no more allowed in the graph, and ancestral graphs are simple graphs.

As shown in [RS02], summary and ancestral graphs capture the same class of independence models, and match exactly the class of independence models that originate from a DAG after marginalization and conditioning. As for MC-graphs, they capture a broader class of independence models, which may not correspond to any DAG with marginalization and conditioning [RS02] (see Figure 2.18).

pairwise Markov prop.

An important desirable property of graphical independence models, sometimes called *pairwise Markov property*, is that the absence of an edge between two distinct nodes always translates an independence relation, that is, if V_1 and V_2 are non-adjacent in $\mathcal{G}$ then $\langle V_1, V_2 \mid \mathbf{Z} \rangle \in I(\mathcal{G})$ for some subset $\mathbf{Z} \subseteq \mathbf{V} \setminus \{V_1, V_2\}$. However, neither MC graphs, summary graphs or ancestral graphs respect this property. See for example Figure 2.17, in which A and C are not adjacent in any of the graphs, however there is no subset of $\{B, D\}$ that separates them. To solve this problem, Richardson

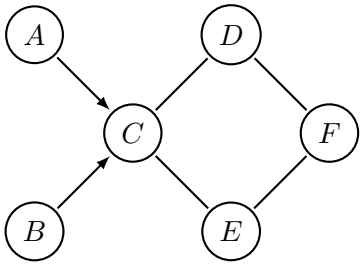


Fig. 2.18. An MC graph whose independence model $I(\mathcal{G})$ does not corresponds to any summary or ancestral graph model, and thereby to any conditional and marginal DAG model.

and Spirtes [RS02] introduces the family of maximal ancestral graphs (MAGs), i.e. ancestral graphs that respect the pairwise Markov property, and shows that every AG can be converted to a MAG without changing its independence model, simply by adding bi-directed edges to the graph. For example, in Figure 2.17c it suffices to add the edge $A \leftrightarrow C$.

Richardson and Spirtes [RS02] shows that ancestral graph models, and therefore summary graph models, are probabilistic, and gives a parameterization of MAG models in the Gaussian case (i.e. when p is assumed to obey a multivariate Gaussian distribution). On the other hand, MC-graph models were not shown to be probabilistic. To the best of our knowledge, no general factorization rule was given for any of these models. Nonetheless, Koster [Kos02] shows that MCG models (and therefore SG and AG models) are compositional graphoids.

To summarize, MCGs subsume AGs and SGs, which in turn subsume DAGs, UGs, BGs and MVR CGs. Maximal ancestral graph (MAG) models appear to exhibit many interesting properties, as they rely on simple graphs, they are probabilistic, and they respect a basic intuitive interpretation with pairwise Markov property. Nonetheless, MAGs (and SGs) do not subsume LWF or AMP chain graph models, as it happens that some of these do not correspond to any conditional and marginal DAG model. See for example the chain graph in Figure 2.9.

Anterial graphs

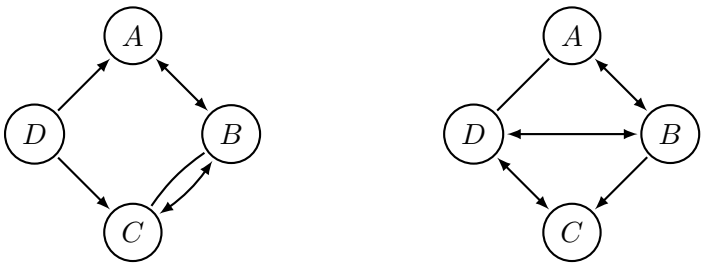
Sadeghi [Sad16] proposed the family of chain mixed graphs (CMGs), with directed $\rightarrow$, undirected $-$ and bi-directed $\leftrightarrow$ edges, to capture the marginal and conditional independence models of LWF-CGs. The separation criterion for CMGs, which we call cm-separation, extends that of LWF and MVR CGs. The notion of a section ρ is that of LWF-CGs, and the notion of a collider section is extended to include patterns from MVR-CGs: $\leftarrow \rho \leftrightarrow$ and $\leftrightarrow \rho \leftrightarrow$.

Def. 2.17 *CMG ind. model* *Given a CMG $\mathcal{G}$, $I(\mathcal{G})$ is the independence model such that a disjoint triplet $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$ belongs to $I(\mathcal{G})$* iff $\mathbf{Z}$ *cm-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$, that is, every walk between a node in $\mathbf{X}$ and a node in $\mathbf{Y}$ contains a non-collider section that has a node in $\mathbf{Z}$, or a collider section that has no node in $\mathbf{Z}$.*

Because the separation criterion for CMGs generalizes the separation criterion for LWF-CGs, the difference between CMG and LWF-CG models comes from restrictions on the structure of the graph $\mathcal{G}$.

chain mixed graph

A *chain mixed graph* is a mixed graph with three types of edges ($-$, $\rightarrow$, $\leftrightarrow$) that respects two constraints: i) no loops; ii) no semi-directed cycles unless they contain an arc $\leftrightarrow$. Note that, due these constraints, CMGs can have multiple edges consisting in arcs and arrows ($V_1 \leftrightarrow V_2$ and $V_1 \rightarrow V_2$) or arcs and lines ($V_1 \leftrightarrow V_2$ and $V_1 - V_2$). Moreover, due to the cm-separation criterion, only multiple edges consisting in distinct edges matter in the graph.



(a) A chain mixed graph. **(b)** An anterial graph.

Fig. 2.19. Two mixed graphs which represent exactly the same independence model according to the cm-separation criterion.

The resulting independence model for CMGs is stable under marginalization and conditioning, and captures LWF-CG and MVR-CG models as a subclass. As a result, CMG models capture the conditional and marginal models of LWF-CGs, MVR-CGs, DAGs, UGs and BGs, and thereby subsume both AGs (which correspond exactly to marginal and conditional DAG models).

However, CMGs may have multiple edges, and do not respect the pairwise Markov property. Sadeghi [Sad16] introduces the so-called anterial graphs (AnGs) which capture exactly the same independence models as CMGs.

anterial graph

An *anterial graph* is a chain mixed graph with one additional constraint: the endpoints of a bi-directed edge $\leftrightarrow$ accept neither a semi-directed path or an undirected path (only lines $-$) between them. As a result, anterial graphs are simple graphs.

It appears that AnGs share the same relationship with CMGs as AGs do with SGs, that is, they capture the same class of independence models with a simpler structure.

However, AnGs do not respect the pairwise Markov property, but it is argued in [Sad16] that it is always possible to derive a maximal anterial graph (MAnG) from an AnG that encodes the same independence model, similarly to MAGs from AGs. Finally, MAnG models were shown to be compositional graphoids, but it was not proved whether they are probabilistic models or not.

MAMP chain graphs

Peña [Peñ14] proposed the family of marginal AMP chain graphs (MAMP-CGs), with directed $\rightarrow$, undirected $-$ and bi-directed $\leftrightarrow$ edges, which subsumes both AMP and MVR chain graph models. The separation criterion for MAMP-CGs, which we call pm-separation, extends that of AMP and MVR CGs. The notion of a collider node includes patterns from both interpretations, that is, a collider node in a walk is any intermediate node V_i that follows one of the following patterns: $\rightarrow V_i \leftarrow$, $\rightarrow V_i \leftrightarrow$, $\rightarrow V_i -$, $\leftrightarrow V_i \leftrightarrow$ or $\leftrightarrow V_i -$.

MAMP-CG ind. model

Def. 2.18 *Given an MAMP-CG $\mathcal{G}$, $I(\mathcal{G})$ is the independence model such that a disjoint triplet $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$ belongs to $I(\mathcal{G})$* iff *$\mathbf{Z}$ pm-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$, that is, every walk between a node in $\mathbf{X}$ and a node in $\mathbf{Y}$ contains a non-collider node in $\mathbf{Z}$, or a collider node that is not in $\mathbf{Z}$.*

MAMP chain graph

A *marginal AMP chain graph* is a mixed graph with three types of edges ($-$, $\rightarrow$, $\leftrightarrow$) that respects several constraints: i) no loops; ii) no multiple edges; iii) no semi-directed cycles[4]; iv) the endpoints of a bi-directed edge $\leftrightarrow$ accept no undirected path (only lines $-$) between them; v) if a node V is the endpoint of a bi-directed $\leftrightarrow$ edge then $\mathbf{NE}_V$ forms a clique.

The resulting independence model for MAMP-CGs captures AMP-CG and MVR-CG models as a subclass, but is not stable under marginalization and conditioning. As a result, MAMP-CG models subsume DAGs, UGs and BGs, but not AGs or AnGs. Note that AGs do not subsume MAMP-CGs either, since there are AMP-CGs that do not correspond to any AG.

Finally, Peña [Peñ14] shows that MAMP chain graphs are probabilistic models, and are compositional graphoids that satisfy the weak transitivity property. Another interesting property of MAMP-CGs is that they are simple graphs and they respect the pairwise Markov property. The general factorization of MAMP-CG models is not given.

[4] Recall that in the context of mixed graphs a semi-directed cycle may contain undirected $-$ and/or bi-directed $\leftrightarrow$ edges.

Acyclic graphs

Recently, Sadeghi and Lauritzen [SL15] proposed the family of acyclic graphs (ACGs), in an attempt to unify all know graphical representations of independence models to date. Quoting the authors, "the idea is to provide one type of edge for every type of chain graph discussed in the literature". ACGs have four types of edges, namely arrows $\rightarrow$, lines $-$, arcs $\leftrightarrow$ and dashes $\leftarrow\!\!-\!\!\rightarrow$. The lines, arcs and dashes respectively correspond to the undirected edges of LWF-CGs, MVR-CGs and AMP-CGs. We call the separation criterion for ACGs cpm-separation. The notion of a section ρ is that of LWF-CGs, that is, a maximal intermediate subwalk made of lines $-$ only. The notion of a collider section mixes up patterns from all the chain graph interpretations, that is: $\rightarrow \rho \leftarrow$, $\rightarrow \rho \leftrightarrow$, $\rightarrow \rho \leftarrow\!\!-\!\!\rightarrow$, $\leftrightarrow \rho \leftrightarrow$ or $\leftrightarrow \rho \leftarrow\!\!-\!\!\rightarrow$.

Def. 2.19 *ACG ind. model*

Given an ACG $\mathcal{G}$, $I(\mathcal{G})$ is the independence model such that a disjoint triplet $\langle \mathbf{X}, \mathbf{Y} \mid \mathbf{Z} \rangle$ belongs to $I(\mathcal{G})$ iff *$\mathbf{Z}$ cpm-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$, that is, every walk between a node in $\mathbf{X}$ and a node in $\mathbf{Y}$ contains a non-collider section that has a node in $\mathbf{Z}$, or a collider section that has no node in $\mathbf{Z}$.*

acyclic graph

An *acyclic graph* is a mixed graph with four types of edges ($\rightarrow$, $-$, $\leftrightarrow$, $\leftarrow\!\!-\!\!\rightarrow$) that respects two constraints: i) no loops; ii) no semi-directed cycles unless they contain an arc $\leftrightarrow$ or a dash $\leftarrow\!\!-\!\!\rightarrow$. Notice that, due to this last constraint, acyclic graphs can not have multiple edges with an arrow and a line, or with two arrows in opposite directions. Any other combination of edges is allowed though.

According to Sadeghi and Lauritzen [SL15], acyclic graphs generalize all graphical models discussed in the literature, except MC graphs. It has been shown that ACGs are compositional graphoids, and therefore all graphical models it subsumes are compositional graphoids. However, several properties of ACGs are still to be studied. Clearly ACGs do not respect the pairwise Markov property, and it is not known if and how an ACG can be turned into a maximal ACG. Moreover, ACGs have not yet been shown to be probabilistic, nor stable under marginalization and conditioning. And finally, no general factorization rule has been given for ACGs.

2.3 Discussion

The study of advanced graphical models is still an active area of research today, with a lot of different families of models present in the literature. Obviously our state-of-the-art does not cover all families of PGMs proposed so far, but we hope it gives a fair overview of the last developments and the current trends. Among the families of PGMs we did not cover, we may cite the acyclic directed mixed graphs

(ADMGs) [Ric03] which appear to be summary graphs without lines, or the loopless mixed graphs (LMGs) and ribbonless graphs (RGs) [Sad13; SL14] which appear to capture the same independence models as MC graphs. Let us now summarize our state-of-the art in Figure 2.20 and Table 2.8.

In Figure 2.20, we present a hierarchy of the different families of PGMs that we have encountered so far, which corresponds to a partial ordering of these families by set inclusion with respect to the class of independence models they represent. Families that capture the same class of independence models are grouped together (e.g. chain mixed graphs and anterior graphs). A deeper study of the intersection between several of these families can be found in [Sad11; SP15].

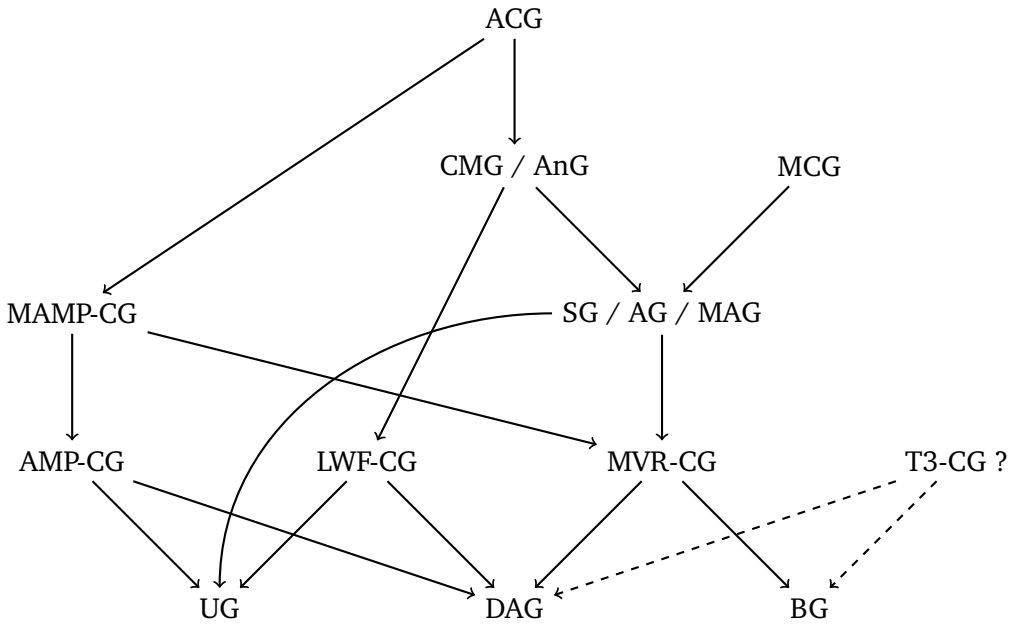


Fig. 2.20. Hierarchy of PGM families by order of inclusion (in terms of independence model classes).

2.3.1 Trends

Advanced PGMs are much less popular and have known fewer practical applications than classical PGMs, for two reasons. The first obvious reason is that increased expressive power of advanced PGMs comes at the price of an increased complexity, both to understand the underlying independence model and to exploit the resulting factorization of p. The second reason is that the field is still very active as we have seen, and current developments are mainly focused on improving the theoretical properties of the structure of these models, without necessarily providing a practical parameterization.

In Table 2.8, we present a (non-exhaustive) list of properties that have been established for each of these families. From left to right, these properties are formulated

as answers to the following questions: i) does every graph yields a probabilistic independence model?; ii) is the family stable under marginalization and conditioning?; iii) is every graph maximal (i.e. it respects the pairwise Markov property)?; iv) if given, when does the factorization of p characterize the independence model (as an I-map)?; and v) are the independence models characterized by a finite set of axioms (conditional independence properties) ? Questions that have not been answered yet are assigned a question mark (?) in the table.

Tab. 2.8. A list of properties exhibited by the different families of PGMs. *See [SL15].

	probabilistic	stable	maximal	I-map $\iff$ fact.	axiom. charac.
UG	yes	yes	yes	$p > 0$	yes
BG	yes	yes	yes	any p	?
DAG	yes	no	yes	any p	no
LWF-CG	yes	no	yes	$p > 0$	?
AMP-CG	yes	no	yes	$p > 0$	?
MVR-CG	yes	no	yes	any p	?
MAMP-CG	yes	no	yes	?	?
MAG	yes	yes	yes	p Gaussian	?
AnG	?	yes	?	?	?
MCG	?	yes	no*	?	?
ACG	?	?	no*	?	?

If we look at the early development of probabilistic graphical models (e.g. the Ising model [Isi25]), we observe that it was factorization-driven, that is, a practical factorization of p over a graph led to the study of the independence model encoded in the graph. In the late development of advanced graphical models, it seems that the trend is reversed, that is, new graphical representations are developed so that they exhibit interesting properties as independence models, while a practical factorization of p seems to be of a second interest, as shown in Table 2.8.

Finally, an interesting property not shown in Table 2.8 is that all the independence models captured by these PGMs are compositional graphoids. The composition property is actually pretty simple to explain, and is inherent to definition of separation between two sets of nodes. Indeed, in all these models set separation is defined as the result of pairwise separation between all pairs of elementary subsets (nodes) of these sets, This necessarily results in the composition property being respected. The necessity of intersection appears less obvious, but also seems to be a consequence of pairwise separation in graphs. See [Kos02][Proposition 2.10] and [SL15][Theorem 1].

From these observations, two interesting questions arise: i) is it possible to define a probabilistic graphical model that is not a compositional graphoid? and ii) because advanced PGMs are more and more expressive, but still seem restricted to composi-

tional graphoids, is it possible to define a PGM that matches exactly the intersection between probabilistic independence models and compositional graphoids? (See Figure 2.7).

2.3.2 Limitations

Graphs vs probabilistic independence models

A first limitation of PGMs comes from their restricted expressiveness as independence models. Studeny [Stu05][3.6] raises the question of how many probabilistic independence models can be described by graphs. We report his results in Table 2.9. As expected, chain graphs are able to capture more independence models than DAGs, which in turn capture more independence models than UGs. However, with only 3 variables these PGMs can capture at most 50% of all possible independence models, and with 4 variables the gap explodes with only 1% of models covered. These numbers provide a strong argument to motivate the development of PGMs with an increased expressive power. But still, Studeny argues that no sufficiently wide class of graphs could possibly cure the problem. Consider a graph with n nodes and m types of edges, in which loops and multiple edges are allowed. The number of possible edges is then $m \times n^2$, and the number of distinct graphs is 2^{mn^2}, the size of the power set of these edges. On the other hand, the number of distinct independence models induced by discrete probability measures is lower bounded by $2^{2^{\lfloor n/2 \rfloor}}$ when $n > 2$. With m fixed, the number of distinct graphical structures grows at an exponential speed of a polynomial of n, while the number of independence models grows at an exponential speed of an exponent of n. In other words, to solve this problem either graphical models should include additional nodes or hyper-edges, or non-graphical independence models should be developed. The latter approach is discussed in [Stu05], who proposes the concept of *structural imsets* to represent independence models.

Tab. 2.9. The number of independence models that can be captured by undirected graphs (UG), directed acyclic graphs (DAG), chain graphs under the LWF interpretation (LWF-CG), and discrete probability distributions (p), with respect to the number of random variables in $\mathbf{V}$. Reproduced from [Stu05].

	$\lvert\mathbf{V}\rvert = 2$	$\lvert\mathbf{V}\rvert = 3$	$\lvert\mathbf{V}\rvert = 4$	$\lvert\mathbf{V}\rvert = 5$
UG	2	8	64	1024
DAG	2	11	185	8782
LWF-CG	2	11	200	11519
p	2	22	18300	?

Independence vs factorization constraints

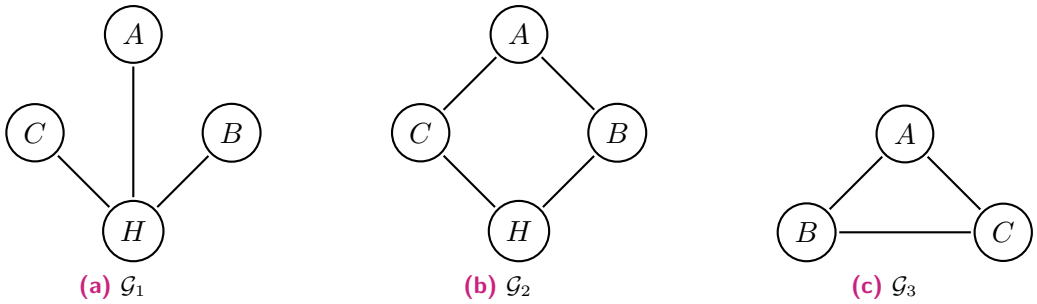
A second limitation of PGMs comes from the restricted expressiveness of independence models themselves. Indeed, an independence model I corresponds to a factorization of p, but not every factorization of p can be induced by an independence model. Consider the two following factorizations: $p(a,b,c) = \phi(a,b,c)$ and $p(a,b,c) = \phi(a,b)\phi(b,c)\phi(c,a)$. Both these distributions are faithful to the same independence model, which corresponds to the UG (c) in Figure 2.21. Unlike the first one, the second factorization simplifies the expression of p and it would be useful to model it. However, it does not induce any independence relation between A, B and C and thus can not be induced by any independence model over A, B, C, graphical or not. Nonetheless, in this particular example it is possible to express the desired factorization with an "augmented" PGM, using latent variables. From the UG in (b), we obtain the desired factorization of $p(a,b,c)$ is obtained after marginalizing out H,

$$p(a,b,c) = \sum_h p(a,b,c,h) = \phi(a,b)\phi(a,c)\sum_h \phi(c,h)\phi(b,h) = \phi(a,b)\phi(a,c)\phi(b,c).$$

From the UG in (a) however, the marginal probability $p(a,b,c)$ does not factorize,

$$p(a,b,c) = \sum_h p(a,b,c,h) = \sum_h \phi(a,h)\phi(b,h)\phi(c,h) = \phi(a,b,c).$$

Interestingly, the additional hidden nodes in the graph can be seen as hyper-edges (i.e. edges between more than two nodes), which was suggested as a solution for solving the previous limitation of PGMs.

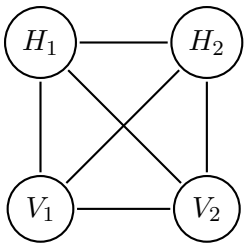


(a) $\mathcal{G}_1$ (b) $\mathcal{G}_2$ (c) $\mathcal{G}_3$

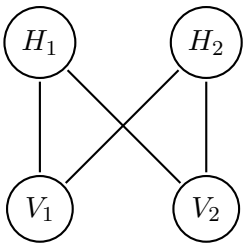
Fig. 2.21. Three UGs $\mathcal{G}_1$, $\mathcal{G}_2$, $\mathcal{G}_3$ such that $I(\mathcal{G}_1)^{\emptyset}_H = I(\mathcal{G}_2)^{\emptyset}_H = I(\mathcal{G}_3)$, that is, the independence model of the marginal distribution $p(a,b,c)$ is the same. Still, $\mathcal{G}_2$ induces the factorization $p(a,b,c) = \phi_1(a,b)\phi_2(b,c)\phi_3(c,a)$, while $\mathcal{G}_1$ does not.

This idea of modeling the relations between the variables of interest with hidden variables is exploited for example within *Boltzmann machines*, or their simpler counterpart the *restricted Boltzmann machines* (RBMs) which impose independence constraints in the global distribution $p(\mathbf{v}, \mathbf{h})$ of the hidden and non-hidden vari-

ables, without any independence constraint on the marginal distribution $p(\mathbf{v})$ (See Figure 2.22).



(a) $\mathcal{G}_1$ a Boltzmann machine



(b) $\mathcal{G}_2$ a restricted Boltzmann machine

Fig. 2.22. Two UGs $\mathcal{G}_1, \mathcal{G}_2$ such that $I(\mathcal{G}_1) \neq I(\mathcal{G}_2)$ but $I(\mathcal{G}_1)^{\emptyset}_{H} = I(\mathcal{G}_2)^{\emptyset}_{H}$.

Contextual independence relations

A third limitation of PGMs is that they do not model contextual independence relations. Recall that in Chapter 1 we defined conditional independence between random variables as $p(\mathbf{x}, \mathbf{y}, \mathbf{z})p(\mathbf{z}) = p(\mathbf{x}, \mathbf{z})p(\mathbf{y}, \mathbf{z})$, for every value of $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$. However, it may be that two random variables are not independent in general, but only in some sub-space of the universe. For example, consider the discrete probability distribution in Table 2.10. The two variables A and B are not independent in general, however they are independent in the particular sub-space where $a \in \{a_3, a_4\}$. If we go back to the definition of independence between events, this may be expressed as either $A = a \perp\!\!\!\perp B = b$ for every $a \in \{a_3, a_4\}$ and $b \in \mathcal{B}$, or $A = a \perp\!\!\!\perp B = b \mid A \in \{a_3, a_4\}$ for every $a \in \mathcal{A}$ and $b \in \mathcal{B}$.

Tab. 2.10. The joint distribution $p(a, b)$ of two discrete random variables A and B. In general the two variables are dependent, however they are independent in the context where $A \in \{a_3, a_4\}$.

		B	
		b_1	b_2
A	a_1	.10	.05
	a_2	.10	.15
	a_3	.10	.20
	a_4	.10	.20

Again, in this particular case we may introduce a hidden variable H such that $H = A$ when $A \in \{a_1, a_2\}$, and $H = c$ a constant value otherwise. In that case the UG $A - H - B$ can represent the contextual independence relation with $A \perp\!\!\!\perp B \mid H$. Note that there exists probabilistic models, also based on graphs, which are able to represent such contextual independence relations. These are for example sum-product networks (SPNs) [PD11], which are equivalent to BNs with hidden variables and compact probability tables [ZMP15], discussed in Section 4.3.5. In both cases

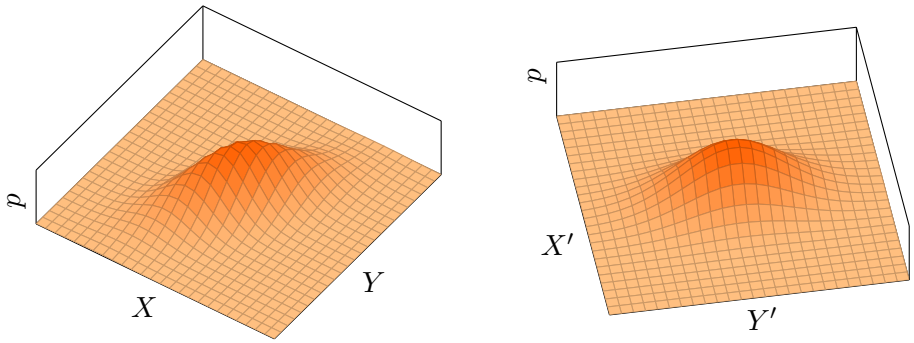
these graphical representation involve much more nodes that the number of variables $|\mathbf{V}|$ in the model.

A-priori on the universe

Finally, one last limitation of PGMs is that their ability to represent independence relations in p is highly related to how the sample space Ω is represented. Consider the extreme situation where only one random variable represents the universe, i.e. $X = \Omega$. Obviously the only independence model for p is $I = \emptyset$, not very useful. For an independence model to exist, it requires several random variables, or more precisely a multi-dimensional representation of the universe. Most of the time these dimensions come naturally because they are meaningful to us, i.e. we may describe a car with its color, its engine capacity, its size, the number of seats etc. But the way we define these random variables is not unique and arbitrary. For example, what about the color of a car? Such an information is commonly represented as a multi-valued discrete variable (white, red, blue etc.), or several binary variables. But we may also represent it as an ordered variable, by arranging the colors from the lightest to the darkest. Or, we may even represent it with three continuous variables, i.e. the RGB combination corresponding to the color. Each of these representations may not be the most appropriate one to represent the sample space in every situation, and choosing a particular representation will necessarily have an impact on how we will model p.

Suppose you are given two random variables X and Y, sampled according to the probability distribution in Figure 2.23a. Obviously these variables are dependent, and a standard PGM will not induce any factorization of p. But if you could rotate your representation of the sample space as in Figure 2.23b, then you would find two independent dimensions X' and Y', with a factorization of p into $p(x')p(y')$. In this simple example a linear transformation of the sample space can exhibit an interesting representation of p, but in general non-linear transformations will also be helpful. This problem of finding a good representation of the sample space is a recurrent issue in machine learning, and is sometimes referred to as feature extraction, feature construction, or representation learning.

In the literature of PGMs the representation problem is often skipped, and people tend to assume that the features of interest are adequate to model p. Sometimes it is also desirable that the representation of the universe is meaningful to humans, so that the structure of the PGM is readable. For example we may collect data samples from a population study, and then learn a PGM structure to uncover the relationships between some variable of interests (e.g. in genomics, social sciences etc.). However,



(a) X and Y are dependent. **(b)** X' and Y' are independent.

Fig. 2.23. The same probability distribution p seen from two different representations of the sample space.

machine learning tasks in general require an accurate and efficient modeling of p, and the way we represent ourselves the universe, i.e. the data samples, is a huge a-priori that the model has to deal with. Since every independence model is tightly related to the underlying representation of the sample space, so are probabilistic graphical models. This is in our view the major limitation of classical probabilistic graphical models in the field of machine learning.

3 Bayesian network structure learning

> *There is no such thing as a true distribution,*
> *[. . .] we only have the data.*
>
> **— Peter D. Grünwald**
> 2007

We now turn to the problem of learning PGM structures from data, and more particularly Bayesian network structures. Learning the structure of a Bayesian network from data is a hard problem in general [CHM04]. In the literature, two main approaches to structure learning are distinguished, namely the score-based approach which iterates over the space of all possible graphs to find the one that maximizes a given score, and the constraint-based approach which reads structural constraints (i.e., independence relations) from the data and builds a graph that respects these constraints. Each of these approaches having its own flaws, recently a new hybrid approach that combines both constraint-based and score-based methods has emerged, which appear to capture the best of both worlds without the disadvantages. In the following we will introduce formally the problem of Bayesian network structure learning, discuss the main score-based, constraint-based and hybrid approaches to Bayesian network structure learning. Finally we will present an experimental comparison of two algorithms: the state-of-the art hybrid algorithm MMHC from [TBA06] and a new hybrid algorithm H2PC [GAE12; GAE14], which constitutes our first major contribution to the field.

3.1 Motivation

Formally, given a set of random variables $\mathbf{V} = \{V_1, \ldots, V_n\}$ and a set of observations $\mathcal{D} = \{\mathbf{v}^{(j)}\}_{j=1}^{s}$ drawn independently from $p(\mathbf{v})$, PGM structure learning consists in finding a graph $\mathcal{G}$ from a specific model family (Markov networks, Bayesian networks, chain graphs. . .) for which there exists a parameterization Θ that encodes the joint distribution $p(\mathbf{v})$. Since a complete graph always respects this condition (i.e., it can encode any distribution), a second desired property of $\mathcal{G}$ is that the resulting model should be of lowest complexity. This preference criterion is often implicit in

the structure learning literature, and does not have a unique definition. In terms of independence models, the above formulation often translates into finding a perfect map of p when such a graph exists, and a minimal independence map of p otherwise (where *minimal* needs further definition).

A first motivation for PGM structure learning comes from the field of *knowledge discovery from databases* (KDD), where statistical models can be used as a tool for extracting meaningful information about a system of interest. Indeed, PGMs provide an intuitive graphical representation of the (in)dependence structure between the variables of the system, which an expert may be able to interpret and gain some insights about the underlying system. Discovering such a representation can be particularly useful in empirical studies where the variables of interest have a specific meaning, e.g., social characteristics in econometrics, nutrient measurements in agronomy, sensor levels in process control, genetic indicators in biology, and so on.

A second motivation comes from the field of *inductive reasoning*, where statistical models can be used for answering probabilistic queries such as $\arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$. In the general case, modeling a complex multivariate distribution naively requires an exponential number of parameters with respect to the number of variables, which leads to intractable models. A practical approach to deal with this problem is to consider tractable models only, by imposing arbitrary constraints on p (e.g., pairwise interactions only, tree structures, specific parametric families, etc.). Still, these constraints may be too restrictive, resulting in approximate statistical models. Structure learning offers a principled solution to consider only structural constraints that respect the underlying data distribution (i.e., an independence maps). It is not guaranteed however that these will result in a tractable statistical model. Nevertheless, even in situations where the learned structure is intractable, gaining some insight about the actual independence structure of p can be useful to guide the choice of a tractable model.

For simplicity, note that in the following we will consider only Bayesian network structure learning in the discrete setting, i.e., with $\mathbf{V} = \{V_1, \ldots, V_n\}$ a set of discrete random variables. Still, the main ideas we discuss here remain valid in the continuous setting, for which most of the presented approaches offer a direct extension.

3.2 The score-based approach

Score-based approaches cast the PGM structure learning problem as an optimization problem. Formally, given a scoring criterion S, the optimal graph $\mathcal{G}^\star$ is defined as the highest scoring structure[1],

$$\mathcal{G}^\star = \arg\max_{\mathcal{G}} S(\mathcal{G}, \mathcal{D}). \tag{3.1}$$

search-and-score

The process of solving (3.1) can be seen as a search over the space of all possible graphs, which gave rise to the *search-and-score* terminology. The first problem that arises from this formulation is: how to compute $S(\mathcal{G}, \mathcal{D})$? Obviously, we would like the score to strongly penalize graphical structures which do not respect the independence model of the data (I-map requirement). But we also would like to favor sparse structures over highly connected ones, when both are equally able to model the data distribution (D-map wish). Also, it may be desirable that the scoring function respects some properties, such as *consistency*:

consistent

$$I(\mathcal{G}_2) \subset I(\mathcal{G}_1) \subseteq I(p) \implies \lim_{|\mathcal{D}|\to\infty} S(\mathcal{G}_1, \mathcal{D}) > \lim_{|\mathcal{D}|\to\infty} S(\mathcal{G}_2, \mathcal{D}), \text{ and}$$

$$I(p) \subseteq I(\mathcal{G}_1) \subset I(\mathcal{G}_2) \implies \lim_{|\mathcal{D}|\to\infty} S(\mathcal{G}_1, \mathcal{D}) > \lim_{|\mathcal{D}|\to\infty} S(\mathcal{G}_2, \mathcal{D}),$$

equivalent

or *equivalence*:

$$I(\mathcal{G}_1) = I(\mathcal{G}_2) \implies S(\mathcal{G}_1, \mathcal{D}) = S(\mathcal{G}_2, \mathcal{D}).$$

decomposable

Another interesting property is *decomposability*, which imposes that the scoring function decomposes as a sum of local scores for each node and its parents:

$$S(\mathcal{G}, \mathcal{D}) = \sum_{i=1}^{n} S(V_i, \mathbf{PA}_{V_i}, \mathcal{D}).$$

In the following we will present two families of scoring functions, respectively based on a Bayesian and an information-theoretic criterion. We will then review some search procedures which are typically used to solve the optimization problem in (3.1).

[1] Note that the optimal graph $\mathcal{G}^\star$ is not necessarily unique, thus the $\in$ symbol should be preferred to $=$ in (3.1). For simplicity, we will omit this ambiguity and always refer to *the* optimal graph.

3.2.1 Bayesian scores

In order to derive a proper Bayesian scoring function, we may re-express Equation (3.1) as a MAP estimation problem,

$$\mathcal{G}^\star = \arg\max_{\mathcal{G}} p(\mathcal{G}|\mathcal{D}), \tag{3.2}$$

where $p(\mathcal{G}|\mathcal{D})$ is the posterior probability of the graphical structure $\mathcal{G}$ given the data set $\mathcal{D}$. Recall that $\mathcal{D}$ is fixed and thus $p(\mathcal{D})$ is a constant term, under Bayes' law we have $p(\mathcal{G}|\mathcal{D}) = p(\mathcal{G})p(\mathcal{D}|\mathcal{G})/p(\mathcal{D})$ and thus

$$p(\mathcal{G}|\mathcal{D}) \propto p(\mathcal{G})p(\mathcal{D}|\mathcal{G}). \tag{3.3}$$

The scoring function then decomposes into an a-priori on $\mathcal{G}$ and a likelihood term $p(\mathcal{D}|\mathcal{G})$, i.e., the probability of obtaining the observed data set from a probabilistic model with graphical structure $\mathcal{G}$. We may then introduce the model parameters Θ, and express the likelihood term as a proper marginalization,

$$p(\mathcal{D}|\mathcal{G}) = \int_\Theta p(\mathcal{D}, \Theta|\mathcal{G})\mathrm{d}\Theta. \tag{3.4}$$

We may now decompose further the inner term into $p(\Theta|\mathcal{G})p(\mathcal{D}|\mathcal{G}, \Theta)$. Because $\mathcal{D}$ is made of i.i.d. (independent and identically distributed) samples, we obtain

$$\mathcal{G}^\star = \arg\max_{\mathcal{G}} p(\mathcal{G}) \int_\Theta p(\Theta|\mathcal{G}) \prod_{\mathbf{v}\in\mathcal{D}} p(\mathbf{v}|\mathcal{G}, \Theta)\mathrm{d}\Theta.$$

The probability of a sample $p(\mathbf{v}|\mathcal{G}, \Theta)$ can be induced from the model, and what remains to be defined are an a-priori on the structures, $p(\mathcal{G})$, and an a-priori on the parameters given a particular structure, $p(\Theta|\mathcal{G})$. While the a-priori on the structure can be chosen arbitrarily, for practical considerations the a-priori on the parameters must make the integration $\int_\Theta$ tractable.

Bayesian Dirichlet family

In the discrete case, a convenient choice of prior is the Dirichlet distribution, which conjugates nicely with multinomial probability distributions. Such a prior is used in
BD the so-called family of *Bayesian Dirichlet* (BD) scores [HGC95], based on a factorized Dirichlet distribution of the parameters. Formally, the global distribution $p(\Theta|\mathcal{G})$ is assumed to factorize over each node V_i of the graph,

$$p(\Theta|\mathcal{G}) = \prod_{i=1}^{n} p(\Theta_i|\mathcal{G}),$$

where Θ_i is the set of parameters specific to the conditional probability table of V_i. Each of these local distributions is then assumed to factorize further over each possible instantiation the parents of V_i, that is,

$$p(\Theta_i|\mathcal{G}) = \prod_{j=1}^{q_i} p(\Theta_{i,j}|\mathcal{G}),$$

where q_i denotes the number of possible instantiations of $\mathbf{PA}_{V_i}$, and $\Theta_{i,j}$ is the set of parameters specific to the conditional probability distribution of V_i when its parents take their j-th value. Finally, each of these local, context-specific distributions is expressed as a Dirichlet distribution parameterized by $\{\alpha_{i,j,k}\}_{k=1}^{r_i}$,

$$p(\Theta_{i,j}|\mathcal{G}) = \frac{\Gamma(\sum_{k=1}^{r_i} \alpha_{i,j,k})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{i,j,k})} \prod_{k=1}^{r_i} \theta_{i,j,k}^{\alpha_{i,j,k}-1},$$

where Γ is the gamma function, r_i denotes the number of possible instantiations of V_i, and $\theta_{i,j,k}$ is the probability that V_i takes its k-th value when its parents take their j-th value. Because this Dirichlet prior conjugates nicely with multinomial distributions, marginalizing out Θ in (3.4) results in

$$p(\mathcal{D}|\mathcal{G}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \left(\frac{\Gamma(\alpha_{i,j})}{\Gamma(s_{i,j} + \alpha_{i,j})} \prod_{k=1}^{r_i} \frac{\Gamma(s_{i,j,k} + \alpha_{i,j,k})}{\Gamma(\alpha_{i,j,k})} \right), \tag{3.5}$$

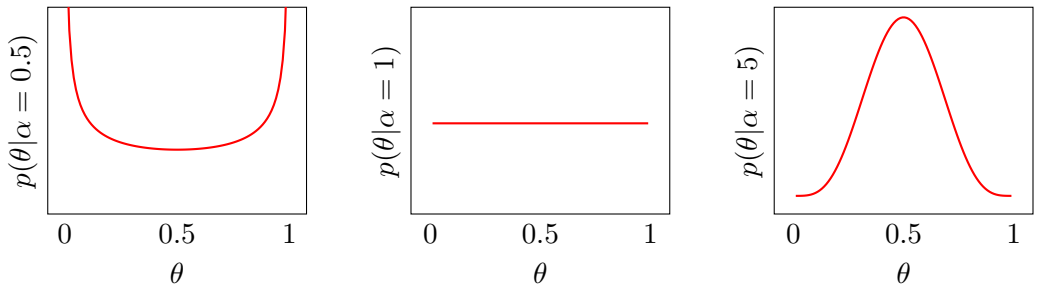
where $s_{i,j,k}$ counts the number of samples in $\mathcal{D}$ where the variable V_i takes its k-th value while its parents take their j-th value, $s_{i,j} = \sum_k s_{i,j,k}$ and $\alpha_{i,j} = \sum_k \alpha_{i,j,k}$. Finally, (3.3) is turned into a logarithmic score $\log p(\mathcal{D}|\mathcal{G}) + \log p(\mathcal{G})$, which preserves the scoring order while turning the multiple products in (3.5) into convenient summations,

$$S_{BD}(\mathcal{G}, \mathcal{D}) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left(\log \frac{\Gamma(\alpha_{i,j})}{\Gamma(s_{i,j} + \alpha_{i,j})} \sum_{k=1}^{r_i} \log \frac{\Gamma(s_{i,j,k} + \alpha_{i,j,k})}{\Gamma(\alpha_{i,j,k})} \right) + \log p(\mathcal{G}).$$

Such a scoring function has the desirable property that it is decomposable, i.e., if one changes the parent set of a single node in $\mathcal{G}$, then the score of the new graph $\mathcal{G}'$ needs not be re-computed entirely, instead only the local term that corresponding that node must be updated.

K2

In order for the BD score to be used in practice, one needs to explicitly define the parameters of each of the Dirichlet prior. One common choice is the non-informative parameter $\alpha_{i,j,k} = 1$ everywhere, in which case the Dirichlet boils down to a uniform



(a) $s' = 1(\alpha = 0.5)$. **(b)** $s' = 2(\alpha = 1)$. **(c)** $s' = 10(\alpha = 5)$.

Fig. 3.1. Shape of the BDeu prior for a binary variable V_i with no parents ($r_i = 2$, $q_i = 1$, $\alpha = s'/2$), with different imaginary sample sizes. The probability distribution $p(v_i)$ resumes to a single parameter $\theta = p(v_i = 1)$. The BDeu prior is distributed symmetrically around the uniform distribution $\theta = 0.5$, and concentrates either towards it when $s' < r_i q_i$, or towards the deterministic distributions $\theta = 0$, $\theta = 1$, when $s' > r_i q_i$.

K2 prior. Such a parameterization is commonly referred to as the *K2* score introduced by Cooper and Herskovits [CH91], whose expression simplifies to

$$S_{K2}(\mathcal{G}, \mathcal{D}) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left(\log \frac{(r_i - 1)!}{(s_{i,j} + r_i - 1)!} \sum_{k=1}^{r_i} \log s_{i,j,k}! \right) + \log p(\mathcal{G}).$$

BDeu

Another common choice of parameters is $\alpha_{i,j,k} = s'/(q_i r_i)$, proposed in [Bun91],
BDeu which results in the so-called *BDeu* score,

$$S_{BDeu}(\mathcal{G}, \mathcal{D}) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left(\log \frac{\Gamma(\frac{s'}{q_i})}{\Gamma(s_{i,j} + \frac{s'}{q_i})} \sum_{k=1}^{r_i} \log \frac{\Gamma(s_{i,j,k} + \frac{s'}{q_i r_i})}{\Gamma(\frac{s'}{q_i r_i})} \right) + \log p(\mathcal{G}).$$

As a result, BDeu requires a single parameter, s', called the *imaginary sample size*. The under scripts in BDeu stand for *equivalent* and *uniform*. Indeed, the equivalence property is guaranteed with BDeu, i.e. two graphs with the same independence model are always given the same score, which is not necessarily true with other BD scores such as K2. On the other hand, *uniform* refers to the shape of the resulting Dirichlet distributions, whose density is symmetrically distributed around uniform $\Theta_{i,j}$ values, i.e. $p(v_i|\mathbf{pa}_{V_i}, \mathcal{G})$ uniform. The imaginary sample size parameter s' then expresses how the prior concentrates towards or steps aside from this point, with $p(v_i|\mathbf{pa}_{V_i}, \mathcal{G})$ more likely uniform when $\frac{s'}{q_i r_i} > 1$, more likely deterministic when $\frac{s'}{q_i r_i} < 1$, and equally likely when $\frac{s'}{q_i r_i} = 1$, as illustrated in Figure 3.1. As a result, BDeu appears to be rather sensitive to the s' parameter.

Discussion

While this Bayesian formulation of a scoring metric is principled and allows to easily integrate any prior on the graphical structures, in practice a non-informative uniform prior is almost always used for $p(\mathcal{G})$. An interesting question is then: how can BD scores favor sparse structures over complex ones, given that a uniform prior is given to every possible structure? If we consider the K2 score, the situation is even more paradoxical: a uniform prior is used for $p(\mathcal{G})$, as well as a uniform prior for every parameterization of the graph, $p(\Theta|\mathcal{G})$. And yet the K2 score favors sparse structures, as shown in [CH91]. The explanation to this counter-intuitive effect is the following: K2 does not give a uniform prior over $p(\mathcal{G}, \Theta)$, because Θ implicitly depends on $\mathcal{G}$. Consider Θ not as a set of parameters, but rather as the resulting probability distribution. Clearly, with $\mathcal{G}_1$ and $\mathcal{G}_2$ two graphs such that $I(\mathcal{G}_2) \subset I(\mathcal{G}_1)$, the denser graph $\mathcal{G}_2$ can represent a broader class of probability distributions than $\mathcal{G}_1$. Therefore, when Θ can be encoded both by $\mathcal{G}_1$ and $\mathcal{G}_2$ then $p(\Theta|\mathcal{G}_1)$ will be greater than $p(\Theta|\mathcal{G}_2)$ with uniform priors, simply because Θ is one among the set of possible parameterizations for $\mathcal{G}_1$, and one among a larger set for $\mathcal{G}_2$. Therefore, a side-effect of the K2 prior is that a sparse structure will be preferred over a denser one, if both are equally able to encode the data distribution. A second side-effect, pointed out in [Aye94], is that the K2 score is not equivalent in term of independence models, given two graphs such that $I(\mathcal{G}_1) = I(\mathcal{G}_2)$, the structure that encodes the underlying distribution with the denser parameterization (i.e. non-deterministic probability tables) will always be preferred. For example, with A and B two random variables such that A is deterministic for B but B is not for A, the graph $B \rightarrow A$ will be preferred by K2 over the graph $A \rightarrow B$.

Note that both K2 and BDeu are decomposable and consistent scoring functions, though only BDeu is score-equivalent. Therefore BDeu is almost always favored in practice, despite its sensitivity to the choice of α.

3.2.2 Information-theoretic scores

The basic idea behind information-theoretic scores is to formulate the structure learning problem as a data compression problem, based on the idea that the more the data is compressed, the more regularities have been found. Formally, given a hypothesis space $\mathcal{H}$, the *Minimum Description Length* (MDL) [Ris78] of a data set $\mathcal{D}$ is

$$L(\mathcal{D}) = \min_{H \in \mathcal{H}} L(H) + L(\mathcal{D}|H),$$

where $L(H)$ measures the length of the shortest sequence for describing H, and $L(\mathcal{D}|H)$ the length of the shortest sequence for describing $\mathcal{D}$ under hypothesis H.

The MDL principle offers a natural protection against overfitting, the best hypothesis for describing the data being the one that reaches the optimal balance between model complexity and data compression [Grü07]. Interestingly, finding H that minimizes $L(H) + L(\mathcal{D}|H)$ can be interpreted as a MAP inference problem over $p(H|\mathcal{D}) \propto p(H)p(\mathcal{D}|H)$, where the hypothesis complexity defines a particular prior distribution $p(H) = 2^{-L(H)}$ (equivalently $L(H) = -\log p(H)$)[2] and the data complexity given the hypothesis defines a likelihood distribution $p(\mathcal{D}|H) = 2^{-L(\mathcal{D}|H)}$ (equivalently $L(\mathcal{D}|H) = -\log p(\mathcal{D}|H)$). While in a Bayesian setting the prior distribution corresponds to a prior belief based on expert knowledge, in MDL it corresponds to a complexity measure of the hypothesis, which favors simple ones over complex ones according to Occam's razor principle. The main difficulty is then: how to define the complexity of a hypothesis?

In the context of probabilistic graphical models our hypotheses take the form $H = (\mathcal{G}, \Theta)$. Therefore, the best model for describing the data is given by

$$(\mathcal{G}, \Theta)^\star = \arg\min_{\mathcal{G},\Theta} L(\mathcal{G}, \Theta) + L(\mathcal{D}|\mathcal{G}, \Theta), \tag{3.6}$$

where $L(\mathcal{G}, \Theta)$ measures the length of the shortest sequence (in bits) required to describe the model structure and parameters, and $L(\mathcal{D}|\mathcal{G}, \Theta)$ that for describing the data set given the model. According to information theory, the minimum number of bits required to describe an i.i.d. data set with a probabilistic model is $-\log p(\mathcal{D}|\mathcal{G}, \Theta)$, a.k.a. the negative log-likelihood of the data. On the other hand, the minimum number of bits required to represent a model is related to its Kolmogorov complexity [Kol63], which is uncomputable and must be approximated with some arbitrary complexity measure. Following this formulation, the best structure for describing the data is given by

$$\mathcal{G}^\star = \arg\min_{\mathcal{G}} L(\mathcal{G}) + \min_{\Theta} L(\Theta|\mathcal{G}) + L(\mathcal{D}|\mathcal{G}, \Theta), \tag{3.7}$$

where $L(\mathcal{G})$ measures the length of the shortest sequence required to describe the model structure, and $L(\Theta|\mathcal{G})$ that for describing the model parameters given the structure.

Parametric complexity

Two problems arise with the MDL formulation for structure learning. First, in (3.6) both $\mathcal{G}$ and Θ are learned at the same time, by measuring the total model complexity $L(\mathcal{G}, \Theta)$. Therefore, it may very well be that $L(\mathcal{G}, \Theta)$ is small due to structural

[2] More formally, $p(H) = \frac{1}{Z} 2^{-L(H)}$ and $L(H) = -\log Z - \log p(H)$, with Z some constant.

constraints in the parameters Θ, while $\mathcal{G}$ remains a fully connected graph. This is problematic for structure learning since we would like to capture constraints in $\mathcal{G}$, not in Θ. Second, the inner minimization in (3.7) in general does not accept a closed-form solution, and requires a cumbersome exploration of the parameter space. A practical solution to both these problems is to replace $L(\Theta|\mathcal{G})$ with $C^{|\mathcal{D}|}(\mathcal{G})$, an upper bound called the *parametric complexity* of the model, which measures the expressiveness of the model for fitting data sets of size $s = |\mathcal{D}|$. The best structure under this refined MDL principle is then given by

$$\mathcal{G}^\star = \arg\min_{\mathcal{G}} L(\mathcal{G}) + C^{|\mathcal{D}|}(\mathcal{G}) + \min_{\Theta} L(\mathcal{D}|\mathcal{G}, \Theta). \tag{3.8}$$

With this new formulation, the model complexity now depends only on the graphical structure $\mathcal{G}$, relative to the size of the data set to be described. Still, in order to derive proper scores one must give appropriate measures for $L(\mathcal{G})$ and $C^{|\mathcal{D}|}(\mathcal{G})$.

LL

A first straightforward approach is to consider only the likelihood term $L(\mathcal{D}|\mathcal{G}, \Theta)$.
LL This results in the *log-likelihood* (LL) scoring function, which in the discrete case is expressed as

$$S_{LL}(\mathcal{G}, \mathcal{D}) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} s_{i,j,k} \log \frac{s_{i,j,k}}{s_{i,j}},$$

where r_i is the number of possible instantiations of the random variable V_i, q_i is the number of possible instantiation of its parents $\mathbf{PA}_{V_i}$, $s_{i,j,k}$ counts the number of samples in $\mathcal{D}$ where the variable V_i takes its k-th value while its parents take their j-th value, and $s_{i,j} = \sum_k s_{i,j,k}$. Obviously LL does not favor simple models over complex ones, and therefore is not favored in practice.

AIC

Another common choice is to include the parametric complexity $C^{|\mathcal{D}|}(\mathcal{G})$ and ignore the structure complexity $L(\mathcal{G})$. A simple approximate is the number of free
AIC parameters in Θ, resulting in the *Akaike Information Criterion* (AIC) scoring function [Aka74],

$$S_{AIC}(\mathcal{G}, \mathcal{D}) = S_{LL}(\mathcal{G}, \mathcal{D}) - \sum_{i=1}^{n} (r_i - 1) q_i.$$

BIC

A finer approximate of the parametric complexity is given by the number of bits required to store the free parameters. Since the precision of the maximum-likelihood parameters aligns with the observed frequencies in the data set, the number of bits required to express each parameter can be related to the sample size s [FY96], resulting in the so-called *Bayesian Information Criterion* (BIC) scoring function [Sch78; Ris78],

BIC

$$S_{BIC}(\mathcal{G}, \mathcal{D}) = S_{LL}(\mathcal{G}, \mathcal{D}) - \frac{\log s}{2} \sum_{i=1}^{n} (r_i - 1) q_i.$$

MDL

Finally, the structure complexity $L(\mathcal{G})$ can be approximated by the number of bits required to encode the structure, that is, one integer in $\{0, \ldots, n\}$ for each node to indicate the size of its parent set, plus one integer in $\{1, \ldots, n\}$ per parent. The resulting criterion is referred in [Cru+06] as the *Minimum Description Length* (MDL) scoring function,

MDL

$$S_{MDL}(\mathcal{G}, \mathcal{D}) = S_{LL}(\mathcal{G}, \mathcal{D}) - \frac{\log s}{2} \sum_{i=1}^{n} (r_i - 1) q_i - \sum_{i=1}^{n} (|\mathbf{PA}_{V_i}| + 1) \log n.$$

Note that in the literature many authors consider BIC as *the* MDL criterion. For example, Burnham and Anderson [BA02][p. 286] write

> "Rissanen [Ris89] proposed a criterion that he called minimum description length (MDL) [...] , his result is equivalent to BIC."

This is wrong, as argued in [Grü07][p. 552],

> "We see that in AIC and BIC, the penalty of each model only depends on the number of parameters and the sample size. In MDL model selection, it also depends on the functional form of the model. [...] We note that researchers who claim MDL = BIC do have an excuse: in early work, Rissanen himself has used the phrase "MDL criterion" to refer to BIC, and, unfortunately, the phrase has stuck."

Discussion

Note that all the information-theoretic scores discussed here (LL, AIC, BIC and MDL) are decomposable scoring functions, however only BIC was shown to be consistent and equivalent, while LL is known to be inconsistent. In practice, empirical results suggest that information-theoretic scoring functions based on the MDL principle perform equally well, if not better, than Bayesian scoring functions [LMY12].

3.2.3 The optimization problem

When adopting a score-based approach, finding the best structure involves solving an optimization problem over the set of all possible DAGs, which grows exponentially with the number of variables [Rob73]. Unsurprisingly, Chickering [Chi95] shows that finding the optimal DAG according to the BDe scoring function is NP-hard, and later extends this result to any consistent scoring function [CMH03].

Several works investigate on exact optimization procedures by using dynamic programming [KS04; OIM04; SM05; SM06], integer linear programming [Jaa+10; Cus11; CB13], branch-and-bound [dJ11], or shortest path exploration [YMW11; YM13; FMY14]. Currently, the time complexity of such procedures remains $O(n2^n)$ [Koj+10]. In the meantime, several authors focus on approximate procedures to recover a high-scoring structure in reasonable time. The simplest such approach is greedy hill climbing, which starts with a random (usually empty) graph and repeatedly apply single edge addition, deletion or reversal until it reaches a locally optimal structure. In order to overcome the local optima problem, all kinds of approximate search procedure are conceivable, among which simulated annealing [HGC95], genetic programming [Lar+96], tabu search [GL99] or ant colony optimization [de +02]. Still, applying the search-and-score approach to large-scale problems remains computationally very expensive, and the sub-optimal structures resulting from different search strategies can exhibit a high variability [MJM15].

3.2.4 Meek's conjecture

An important theoretical result in Bayesian network structure leaning is the so-called "Meek's conjecture", originally suggested by Meek [Mee97] in his PhD thesis and later proven by [Chi02]. The conjecture implies the notion of covered edges in DAGs, that is, edges which are not part of any v-structure[3].

[3]Formally, an edge $V_i \to V_j$ is said covered when $\mathbf{PA}_{V_i} = \mathbf{PA}_{V_j} \setminus \{V_i\}$.

Algorithm 1 Greedy Equivalent Search (GES)

Require: $\mathbf{V} = \{V_1, \ldots, V_n\}$ a set of random variables whose joint distribution is DAG-faithful, S a consistent scoring function[4].
Ensure: $\mathcal{G}$ a DAG faithful to $p(\mathbf{v})$.
1: $\mathcal{G} \leftarrow$ an empty DAG over $\mathbf{V}$
2: $\mathcal{G} \leftarrow \text{GFS}(\mathcal{G}, S)$ ▷ 1) greedy forward search
3: $\mathcal{G} \leftarrow \text{GBS}(\mathcal{G}, S)$ ▷ 2) greedy backward search

Thm. 3.1 *Meek's conjecture* *With $\mathcal{G}$, $\mathcal{H}$ two DAGs such that $I(\mathcal{G}) \subseteq I(\mathcal{H})$, there exists a finite sequence of edge removal and covered edge reversal operations in $\mathcal{G}$ such that: 1) after each operation $I(\mathcal{G}) \subseteq I(\mathcal{H})$; and 2) after all operations $\mathcal{G} = \mathcal{H}$.*

greedy backward search

One direct consequence of Meek's conjecture is a justification for the *greedy backward search* (GBS) algorithm, that is, start from a complete DAG and repeatedly apply edge removal or covered edge reversal while the score increases. In the limit of large sample size[5] and with a consistent scoring function, GBS is guaranteed to converge to a faithful DAG when p is DAG-faithful [Chi02]. When p is not DAG-faithful, then GBS results in an inclusion-optimal DAG, that is, $I(\mathcal{G}) \subseteq I(p)$ and there exists no DAG $\mathcal{H}$ such that $I(\mathcal{G}) \subset I(\mathcal{H})$ and $I(\mathcal{H}) \subseteq I(p)$ [CM02]. Therefore, GBS can offer theoretical guarantees while exploring only a small subset of the entire search space, which is a very appealing property. In situations where the optimal DAG is known to be sparse, it can be more efficient to first perform a *greedy forward search* (GFS), that is, start from an empty DAG $\mathcal{G}$ and repeatedly apply edge addition while the score increases. In the limit of large sample size and given a consistent scoring function, Nielsen et al. [NKP03] show that GFS is guaranteed to converge to an independence map ($I(\mathcal{G}) \subseteq I(p)$) under the Composition assumption. When combining GFS and GBS we obtain the *Greedy Equivalence Search* (GES) algorithm [Chi02] (Algorithm 1), which is guaranteed to output in the large-sample limit an inclusion-optimal DAG under the Composition assumption, and a faithful DAG under the DAG-faithfulness assumption.

greedy forward search

GES

3.3 The constraint-based approach

Constraint-based approaches make use of statistical independence tests to read the independence model of the data-generating distribution, $I(p)$, in order to build a graphical structure $\mathcal{G}$ that respects the data constraints, $I(\mathcal{G}) \subseteq I(p)$.

[4] $S(\mathcal{G}) = \lim_{|\mathcal{D}|\to\infty} S(\mathcal{G}, \mathcal{D})$, $\mathcal{D} \sim p(\mathbf{v})$.
[5] That is, when the empirical distribution of $\mathcal{D}$ converges to p.

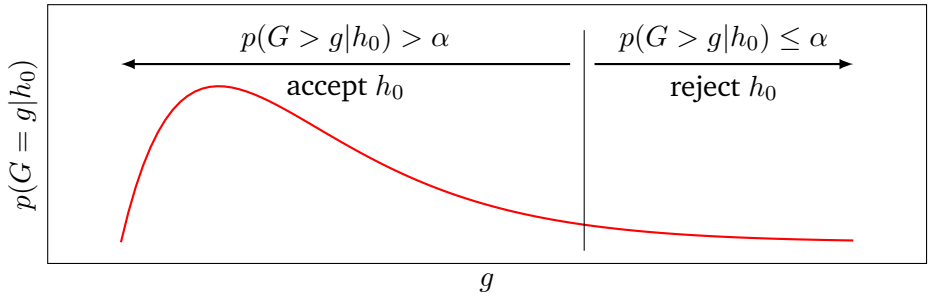


Fig. 3.2. Distribution of the G statistic under h_0 with 4 degrees of freedom (e.g., $|\mathcal{X}| = 3$, $|\mathcal{Y}| = |\mathcal{Z}| = 2$). The vertical bar indicates the lower threshold for rejecting the null hypothesis with confidence level $\alpha = 5\%$, that is, the g value where $\int_g^{+\infty} p(G = g|h_0) = 5\%$.

3.3.1 Conditional independence tests

In order to evaluate whether an independence relation $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$ holds in p or not, one typically runs an asymptotic statistical test of conditional independence (CI test) on $\mathcal{D}$, such as a χ^2 test or a G-test for discrete variables, and a Fisher's Z-test for continuous variables. Each of these tests computes a statistic based on the observations for $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ in $\mathcal{D}$, whose distribution under the null hypothesis of independence h_0 is known. When the statistic is considered too unlikely under h_0, then the null hypothesis is rejected and $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$ is preferred. Otherwise, the null hypothesis $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$ is accepted. As such, "classical" CI tests follow a frequentist approach, as they make a decision based on a data-likelihood measure under the null hypothesis, $p(\mathcal{D}|h_0)$.

G-test

As an example, consider $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ discrete random variables, and the G-test of the null hypothesis $h_0 = \mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$. The statistic of the test is $g = 2s \times I(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})$, where s denotes the number of samples in $\mathcal{D}$ and $I(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})$ denotes the conditional mutual information between $\mathbf{X}$ and $\mathbf{Y}$ given $\mathbf{Z}$ in the empirical data distribution[6]. Formally, the G statistic is expressed as

$$g = 2 \sum_{\mathbf{x} \in \mathcal{X}} \sum_{\mathbf{y} \in \mathcal{Y}} \sum_{\mathbf{z} \in \mathcal{Z}} s_{\mathbf{x},\mathbf{y},\mathbf{z}} \log \frac{s_{\mathbf{x},\mathbf{y},\mathbf{z}} s_{\mathbf{z}}}{s_{\mathbf{x},\mathbf{z}} s_{\mathbf{y},\mathbf{z}}},$$

where $0/0 = 1$ by definition, $s_{\mathbf{x},\mathbf{y},\mathbf{z}}$ counts the number of data samples where $\mathbf{X} = \mathbf{x}$, $\mathbf{Y} = \mathbf{y}$ and $\mathbf{Z} = \mathbf{z}$, $s_{\mathbf{x},\mathbf{z}}$ the number of samples where $\mathbf{X} = \mathbf{x}$ and $\mathbf{Z} = \mathbf{z}$, and so on. Then, under the null hypothesis h_0, by the central limit theorem the G statistic asymptotically[7] follows a χ^2 distribution with $(|\mathcal{X}| - 1)(|\mathcal{Y}| - 1)(\mathcal{Z})$ degrees of freedom [Kul68]. The p-value of the test is then given by the asymptotic probability $p(G > g|h_0)$ (see Figure 3.2), and the null hypothesis is rejected when that probability is lower than a given threshold α, usually 5% or 1%.

[6] A.k.a. the *conditional relative entropy* or *conditional Kullback-Leibler divergence* between $p(\mathbf{x}, \mathbf{y}|\mathbf{z})$ and $p(\mathbf{x}|\mathbf{z})p(\mathbf{y}|\mathbf{z})$.

[7] That is, when the empirical distribution of $\mathcal{D}$ converges to p.

When the available sample size is too small, asymptotic CI tests are likely to fail to reject the null hypothesis, and always accept independence ($\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$). In fact, the required sample size depends implicitly upon the number of degrees of freedom of the test, which increases exponentially with the number of variables in the $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ subsets. Therefore, it is of practical interest to read (in)dependence relations from $\mathcal{D}$ only between and conditioned on small variable sets, and avoid high-order statistical tests as much possible.

3.3.2 The faithfulness assumption

In the general case, one can infer a rich independence model $I(p)$ from a smaller subset of conditional independence tests, by using general CI properties such as the semi-graphoid axioms (Section 1.1.3). For example, from $A \perp\!\!\!\perp \{B, C\} \mid \emptyset$ we can deduce $A \perp\!\!\!\perp B \mid C$ without performing a second statistical test, due to the *Decomposition* property. Still, in the general case, the problem identifying a parameter-minimal DAG with an independence oracle is NP-hard [CHM04].

In situations where the underlying distribution is known to be DAG-faithful, one can characterize rich independence models with even fewer CI tests, by using additional CI properties which are specific to DAGs (e.g., Theorem 2.5). Therefore, many constraint-based algorithms for Bayesian network structure learning assume that p is DAG-faithful in order to derive efficient correct procedures [CHM04]. However, the behaviour of such procedures when the DAG-faithfulness assumption is not met is usually unknown.

3.3.3 Algorithms

We may now present the most popular constraint-based algorithms for Bayesian network structure learning, according to a chronological order.

SGS

Under the DAG-faithfulness assumption, a faithful DAG $\mathcal{G}$ can be characterized with only two properties [VP90]. The first one, known as the *pairwise Markov property*, gives a sufficient and necessary condition for two nodes to be adjacent in $\mathcal{G}$,

$$Y \in \mathbf{PC}_X \iff X \not\perp\!\!\!\perp Y \mid \mathbf{Z}, \quad \forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}. \tag{3.9}$$

Algorithm 2 Spirtes-Glymour-Scheines (SGS)

Require: $\mathbf{V} = \{V_1, \dots, V_n\}$ a set of random variables whose joint distribution is DAG-faithful, $(\cdot \perp\!\!\!\perp \cdot | \cdot)$ a CI oracle.
Ensure: $\mathcal{G}$ a DAG faithful to $p(\mathbf{v})$.

$\mathcal{G} \leftarrow$ complete UG over $\mathbf{V}$
for all neighbor pairs $V_i - V_j$ **do** ▷ 1) recover skeleton
  if $\exists \mathbf{Z} \subseteq \mathbf{V} \setminus \{V_i, V_j\}$ s.t. $V_i \perp\!\!\!\perp V_j \mid \mathbf{Z}$ **then**
    Remove edge $V_i - V_j$
for all potential v-structures $V_i - V_j - V_k$ **do** ▷ 2) orient edges
  if $\nexists \mathbf{Z} \subseteq \mathbf{V} \setminus \{V_i, V_j, V_k\}$ s.t. $V_i \not\perp\!\!\!\perp V_k \mid \mathbf{Z} \cup \{V_j\}$ **then**
    Orient edges $V_i \rightarrow V_j \leftarrow V_k$
Orient all remaining und. edges, without creating any cycle or v-structure

This property is sufficient to characterize the *skeleton*[8] of $\mathcal{G}$. Once the skeleton is known, a second property gives a necessary and sufficient condition for identifying a *v-structure* $X \rightarrow W \leftarrow Y$. Formally, when $\{X, Y\} \subseteq \mathbf{PC}_W$ and $Y \notin \mathbf{PC}_X$[9],

$$\{X, Y\} \subseteq \mathbf{PA}_W \iff X \not\perp\!\!\!\perp Y \mid \mathbf{Z} \cup \{W\}, \quad \forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, W\}. \tag{3.10}$$

Since all faithful DAGs share the same skeleton and the same set of v-structures [VP90], edges that do not belong to a v-structure can be given any direction, so long as no cycle or new v-structure is created in the graph. A direct application of this characterization is the *Spirtes-Glymour-Scheines* (SGS) algorithm (Algorithm 2), named after its authors [SGS90]. While being asymptotically correct under the DAG-faithfulness assumption, the SGS algorithm still requires an exponential number of statistical independence tests with potentially large conditioning sets, which is not very efficient.

PC

Two other interesting properties implied by the DAG-faithfulness assumption allow for more efficient procedures. The first one extends (3.9), and allows for skeleton identification with fewer CI tests,

$$Y \in \mathbf{PC}_X \iff \begin{cases} X \not\perp\!\!\!\perp Y \mid \mathbf{Z}, & \forall \mathbf{Z} \subseteq \mathbf{PC}_X \setminus \{Y\}, \text{ and} \\ X \not\perp\!\!\!\perp Y \mid \mathbf{Z}, & \forall \mathbf{Z} \subseteq \mathbf{PC}_Y \setminus \{X\}. \end{cases} \tag{3.11}$$

[8] The skeleton of a DAG denotes the UG with same adjacencies.
[9] Recall that $\mathbf{PC}_X$ denotes the set of parents and children of X in $\mathcal{G}$. Likewise, $\mathbf{PA}_X$, $\mathbf{CH}_X$ and $\mathbf{SP}_X$ denote respectively the parents, children, and spouses of X.

Algorithm 3 Peter-Clark (PC)

Require: $\mathbf{V} = \{V_1, \dots, V_n\}$ a set of random variables whose joint distribution is DAG-faithful, $(\cdot \perp\!\!\!\perp \cdot \mid \cdot)$ a conditional independence oracle.
Ensure: $\mathcal{G}$ a DAG faithful to $p(\mathbf{v})$.

$\mathcal{G} \leftarrow$ complete UG over $\mathbf{V}$
for m from 0 to $n-2$ **do** ▷ 1) recover skeleton
  for all ordered neighbor pairs $V_i - V_j$ **do**
    if $\exists \mathbf{Z} \subseteq \mathbf{AD}_{V_i} \setminus \{V_j\}$ s.t. $|\mathbf{Z}| = m$ and $V_i \perp\!\!\!\perp V_j \mid \mathbf{Z}$ **then**
      Remove edge $V_i - V_j$
      $\mathbf{S}_{ij} \leftarrow \mathbf{Z}$
for all potential v-structures $V_i - V_j - V_k$ **do** ▷ 2) orient edges
  if $V_j \notin \mathbf{S}_{ik}$ and $V_j \notin \mathbf{S}_{ki}$ **then**
    Orient edges $V_i \rightarrow V_j \leftarrow V_k$
Orient all remaining und. edges, without creating any cycle or v-structure

The second one allows for v-structure identification without further CI testing. Formally, when $\{X, Y\} \subseteq \mathbf{PC}_W$ and $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ (and therefore $Y \notin \mathbf{PC}_X$),

$$\{X, Y\} \subseteq \mathbf{PA}_W \iff W \notin \mathbf{Z}. \tag{3.12}$$

By exploiting these additional properties, we obtain the *Peter-Clark* (PC) algorithm (Algorithm 3), a more efficient variant of SGS, also named after its authors [SG91; SGS93]. The main idea behind PC is, during the skeleton identification phase, to perform first statistical tests with no conditioning set ($\mathbf{Z} = \emptyset$) to restrain the potential neighbouring sets, then consider subsets of size 1 from the current neighbouring of each tested variable, then subsets of size 2, and so on until convergence. As a result, the size of the largest conditioning set is restricted to the size of the largest current neighbouring set, which shrinks at each step. Then, during the edge orientation phase, PC checks whether a $X - W - Y$ triple constitutes a v-structures simply by inspecting the separating set that removed the $X - Y$ edge, without performing any additional CI test due to (3.12). While both SGS and PC are correct under the faithfulness assumption, PC requires much less statistical tests and smaller conditioning sets for sparse structures, and therefore is more efficient in practice.

MMPC

As we have seen with the SGS and PC algorithms, the process of identifying a faithful DAG can be decomposed into two steps, namely 1) skeleton identification; and 2) edge orientation. The *Max-Min Parents and Children* (MMPC) algorithm (Algorithm 4), proposed in [TAS03b], decomposes further the skeleton identification problem into a set of efficient local searches focused on discovering the neighbourhood of each node, i.e., its parent and children set $\mathbf{PC}$. MMPC consist in a two-phase

Algorithm 4 Max-Min Parents and Children (MMPC)

Require: $\mathbf{V} = \{V_1, \ldots, V_n\}$ a set of random variables whose joint distribution is DAG-faithful, $X \in \mathbf{V}$ a target variable, $(\cdot \perp\!\!\!\perp \cdot | \cdot)$ a CI oracle, $\text{dep}(\cdot, \cdot | \cdot)$ a dependence measure.

Ensure: $\mathbf{PC}$ a superset of the parents and children of X in a DAG faithful to $p(\mathbf{v})$.

$\mathbf{CAN} \leftarrow \mathbf{V} \setminus \{X\}$, $\mathbf{PC} \leftarrow \emptyset$
repeat ▷ 1) add true positives to $\mathbf{PC}$
  for all $Y \in \mathbf{CAN}$ **do**
    $\mathbf{Z}_Y \leftarrow \arg\min_{\mathbf{Z} \subseteq \mathbf{PC}} \text{dep}(X, Y \mid \mathbf{Z})$
    if $X \perp\!\!\!\perp Y \mid \mathbf{Z}_Y$ **then**
      Remove Y from $\mathbf{CAN}$
  $Y_{dep} \leftarrow \arg\max_{Y \in \mathbf{CAN}} \text{dep}(X, Y \mid \mathbf{Z}_Y)$
  Add Y_{dep} to $\mathbf{PC}$, remove Y_{dep} from $\mathbf{CAN}$
until $\mathbf{PC}$ does not change
$\mathbf{CAN} \leftarrow \mathbf{PC}$, $\mathbf{PC} \leftarrow \emptyset$
repeat ▷ 2) remove false positives from $\mathbf{PC}$
  for all $Y \in \mathbf{CAN}$ **do**
    $\mathbf{Z}_Y \leftarrow \arg\min_{\mathbf{Z} \subseteq \mathbf{PC} \setminus \{Y\}} \text{dep}(X, Y \mid \mathbf{Z})$
    if $X \not\perp\!\!\!\perp Y \mid \mathbf{Z}_Y$ **then**
      Add Y to $\mathbf{PC}$, remove Y from $\mathbf{CAN}$
  $Y_{ind} \leftarrow \arg\min_{Y \in \mathbf{CAN}} \text{dep}(X, Y \mid \mathbf{Z}_Y)$
  Remove Y_{ind} from $\mathbf{CAN}$
until $\mathbf{PC}$ does not change

Algorithm 5 Corrected Max-Min Parents and Children (CMMPC)

Require: $\mathbf{V} = \{V_1, \ldots, V_n\}$ a set of random variables whose joint distribution is DAG-faithful, $X \in \mathbf{V}$ a target variable.

Ensure: $\mathbf{PC}$ the set of parents and children of X in a DAG faithful to $p(\mathbf{v})$.

$\mathbf{PC} \leftarrow \text{MMPC}(X)$ ▷ 1) recover a $\mathbf{PC}$ superset
for all $Y \in \mathbf{PC}$ **do** ▷ 2) false positives correction (AND filter)
  if $X \notin \text{MMPC}(Y)$ **then**
    Remove Y from $\mathbf{PC}$

procedure: 1) a growing phase where potential neighbours are added to $\mathbf{PC}$ one at a time, the most promising one first; and 2) a shrinking phase where wrongly added neighbours are removed from $\mathbf{PC}$ one at a time, the least promising one first. Unfortunately MMPC does not assert condition (3.11), and only identifies a super-set of the neighbouring of a node [PBT05; TBA06]. Still, adding a simple symmetry correction to MMPC (Algorithm 5) allows for recovering a faithful DAG skeleton, according to the following property,

$$Y \in \mathbf{PC}_X \iff X \in \text{MMPC}(Y) \text{ and } Y \in \text{MMPC}(X). \tag{3.13}$$

While MMPC is in essence very similar to the skeleton identification phase in the PC algorithm, it is more efficient in practice [TAS03b] since it exploits not only

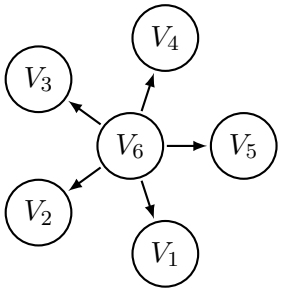


Fig. 3.3. A star-shaped DAG, prone to false negative adjacencies with MMPC.

independence relations, $X \perp\!\!\!\perp Y \mid \mathbf{Z}$, but also measures of how much dependencies hold, $\text{dep}(X, Y \mid \mathbf{Z})$. In practice MMPC implementations use the complementary of the p-value of the CI test as a dependence measure. By considering only one variable at a time, and adding/removing the most/least promising variables first, MMPC ensures that when $\mathbf{PC}$ has grown so big that in cannot grow any more (i.e., a statistical test will always accept $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ when $\mathbf{Z}$ is too large), then it will contain only the most important variables[10]. Still, the AND correction (3.13) implies that, if X and Y are adjacent in the faithful DAG, a false negative $X \notin \text{MMPC}(Y)$ can not be recovered from a true positive $Y \in \text{MMPC}(X)$. This is typically restrictive when one variable has a large neighbouring set while its neighbours have small ones. For example, consider the star-shaped DAG in Figure 3.3, where one node has 5 neighbours, while the remaining ones have only 1. On nodes V_1 to V_5, MMPC will run CI tests with a conditioning set of maximum size 1, and therefore will be able to output node V_6 as the only neighbour even with small data samples. On the other hand, on node V_6 MMPC will have to run CI tests with a conditioning set of maximum size 4, which may fail to reject the null hypothesis with not enough data samples. In such a situation, MMPC would result in an early stopping and missing neighbours for node V_6. Due to the AND correction in Algorithm 5, the false negatives in $\text{MMPC}(V_6)$ will prevail and inevitably result in missing edges in the skeleton.

IAPC

Before introducing the IAPC algorithm, we must first define the notions of *Markov blanket* and *Markov boundary*.

Markov blanket

Def. 3.1 *A* Markov blanket *of* $\mathbf{X}$ *in* $\mathbf{V}$ *is a subset* $\mathbf{M} \subseteq (\mathbf{V} \setminus \mathbf{X})$ *such that* $\mathbf{X} \perp\!\!\!\perp \mathbf{V} \setminus (\mathbf{X} \cup \mathbf{M}) \mid \mathbf{M}$. *A* Markov boundary *is an inclusion-optimal Markov blanket, i.e., none of its proper subsets is a Markov blanket.*

[10] Note that similar ordering heuristics for PC were already discussed in [SGS93][§5.4.2.4].

Algorithm 6 Incremental Association Markov Boundary (IAMB)

Require: $\mathbf{V} = \{V_1, \dots, V_n\}$ a set of random variables whose joint distribution supports the Composition property, $X \in \mathbf{V}$ a target variable, $(\cdot \perp\!\!\!\perp \cdot | \cdot)$ a CI oracle, $\text{dep}(\cdot, \cdot | \cdot)$ a dependence measure.
Ensure: $\mathbf{MB}$ a Markov boundary of X in $\mathbf{V}$.

$\mathbf{MB} \leftarrow \emptyset$
repeat ▷ 1) add true positives to $\mathbf{MB}$
  $Y \leftarrow \arg\max_{Y \in \mathbf{V} \setminus (\mathbf{MB} \cup \{X\})} \text{dep}(X, Y \mid \mathbf{MB})$
  if $X \not\perp\!\!\!\perp Y \mid \mathbf{MB}$ **then**
    Add Y to $\mathbf{MB}$
until $\mathbf{MB}$ does not change
repeat ▷ 2) remove false positives from $\mathbf{MB}$
  $Y \leftarrow \arg\min_{Y \in \mathbf{MB}} \text{dep}(X, Y \mid \mathbf{MB} \setminus \{Y\})$
  if $X \perp\!\!\!\perp Y \mid \mathbf{MB} \setminus \{Y\}$ **then**
    Remove Y from $\mathbf{MB}$
until $\mathbf{MB}$ does not change

IAMB

The *Incremental Association Markov Boundary* (IAMB) algorithm (Algorithm 6), proposed in [TAS03a], recovers a Markov boundary when the underlying distribution satisfies the Composition property [Peñ+07], which is true under the DAG-faithfulness assumption (Theorem 2.5). Moreover, in a faithful DAG the Markov boundary of a variable X is unique and is given by $\mathbf{MB}_X = \mathbf{PC}_X \cup \mathbf{SP}_X$, that is, the parents, children and spouses of X.

Under the DAG-faithfulness assumption, an additional property extends (3.9), which indicates a simple procedure to recover the $\mathbf{PC}_X$ from $\mathbf{MB}_X$,

$$Y \in \mathbf{PC}_X \iff X \not\perp\!\!\!\perp Y \mid \mathbf{Z}, \quad \forall \mathbf{Z} \subseteq \mathbf{M}_X, \tag{3.14}$$

where $\mathbf{M}_X$ is a Markov blanket of X in $\mathbf{V}$. As a direct consequence, the *Incremental Association Parents and Children* (IAPC) algorithm (Algorithm 7) [MA10b] combines IAMB to recover the Markov boundary of a variable, $\mathbf{MB}_x$, with a second pass to filter out the spouses and keep only $\mathbf{PC}_x$. Algorithms such as IAPC are sometimes termed *weak* $\mathbf{PC}$ learners, since they require a first set of CI tests with large conditioning sets to recover $\mathbf{PC}_X \cup \mathbf{SP}_X$, where MMPC recovers directly $\mathbf{PC}_X$ with smaller conditioning sets. Still, IAPC does not require the AND filter of MMPC (symmetry correction), and therefore can be made less prone to false negatives by applying an OR filter on the discovered neighbouring sets. For example, in the star-shaped DAG in Figure 3.3 IAPC with OR filtering may be more efficient than MMPC to recover the DAG skeleton.

IAPC

Algorithm 7 Incremental Association Parents and Children (IAPC)

Require: $\mathbf{V} = \{V_1, \ldots, V_n\}$ a set of random variables whose joint distribution is DAG-faithful, $X \in \mathbf{V}$ a target variable, $(\cdot \perp\!\!\!\perp \cdot | \cdot)$ a CI oracle.
Ensure: $\mathbf{PC}$ the set of parents and children of X in a DAG faithful to $p(\mathbf{v})$.
1: $\mathbf{MB} \leftarrow \text{IAMB}(X)$ ▷ 1) recover $\mathbf{MB}$
2: $\mathbf{PC} \leftarrow \mathbf{MB}$
3: **for all** $Y \in \mathbf{PC}$ **do** ▷ 2) remove spouses from $\mathbf{MB}$
4: **if** $\exists \mathbf{Z} \subseteq \mathbf{MB} \setminus \{Y\}$ s.t. $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ **then**
5: Remove Y from $\mathbf{PC}$

HPC

Finally, under the DAG-faithfulness assumption another interesting property which extends (3.9) is,

$$Y \in \mathbf{PC}_X \iff X \not\perp\!\!\!\perp Y \mid \mathbf{Z}, \quad \forall \mathbf{Z} \subseteq \mathbf{M}_Y^X, \tag{3.15}$$

where $\mathbf{M}_Y^X$ is a Markov blanket of Y in $\mathbf{M}_X \cup \{X\}$, and $\mathbf{M}_X$ a Markov blanket of X in $\mathbf{V}$. The *Hybrid Parents and Children* (HPC) algorithm (Algorithm 8) [MA10b] heavily relies on this property. Roughly speaking, HPC consists in two phases, 1) recover $\mathbf{M}_X$ a superset of $\mathbf{PC}_X \cup \mathbf{SP}_X$ with only low-order CI tests to avoid early false negatives, and 2) combine several runs of IAPC within $\mathbf{M}_X$ with an OR filter to recover $\mathbf{PC}$.

In practice, HPC often results in more accurate neighbouring sets than MMPC or IAPC, particularly for small sample sizes and/or DAGs with large $\mathbf{PC}$ sets [MA10b]. However, this improvement comes at the price of an increased computational time, due to the additional internal calls to IAPC for false negative correction.

3.4 The hybrid approach

Let us now summarize the pros and cons of both score-based and constraint-based approaches. First, Bayesian and MDL scoring functions are well-defined regardless of the data generating distribution p, or the sample size of the data set $\mathcal{D}$. In that sense the score-based approach is theoretically well-suited in every situation, and in practice it is known to produce better structures than the constraint-based approach. Its major drawback, however, is an exponential time complexity with respect to the number of variables considered, which makes it rather prohibitive for high-dimensional data sets.

Algorithm 8 Hybrid Parents and Children (HPC)

Require: $\mathbf{V} = \{V_1, \ldots, V_n\}$ a set of random variables whose joint distribution is DAG-faithful, $X \in \mathbf{V}$ a target variable, $(\cdot \perp\!\!\!\perp \cdot | \cdot)$ a CI oracle.
Ensure: $\mathbf{PC}$ the set of parents and children of X in a DAG faithful to $p(\mathbf{v})$.

$\mathbf{PCS} \leftarrow \mathbf{V} \setminus \{X\}$
for m from 0 to 1 **do** ▷ 1) recover $\mathbf{PCS}$ a superset of $\mathbf{PC}$
  for all $Y \in \mathbf{PCS}$ **do**
    if $\exists \mathbf{Z} \subseteq \mathbf{PCS} \setminus \{Y\}$ s.t. $|\mathbf{Z}| = m$ and $X \perp\!\!\!\perp Y \mid \mathbf{Z}_Y$ **then**
      Remove Y from $\mathbf{PCS}$
      $\mathbf{S}_Y \leftarrow \mathbf{Z}$
$\mathbf{SPS} \leftarrow \emptyset$
for all $W \in \mathbf{PCS}$ **do** ▷ 2) recover $\mathbf{SPS}$ a superset of $\mathbf{SP}$
  $\mathbf{SPS}_W \leftarrow \emptyset$
  for all $Y \in \mathbf{V} \setminus (\mathbf{PCS} \cup \{X\})$ **do**
    if $X \not\perp\!\!\!\perp Y \mid \mathbf{S}_Y \cup \{W\}$ **then**
      Add Y to $\mathbf{SPS}_W$
  for all $Y \in \mathbf{SPS}_W$ **do**
    if $\exists Z \in \mathbf{SPS}_W$ s.t. $X \perp\!\!\!\perp Y \mid \{W, X\}$ **then**
      Remove Y from $\mathbf{SPS}_W$
  Add $\mathbf{SPS}_W$ to $\mathbf{SPS}$
$\mathbf{PC} \leftarrow \text{IAPC}(X, \mathbf{PCS} \cup \mathbf{SPS} \cup \{X\})$ ▷ 3) recover $\mathbf{PC}$
for all $Y \in \mathbf{PCS} \setminus \mathbf{PC}$ **do** ▷ 4) false negatives correction (OR filter)
  if $X \in \text{IAPC}(Y, \mathbf{PCS} \cup \mathbf{SPS} \cup \{X\})$ **then**
    Add X to $\mathbf{PC}$

On the other hand, constraint-based approaches require two major assumptions, that is: 1) the independence model $I(p)$ is faithful to a DAG; and 2) the CI tests performed on $\mathcal{D}$ accurately reflect $I(p)$. Both of these assumptions are problematic. First, the DAG-faithfulness assumption forbids many simple kinds of interactions between the variables of interest, such as deterministic relationships which violate the Intersection property. In practice such relationships are frequent in many systems, making the DAG-faithfulness assumption rather unrealistic. Second, even when $I(p)$ is faithful to a DAG, it may very well be that the independence model extracted empirically with CI tests, $I(\mathcal{D})$, is not. As a result constraint-based methods are known to be quite unstable, and are prone to cascading effects where a single error early on in the building process can result in very a different DAG structure. This is particularly true during the edge-orientation step [SGS93]. Still, constraint-based methods are in practice rather fast, as their computational complexity relates closely to the maximum in-degree of any node in the DAG, regardless of the total size of the graph [DD99].

The idea of a hybrid approach is best formulated by Koller and Friedman [KF09][p. 839],

> "Another open direction of research attempts to combine the best of both worlds. Can we use the efficient procedures developed for constraint-based learning to find high-scoring network structure? [...] A simpleminded combination of these two approaches uses a constraint-based method to find starting point for the heuristic search. More elaborate strategies attempt to use the insight from constraint-based learning to reformulate the search space — for example, to avoid exploring structures that are clearly not going to score well, or to consider global operators."

3.4.1 Early works

Several early works attempted to combine both constraint-based and score-based approaches. In [SV93] the PC algorithm is used to generate an absolute ordering on the nodes, which allows for an efficient search procedure in the DAG space restricted to that ordering [CH91]. In [DD99] PC is run several times with different hyper-parameters (α value, CI tests ordering), and only the highest scoring DAG is kept according to some scoring function. In [FNP99] a standard greedy search procedure is used, where the parent set of each variable is restricted to a candidate set of size k, defined using a heuristic dependence measure. In [Ad00; Ad01], a greedy search procedure is used to minimize a specific cost function, that is, the conditional mutual information $I(X, Y|\mathbf{S})$ between each pair of non-adjacent nodes given their minimal d-separating set in $\mathcal{G}$. Moreover, a CI test $X \perp\!\!\!\perp Y \mid \mathbf{S}$ is used to restrict the new edge candidates during exploration. In [dFP03] the learning procedure alternates between a greedy search and a constraint-based correction, which adds/removes edges in the current DAG by performing on a series of CI tests. In [de 06], greedy search is used with a new hybrid scoring function, based on a penalized mutual information $I(X, \mathbf{PA}_X|\emptyset)$ between each node and its parents.

3.4.2 Max-Min Hill-Climbing (MMHC)

While all the above-mentioned approaches bridge the gap between score-based and constraint-based approaches, a significant break-through was made in [TBA06], with the *Max-Min Hill-Climbing* (MMHC) algorithm (Algorithm 9). While conceptually simple, MMHC combines efficient procedures from both worlds. First a skeleton is learned using the constraint-based MMPC algorithm, then a high-scoring DAG is found using a greedy search within the restricted search space of the skeleton, enhanced with a TABU list as in [FNP99] to escape local maxima. The search begins with an empty graph, and at each iteration evaluates every possible edge addition, deletion or reversal, then performs the operation that leads to the highest

Algorithm 9 Max-Min Hill-Climbing (MMHC)

Require: $\mathbf{V} = \{V_1, \ldots, V_n\}$ a set of random variables.
Ensure: $\mathcal{G}$ a high-scoring Bayesian network structure.
1: **for all** $X \in \mathbf{V}$ **do** ▷ 1) restriction phase
2: $\quad \mathbf{PC}_X \leftarrow \text{MMPC}(X) \cap \{Y | X \in \text{MMPC}(Y)\}$
3: $\mathcal{G} \leftarrow$ empty graph over $\mathbf{V}$ ▷ 2) maximization phase
4: Starting from $\mathcal{G}$, perform a TABU search with operators *add*, *delete*, and *reverse* edge. Only try operator *add* $X \rightarrow Y$ if $Y \in \mathbf{PC}_X$

scoring structure. The TABU list keeps track of the last 100 structures explored, and forbids operations which result in a structure already in the list. In order to escape local maxima, operations resulting in a score decrease are permitted, and the algorithm terminates after 15 iterations without increasing the maximum score ever encountered during search. The overall best scoring structure is then returned.

To some extent, the constraint-based approach efficiently reduces the space of candidate DAGs to consider during the score-based search, resulting in an efficient hybrid procedure which scales to distributions with thousands of variables. In [TBA06] an extensive empirical comparison of MMHC is conducted against a variety of other score-based, constraint-based and hybrid methods, namely the Peter-Clark (PC) [SG91], Sparse Candidate (SC) [FNP99], Three Phase Dependency Analysis (TPDA) [Che+02], Optimal Reinsertion (OR) [MW03], Greedy Equivalent Search (GES) [Chi02], and Greedy Search (GS) algorithms. Overall, MMHC outperforms all other approaches both in terms of quality of the reconstructed network and total running time. Although MMHC is rather heuristic by nature (it returns a local optimum of the score function), it is currently considered as the most powerful state-of-the-art algorithm for BN structure learning capable of dealing with thousands of nodes in reasonable time.

3.5 Our contribution: a new hybrid algorithm

The work presented in this section constitutes our main contribution to the field of Bayesian network structure learning, with a novel algorithm called *Hybrid HPC* (H2PC) [GAE12; GAE14]. We introduce the main motivation behind H2PC, which draws strongly on the MMHC and HPC algorithms, and measure its empirical performance against MMHC, which is currently the most powerful state-of-the-art algorithm for BN structure learning [TBA06].

3.5.1 The ideal skeleton

In order to achieve both efficiency and quality, the skeleton learned within a two-phase hybrid algorithm such as H2PC must support two properties: 1) sparsity in order to facilitate the optimization problem; and 2) sufficiency in order to ensure reachability to the score-optimal DAG $\mathcal{G}^\star$. Both these properties appear rather opposite, as an empty skeleton clearly satisfies sparsity, while a complete skeleton satisfies sufficiency. Under the DAG-faithfulness assumption, the optimal skeleton is unique and can be obtained with any of the constraint-based methods mentioned in Section 3.3. Still, in practice each of these approaches will result in a quite different structure, with potentially missing edges (false negatives) or unnecessary edges (false positives).

In [PIM08], the *Constrained Optimal Search* (COS) algorithm substitutes the greedy search of MMHC with an exact optimization procedure, which is guaranteed to return the highest scoring DAG within the skeleton-restricted search space (called a *super-structure*). As expected, COS compares favorably to MMHC in terms of DAG quality, but the additional computational cost is prohibitive for large-sized networks. An interesting discussion in [PIM08] is the following,

> "MMPC appears to be a good method to learn robust and relatively sparse skeletons; unfortunately, soundness is achieved only for high significance levels, $\alpha > 0.9$, implying a long calculation and a denser structure. Practically, when the constraint is learned with $\alpha > 0.05$, in terms of accuracy, COS is worse than OS since the superstructure is usually incomplete;"

Therefore, we believe that there is room for improvements in the skeleton identification phase of MMHC, specifically to obtain a better trade-off between false negative and false positive edges.

3.5.2 Hybrid Hybrid Parents and Children (H2PC)

The *Hybrid Hybrid Parents and Children* (H2PC) algorithm (Algorithm 10) [GAE12; GAE14] draws strongly on the MMHC algorithm, and is specifically intended to improve the constraint-based phase by learning a skeleton with HPC instead of MMPC. Since HPC is experimentally more accurate than MMPC [MA10b; VM12], especially in presence of large neighbouring sets, it should result in a better skeleton and in the end a better DAG structure after the maximization phase.

Algorithm 10 Hybrid Hybrid Parents and Children (H2PC)

Require: $\mathbf{V} = \{V_1, \ldots, V_n\}$ a set of random variables.
Ensure: $\mathcal{G}$ a high-scoring Bayesian network structure.

1: **for all** $X \in \mathbf{V}$ **do** ▷ 1) restriction phase
2: $\quad \mathbf{PC}_X \leftarrow \text{HPC}(X) \cap \{Y | X \in \text{HPC}(Y)\}$
3: $\mathcal{G} \leftarrow$ empty graph over $\mathbf{V}$ ▷ 2) maximization phase
4: Starting from $\mathcal{G}$, perform a TABU search with operators *add*, *delete*, and *reverse* edge. Only try operator *add* $X \rightarrow Y$ if $Y \in \mathbf{PC}_X$

HPC may be thought of as a way to compensate for the large number of false negatives at the output of a weak PC learner, IAPC, by performing a series extra computations. As this may arise at the expense of the number of false positives, within HPC we employ a modified of IAMB algorithm, namely IAMBFDR Peña [Peñ08], which aims at controlling the false discovery rate (FDR) in the learned Markov boundary. The resulting PC learner, IAPC-FDR, exhibits a better trade-off between false negatives and false positives, and in turn improves the quality of the neighbourhoods learned by HPC. Note that, under the DAG-faithfulness assumption, $X \in \text{HPC}(Y)$ if and only if $Y \in \text{HPC}(X)$. However, in practice this is not always true, particularly when working in high-dimensional domains [Peñ08]. In such a situation, two simple solutions exist for combining contradictory neighbouring sets, either by applying an AND filter, which decreases the number of false negatives at the cost of more false positives, or an OR filter which does the opposite. In our implementation of H2PC we opted for the first solution, which results in a better false negatives / false positives trade-off.

3.5.3 Experimental validation

In order to assess the empirical performance of H2PC, we conduct an experimental comparison of H2PC against MMHC on synthetic data sets sampled from eight well-known benchmark Bayesian networks, presented in Table 3.1. All Bayesian networks (structure and probability tables) can be downloaded from the *bnlearn* repository[11]. We do not claim that those Bayesian networks resemble real-world problems, however, they make it possible to compare the output of the algorithms with the true data-generating DAG. Ten sample sizes have been considered for training: 50, 100, 200, 500, 1000, 2000, 5000, 10000, 20000 and 50000, along with a test set with 50000 samples, generated from each BN with ancestral sampling [Bis06]. All experiments are repeated 10 times for each sample size and each BN.

We implemented H2PC in *R* [R C16], within the *bnlearn* package from Scutari [Scu10] which already contains an implementation of the MMHC algorithm[12].

[11] *http://www.bnlearn.com/bnrepository*
[12] The H2PC source code is publicly available at `https://github.com/gasse/bnlearn-clone-3.4`.

Tab. 3.1. Description of the BN benchmarks used in the experiments.

network	# var.	# edges	max. in/out degree	domain range	min/med/max PC set size
child	20	25	2/7	2-6	1/2/8
insurance	27	52	3/7	2-5	1/3/9
mildew	35	46	3/3	3-100	1/2/5
alarm	37	46	4/5	2-4	1/2/6
hailfinder	56	66	4/16	2-11	1/1.5/17
munin1	186	273	3/15	2-21	1/3/15
pigs	441	592	2/39	3-3	1/2/41
link	724	1125	3/14	2-4	0/2/17

Within both MMHC and H2PC we employed a BDeu scoring function with equivalent sample size 10, as suggested in [KF09], and a G-test of conditional independence with threshold $\alpha = 0.05$. Experiments were carried out on a machine with Intel(R) Core(TM) i5-3470M CPU @3.20 GHz 4GB RAM running Linux 64 bits.

Performance indicators

We first investigate the quality of the skeleton learned during the restriction phase, compared to the skeleton of the true data-generating DAG. From the adjacency matrix of the true skeleton and the learned skeleton, we obtain a contingency table with the number of true positives (TP, edges present in both skeletons), false positives (FP, edges present only in the learned skeleton), true negatives (TN, edges absent in both skeletons) and false negatives (FN, edges present only in the true skeleton). Based on this contingency table we report five indicators:

- the false negative rate: $\text{FN}/(\text{TP} + \text{FN})$;
- the false positive rate: $\text{FP}/(\text{TP} + \text{FP})$;
- the precision: $\text{TP}/(\text{TP} + \text{FP})$;
- the recall: $\text{TP}/(\text{TP} + \text{FN})$;
- the Euclidean distance from perfect precision and recall: $\sqrt{(1 - prec.)^2 + (1 - rec.)^2}$, as proposed in [Peñ+07].

Second, we assess the quality of the DAG learned during the maximization phase as in [SA12]. We report five indicators:

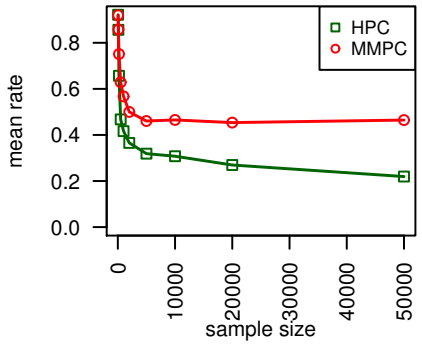
- the BDeu score on train and test data sets;
- the BIC score on train and test data sets;

- the Structural Hamming Distance (SHD) between the learned and the true data-generating DAG, that is, the minimum number of edge additions, removals and reversals required to match the two independence models.

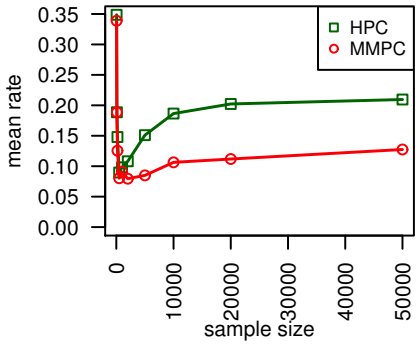
Results

In Figure 3.4, we report the quality of the skeleton obtained with HPC over that obtained with MMPC (before the maximization phase), as a function of the sample size, averaged over the 8 benchmark Bayesian networks. The increase factor for a given performance indicator is expressed as the ratio of the performance value obtained with HPC over that obtained with MMPC (the gold standard). Note that for some indicators, an increase is actually not an improvement but is worse (e.g., false positive rate, Euclidean distance). Regarding the quality of the skeleton, the advantages of HPC against MMPC are noticeable. As expected, HPC consistently increases the skeleton recall, at the cost of a little expense in precision. The overall recall/precision trade-off seems better with HPC, since the Euclidian distance is consistently improved. The main drawback of HPC is that it requires extra computations compared to MMPC, with an increasing number of CI tests. As a consequence the running time during the restriction phase of H2PC is higher than that of MMHC, within order 10 and up to 25 for large Bayesian networks / sample sizes.

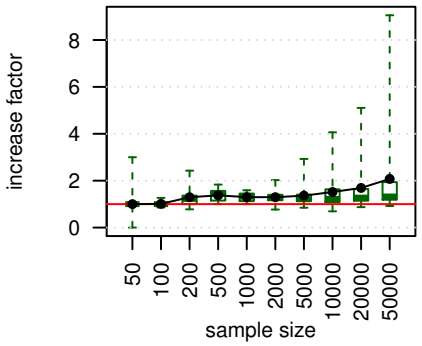
In Figure 3.5, we report the quality of the final DAG obtained with H2PC over that obtained with MMHC (after the maximization phase), as a function of the sample size, averaged over the 8 benchmark Bayesian networks. Regarding the BDeu and BIC scores on both training and test data, the improvements are noteworthy (recall that both scores take negative values, therefore the smaller the ratio the better). The results in terms of goodness of fit to training and test data using H2PC clearly dominate those obtained using MMHC whatever the sample size considered, hence its ability to generalize better. Regarding the quality of the network structure itself (i.e., how close is the DAG to the true dependence structure of the data-generating distribution), we found H2PC to perform significantly better as the sample size increases. The SHD ratio decays rapidly (lower is better), and for 50 000 samples the SHD with H2PC is on average only 50% that of MMHC. Regarding the computational burden involved, we may observe from Table 3.2 the total computational overhead of H2PC compared to MMHC. The ratio grows somewhat linearly with the sample size, with H2PC within order 10 times slower on average than MMHC with 50000 samples.



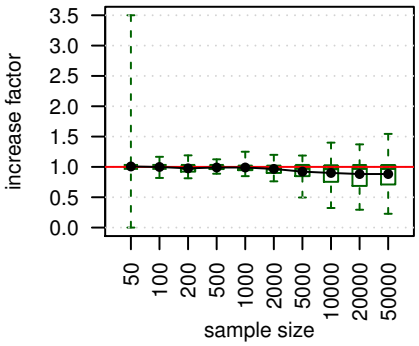
(a) False negative rate (lower is better).



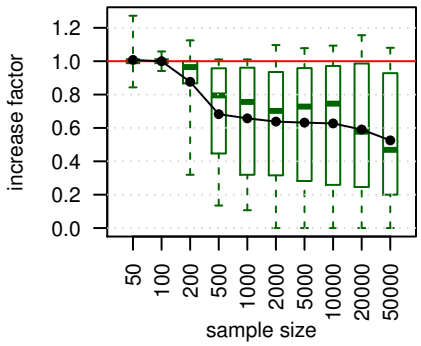
(b) False positive rate (lower is better).



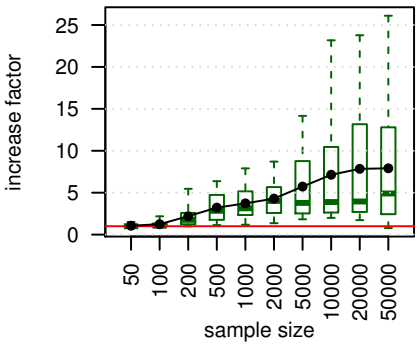
(c) Recall (higher is better).



(d) Precision (higher is better).

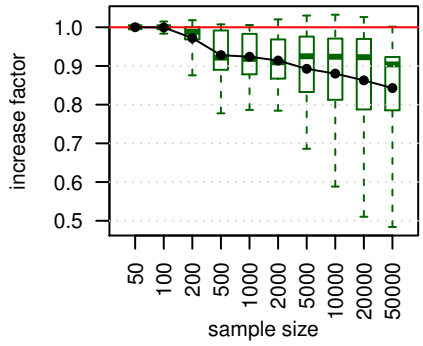


(e) Euclidean distance (lower is better).

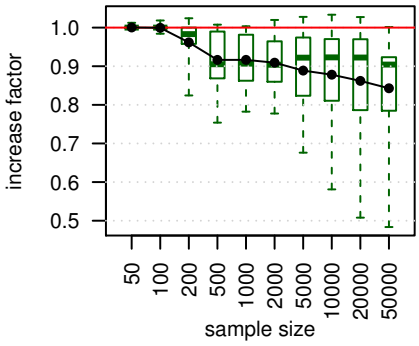


(f) Number of CI tests performed.

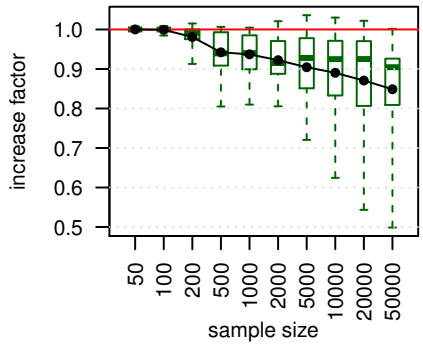
Fig. 3.4. Skeleton quality. The top two figures present quality measures for both HPC and MMPC, while the remaining figures present the ratio HPC / MMPC. Black lines indicate mean values, while boxplots indicate quartiles and most extreme values.



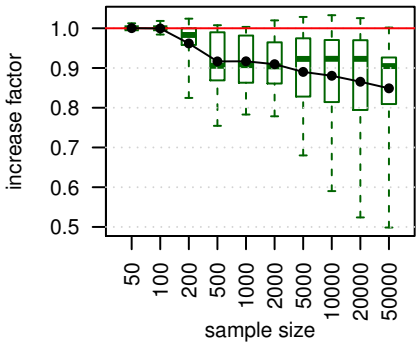
(a) BDeu on training set (lower is better).



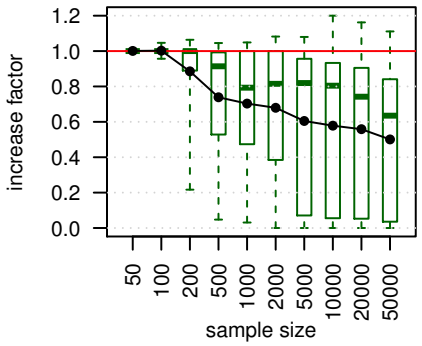
(b) BDeu on test set (lower is better).



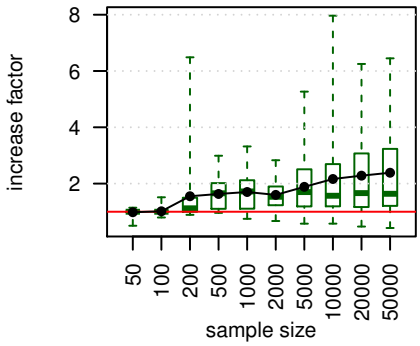
(c) BIC on training set (lower is better).



(d) BIC on test set (lower is better).



(e) SHD (lower is better).



(f) Number of search iterations.

Fig. 3.5. DAG quality. All figures present the ratio HPC / MMPC. Black lines indicate mean values, while boxplots indicate quartiles and most extreme values.

Tab. 3.2. Total running time ratio (H2PC / MMHC).

Network	Sample Size									
	50	100	200	500	1000	2000	5000	10000	20000	50000
child	0.94 ±0.1	0.87 ±0.1	1.14 ±0.1	1.99 ±0.2	2.26 ±0.1	2.12 ±0.2	2.36 ±0.4	2.58 ±0.3	1.75 ±0.6	1.78 ±0.5
insurance	0.96 ±0.1	1.09 ±0.1	1.56 ±0.1	2.93 ±0.2	3.06 ±0.3	3.48 ±0.4	3.69 ±0.3	4.10 ±0.4	3.76 ±0.6	3.75 ±0.5
mildew	0.77 ±0.1	0.80 ±0.1	0.79 ±0.1	0.94 ±0.1	1.01 ±0.1	1.23 ±0.1	1.74 ±0.2	2.14 ±0.2	3.26 ±0.6	6.20 ±1.0
alarm	0.88 ±0.1	1.11 ±0.1	1.75 ±0.1	2.43 ±0.1	2.55 ±0.1	2.71 ±0.1	2.65 ±0.2	2.80 ±0.2	2.49 ±0.3	2.18 ±0.6
hailfinder	0.85 ±0.1	0.85 ±0.1	1.40 ±0.1	1.69 ±0.1	1.83 ±0.1	2.06 ±0.1	2.13 ±0.1	2.12 ±0.2	1.95 ±0.2	1.96 ±0.6
munin1	0.77 ±0.0	0.85 ±0.0	0.93 ±0.0	1.35 ±0.0	2.11 ±0.0	4.30 ±0.2	12.92 ±0.7	23.32 ±2.6	24.95 ±5.1	24.76 ±6.7
pigs	0.80 ±0.0	0.80 ±0.0	4.55 ±0.1	4.71 ±0.1	5.00 ±0.2	5.62 ±0.2	7.63 ±0.3	11.10 ±0.6	14.02 ±1.7	11.74 ±3.2
link	1.16 ±0.0	1.93 ±0.0	2.76 ±0.0	5.55 ±0.1	7.04 ±0.2	8.19 ±0.2	10.00 ±0.3	13.87 ±0.4	15.32 ±2.5	24.74 ±4.2
all	0.89 ±0.1	1.04 ±0.1	1.86 ±1.2	2.70 ±1.5	3.11 ±1.9	3.71 ±2.2	5.39 ±4.0	7.75 ±7.3	8.44 ±8.4	9.64 ±9.7

3.5.4 Discussion

We discussed a hybrid algorithm for Bayesian network structure learning called Hybrid HPC (H2PC), intended for improving the state-of-the-art H2PC algorithm from Tsamardinos et al. [TBA06], with a better skeleton identification phase. Our extensive experiments showed that the skeleton learned by H2PC reduces the number of missing edges without sacrificing the number of extra edges, which is crucial for the soundness of two-stage hybrid methods [PIM08; Koj+10]. As a result H2PC outperforms MMHC by a significant margin in terms of structure quality in almost every situation, at the cost of a higher computational overhead, within order 10. Still, we showed experimentally that H2PC is scalable to problems with several hundreds of variables and large sample sizes, and therefore should be preferred over MMHC when affordable.

The main drawback of H2PC being its demanding computational time, it is worth noting that in our experiments HPC was run independently on each node, without keeping track of the (in)dependencies previously found. This clearly leads to some loss of efficiency due to redundant calculations, and we believe that the computational cost of H2PC may be reduced by using a cache to store the result of previous CI tests, or by optimizing the inner IAMB / IAPC procedures within HPC for an early stopping when appropriate.

Finally, we believe that constraint-based approaches for Bayesian network structure learning could benefit from theoretical analysis under milder assumptions than DAG-faithfulness, as in [Peñ+07] where IAMB was shown to require only the Composition property. Unfortunately, such theoretical studies are rather few. Still, we observe that each of the constraint-based approaches discussed in Section 3.4 require at least the Composition property, as none of these is able to recover the skeleton of a DAG $A \rightarrow C \leftarrow B$ with an exclusive OR relationship ($p(C = A \oplus B) = \alpha$). This is also true of any greedy score-based approach that starts from an empty DAG, such as GES. As a consequence, the Composition property seems to play an important role in Bayesian structure learning, and PGM structure learning in general, as it allows for the design of efficient procedures. Conversely, complex relationships (i.e., which do not imply pairwise relationships) seem very challenging for structure learning.

4 Multi-label classification

> *When we try to pick out anything by itself we find that it is bound fast by a thousand invisible cords that cannot be broken, to everything in the universe.*
>
> **— John Muir**
> 1869

In a general setting, supervised machine learning consists in learning a mapping from an input space $\mathcal{X}$ to an output space $\mathcal{Y}$, given a set of examples $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{n}$. The problem is well-studied in the uni-dimensional output space context, leading to standard classification problems for discrete-valued outputs (i.e. $\mathbf{y} \in \{c_1, \ldots, c_m\}$), or regression problems for continuous outputs (i.e. $\mathbf{y} \in \mathbb{R}$). In situations where the output space is multi-dimensional, the literature typically refers to the problem as multivariate prediction, multivariate output learning, or structured output prediction. Multi-label classification (MLC) refers to a specific class of multivariate prediction problems, in situations where all output variables are binary (i.e. $\mathbf{y} \in \{0, 1\}^m$).

Multi-label classification has received an increasing attention in the last years from the machine learning community. This setting corresponds to classification problems with non-mutually exclusive classes, which is encountered in many recent real-world problems, including image indexing and annotation [Wan+14], facial expression analysis [Wan+15; Zha+15b], text categorization [AAN15], sentiment analysis [LC15], fault-control [Li+15], drug side effects prediction [Zha+15a], genome-wide protein function assignment [Han+15; WHZ14], and early detection of chronic diseases [Zuf+15] to cite a few.

So far, there is a consensus among researchers that, to improve the performance of multi-label learning algorithms, label dependencies have to be incorporated into the learning process [Lua+12; TV07; GG11; ZZ10; CMM12; Rea+09; BRR98; Koc+07; BLL11; Cor+14]. Indeed, the problem of learning a mapping $\mathcal{X} \mapsto \mathcal{Y}$ from data is tightly related to the problem of modeling $p(\mathbf{y}|\mathbf{x})$, and therefore exploiting the (in)dependence structure between the labels seems a good idea to help in modeling their distribution. However, multi-label learning is often cast as a loss-minimization problem, and Dembczynski et al. [Dem+12] reminds us that one cannot expect

the mapping to be optimal for different types of losses at the same time. Most importantly, the expected benefit of exploiting label dependence depends on the loss function to be minimized, e.g., the popular label-wise decomposable *Hamming loss* does not require to model such dependencies. On the other hand, the label-wise non-decomposable *subset* $0/1$ *loss* clearly benefits from modeling label dependencies. An open question remains: given a particular loss function, what shall we capture exactly from the statistical relationships between labels to solve the multi-label classification problem?

Since we are interested in learning the structure of probabilistic graphical models, the multi-label classification problem seems a good setting to study and apply structure learning algorithms in this context. This chapter is intended to present a formal introduction the multi-label classification problem, its challenges, and a state of the art of the main approaches that can be found in the literature.

4.1 Supervised learning

In this chapter, we place ourselves in the context of fully-supervised multi-label learning. Formally, given a set of data samples $\mathcal{D} = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \dots, (\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}$ drawn independently from a joint distribution $p(\mathbf{x}, \mathbf{y})$, we want to find a mapping $\mathbf{h} : \mathcal{X} \to \mathcal{Y}$ such that, on average, if we draw a new sample $(\mathbf{x}, \mathbf{y})$ from p, $\mathbf{h}(\mathbf{x})$ will be close to $\mathbf{y}$. This definition resumes very well the idea of supervised learning, and appears to be rather precise. However, it raises one question: what does "close" mean ? Closeness requires a notion of distance in the output space $\mathcal{Y}$, which is often defined as a mapping to a positive cost $\mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_{\geq 0}$, a.k.a. a loss function $L(\mathbf{h}(\mathbf{x}), \mathbf{y})$. Finding the best mapping then boils down to minimizing the expected loss over $p(\mathbf{x}, \mathbf{y})$, and thus the best mapping for a particular loss function may not be the best for some other loss function. Although this is true for all machine learning problems, it is even more true in the context of multi-label learning, as we will see.

4.1.1 Risk minimization

Formally, the risk of a particular mapping $\mathbf{h}$ under a particular loss function L is defined as the expected loss over the joint distribution,

$$R_L(\mathbf{h}) = \mathbb{E}_{\mathbf{x},\mathbf{y}}[L(\mathbf{h}(\mathbf{x}), \mathbf{y})],$$

which translates to

$$\int_{\mathbf{x},\mathbf{y}} L(\mathbf{h}(\mathbf{x}), \mathbf{y}) p(\mathbf{x}, \mathbf{y}) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}.$$

Given a particular input sample $\mathbf{x}$, the point-wise risk-minimizing prediction $\mathbf{h}^\star(\mathbf{x})$, a.k.a the Bayes-optimal prediction, is given by

$$\mathbf{h}^\star(\mathbf{x}) = \underset{\mathbf{h}(\mathbf{x})}{\arg\min} \int_{\mathbf{y}} L(\mathbf{h}(\mathbf{x}), \mathbf{y}) p(\mathbf{y}|\mathbf{x}) \mathrm{d}\mathbf{y}. \tag{4.1}$$

To alleviate our notation, in the following we will consider without loss of generality that $\mathbf{x}$ is fixed, so that we can write $\mathbf{h}(\mathbf{x}) = \mathbf{h}$ and $p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y})$. Also, note that the Bayes-optimal prediction $\mathbf{h}^\star$ is not necessarily unique, thus the $\in$ symbol should be preferred to $=$ in (4.1). For simplicity, in the following we will omit this ambiguity and always refer to *the* Bayes-optimal prediction.

Complexity

Let us consider the complexity of solving equation (4.1) in the multi-label classification setting, that is, $\mathbf{y} \in \{0, 1\}^m$ a binary-valued output space. First, evaluating the risk of an arbitrary mapping under an arbitrary loss function requires to model $p(\mathbf{y})$, which amounts to estimating $2^m - 1$ parameters in the worst case. Then, finding the risk-minimizing prediction $\mathbf{h}^\star$ is achieved by evaluating all 2^m combinations of $\mathbf{h}$, each requiring in turn a summation over all 2^m combinations of $\mathbf{y}$. The overall parameter complexity of multi-label classification is then $O(2^m)$, and the inference complexity $O(2^{2m})$.

In this thesis, our main contribution is based on the idea of exploiting the dependence structure between the labels to decompose the expression of the risk-minimizer in (4.1), and simplify both the estimation of the parameters during the learning phase and the evaluation of the optimal prediction during the inference phase. Clearly the benefit of this approach is highly dependent of the loss function to be minimized, as for some loss functions the expression of the risk naturally decomposes into simple terms, regardless of the dependence structure between the labels.

Decomposable loss

Consider a loss function that decomposes into a sum of terms over each of the labels, i.e. a label-wise decomposable loss function $L(\mathbf{h}, \mathbf{y}) = \sum_{i=1}^m L_i(h_i, y_i)$. Such

a decomposition is present, for example, in the popular *Hamming loss*. Then, the risk of a particular output $\mathbf{h}$ becomes

$$\sum_{\mathbf{y}} p(\mathbf{y}) \sum_{i=1}^{m} L_i(h_i, y_i) = \sum_{i=1}^{m} \sum_{\mathbf{y}} p(\mathbf{y}) L_i(h_i, y_i).$$

From the chain rule of probabilities we can write

$$\sum_{\mathbf{y}} p(\mathbf{y}) L_i(h_i, y_i) = \sum_{y_i} p(y_i) L_i(h_i, y_i) \sum_{\mathbf{y}_{-i}} p(\mathbf{y}_{-i}|y_i),$$

where $\mathbf{y}_{-i}$ denotes the $\mathbf{y}$ vector deprived of its i-th element y_i. The right-most term vanishes (it sums to one by definition), and finally the risk-minimizing output is given by

$$\mathbf{h}^\star = \arg\min_{\mathbf{h}} \sum_{i=1}^{m} \sum_{y_i} L_i(h_i, y_i) p(y_i).$$

Note that knowing only the $p(y_i)$ terms is sufficient to solve this problem, and the estimation of the full joint distribution $p(\mathbf{y})$ is no longer required. Moreover, since $min(a + b) = min(a) + min(b)$, the problem naturally decomposes into m independent minimization problems, one for each label. Overall, the parameter complexity is reduced from $O(2^m)$ to $O(m)$, and the inference complexity is reduced from $O(2^{2m})$ to $O(4m)$.

Non-decomposable loss

But what if the loss function does not decompose at all? This is where modeling the label dependence structure can help. Obviously, taking into account the independence relations between the labels (more exactly the conditional independence relations given the features) will reduce the number of parameters required to estimate the joint distribution $p(\mathbf{y}|\mathbf{x})$, and make the learning step easier. However, we can not say that it will in general reduce the inference complexity. Suppose that we find a partition $\{\mathbf{Y}_1, \mathbf{Y}_2\}$ of the label set with respectively m_1 and m_2 labels, such that $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{Y}_2 \mid \mathbf{X}$. Then the point-wise risk-minimizing prediction is given by

$$\mathbf{h}^\star = \arg\min_{\mathbf{h}} \int_{\mathbf{y}_1} p(\mathbf{y}_1) \int_{\mathbf{y}_2} p(\mathbf{y}_2) L(\mathbf{h}, \mathbf{y}).$$

The number of required parameters to estimate is reduced from $2^m - 1$ to $2^{m_1} + 2^{m_2} - 2$, however solving the minimization problem still requires the evaluation of 2^{2m} combinations of $(\mathbf{h}, \mathbf{y})$, thus the inference complexity remains the same.

What can we conclude from these examples? First, the complexity of the multi-label learning problem is highly dependent on the loss function we seek to minimize.

Considering a particular loss function can reduce the number of parameters to estimate from the training set, which improves the generalization capacity of the model, and can also reduce the inference complexity, which improves the scalability of the model in high dimensional output spaces. Also, it is worth to notice that in some situations it is not required to consider the dependencies between the labels. In the best case, i.e. a label-wise decomposable loss function as shown above, modeling marginal distributions is sufficient, regardless of the structure of the joint distribution $p(\mathbf{y})$. Similarly, for the F-*loss* function (the complement of the F-measure) it was shown that the parameter complexity can be reduced $O(m^2)$, and the inference complexity to $O(m^3)$, regardless of the dependency structure between the labels [Dem+11]. In general, the problem of establishing whether, and how, the risk of a particular loss function decomposes is not trivial, and remains an open question.

4.1.2 Multi-label loss functions

A variety of evaluation measures have been presented in the literature to cover the different requirements encountered in practical multi-label classification problems, see for instance [TKV10]. Commonly used measures for assessing the performance of MLC algorithms include the *Hamming loss*, *subset* $0/1$ *loss* [Dem+10], *accuracy*, *precision*, *recall*, F*-measure*, *one-error*, *coverage*, *average precision* [TKV10], and more recently the *balanced error rate* [SB15]. This is not surprising as MLC applications have different goals and requirements. In e-discovery applications, all the relevant documents should be retrieved, so recall is the most relevant measure. In Web search, on the other hand, precision should be as important as the recall, so the F-measure might be more appropriate [PC10]. In many usual quiz-based examinations provided by Massive open on-line courses (MOOCs), assessment is based on multiple choice questions, so the subset $0/1$ loss is more appropriate. In the following we will review the most common loss functions for multi-label classification.

Hamming loss

Certainly the most intuitive and commonly used loss function in the multi-label setting is the *Hamming loss*. Formally, the Hamming loss is defined as

$$L_H(\mathbf{h}, \mathbf{y}) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{I}(y_i \neq h_i),$$

where $\mathrm{I}(\cdot)$ is the standard $\{False, True\} \rightarrow \{0, 1\}$ mapping.

Clearly, the Hamming loss is label-wise decomposable, and thus theoretically does not benefit from modeling label dependencies. It is easily shown that the point-wise risk-minimizing prediction is given by the mode of the marginal distribution of each label,

$$h_i^* = \arg\max_{y_i} p(y_i).$$

As a result, both the parameter and inference complexity of multi-label classification under the Hamming loss is $O(m)$.

Multi-label classification is often reformulated as a multivariate regression problem, in order to apply calculus-based machine learning algorithms, such as support vector machines or feed-forward neural networks. A popular alternative definition of the Hamming loss is then $\frac{1}{m}\|\mathbf{y} - \mathbf{h}\|_2^2$, where $\|\cdot\|_2$ is the Euclidean distance, a.k.a $L2$-norm. Indeed, the squared $L2$-norm reduces to a sum of squared absolute values, $\frac{1}{m}\sum_i |y_i - h_i|^2$, which is equivalent to $\frac{1}{m}\sum_i \mathrm{I}(y_i \neq h_i)$ in binary output spaces. In recent works [Bor+15; CH16], minimizing the empirical loss is often reformulated as a minimization problem over $\|\mathbf{Y} - \mathbf{H}\|_F^2$, where $\|\cdot\|_F^2$ denotes the squared Frobenius norm, $\mathbf{Y}$ the matrix of training samples, and $\mathbf{H}$ the matrix of predictions.

Note that, in general, any loss function which is expressed as $\frac{1}{m}\|\mathbf{y} - \mathbf{h}\|_p^p$ reduces to the Hamming loss in the case of a binary output space, with $\|\cdot\|_p$ the Lp-norm and $p \in \mathbb{R}_{>0}$. Any such loss function is clearly label-wise decomposable, and thus theoretically does not require to take into account label dependencies.

Subset zero-one loss

Another common loss function in multi-label classification is the *subset zero-one loss*. Formally, the subset zero-one loss is defined as

$$L_S(\mathbf{h}, \mathbf{y}) = \mathrm{I}(\mathbf{y} \neq \mathbf{h}).$$

This loss function appears to be overly stringent, with a very singular notion of distance in the output space. The distance between any pair of points in $\mathcal{Y}$ is always 1, except in the particular situation where $\mathbf{h} = \mathbf{y}$ and it is 0. Subset zero-one loss is known to be a particularly difficult loss function for risk-minimization. However, in our work it will be of particular interest since it is not label-wise decomposable. The point-wise risk-minimizing prediction is given by the mode of the joint distribution of the labels, a.k.a. the maximum a-posteriori estimate (MAP), or most probable expectation (MPE)

$$\mathbf{h}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}).$$

As a result, both the parameter and inference complexity of multi-label classification under the subset zero-one loss are $O(2^m)$.

Again, multi-label classification under *subset zero-one loss* may be cast as a multivariate regression problem. In general, any loss function which is expressed as $\lim_{q \to 0} \|\mathbf{y} - \mathbf{h}\|_p^q$ reduces to the subset zero-one loss function in the case of a binary output space, with $\|\cdot\|_p$ the Lp-norm and $p \in \mathbb{R}_{>0}$.

Recall, precision, accuracy

In binary classification, it is common to synthesize the performance of a classifier over a sample set in a contingency table. Given a data set $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$ and a vector of predicted values $(h^{(1)}, \dots, h^{(n)})$, such that $h^{(i)} = h(\mathbf{x}^{(i)})$, a contingency table reports the number of $(h^{(i)}, y^{(i)})$ combinations corresponding to true positive $(1, 1)$, true negative $(0, 0)$, false positive $(1, 0)$ and false negative $(0, 1)$ predictions.

Tab. 4.1. A binary contingency table.

		h	
		0	1
y	0	tn	fp
	1	fn	tp

recall

Several evaluation measures can then be extracted from the contingency table, such as the *recall*, i.e. the ratio of correct predictions among the positive observations,

$$\text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \in [0, 1],$$

precision

the *precision*, i.e. the ratio of correct predictions among the positive predictions,

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \in [0, 1],$$

accuracy

and the *accuracy*, i.e. the ratio of correct predictions overall,

$$\text{Accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}} \in [0, 1].$$

In a multi-label setting with m labels, given a data set $\mathcal{D}$, the observed and predicted values take the form of two $n \times m$ matrices $(\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)})$ and $(\mathbf{h}^{(1)}, \dots, \mathbf{h}^{(n)})$, and building a contingency table appears to be less straightforward. Three types of contingency tables are usually considered, based either on:

- global counts, i.e. $\sum_{i=1}^n \sum_{j=1}^m \mathrm{I}((h_j^{(i)}, y_j^{(i)}) = (\cdot, \cdot))$;

- label-wise counts for a given label Y_j, i.e. $\sum_{i=1}^{n} \mathrm{I}((h_j^{(i)}, y_j^{(i)}) = (\cdot, \cdot))$;
- instance-wise counts for a given sample $\mathbf{y}^{(i)}$, i.e. $\sum_{j=1}^{m} \mathrm{I}((h_j^{(i)}, y_j^{(i)}) = (\cdot, \cdot))$.

Although several evaluation metrics have been proposed in the literature based on global and label-wise contingency tables [ZZ14], in a standard risk-minimizing framework we are interested in loss functions that are point-wise defined, so in this work we will consider only the last option, i.e. performance measures based on instance-wise contingency tables. In the following we will refer to the recall, precision and accuracy as instance-wise functions, i.e.

$$\mathrm{rec}(\mathbf{h}, \mathbf{y}) = \frac{\mathbf{h} \cdot \mathbf{y}}{\mathbf{y} \cdot \mathbf{y}}, \quad \mathrm{prec}(\mathbf{h}, \mathbf{y}) = \frac{\mathbf{h} \cdot \mathbf{y}}{\mathbf{h} \cdot \mathbf{h}}, \quad \mathrm{acc}(\mathbf{h}, \mathbf{y}) = \frac{\mathbf{h} \cdot \mathbf{y}}{m},$$

where $\cdot$ denotes the dot product operator and $0/0 = 1$ by definition.

Note that these measures are not distance metrics, but rather similarity metrics, which translate to proper loss functions in their complementary form, i.e. $1 - \mathrm{rec}(\mathbf{h}, \mathbf{y})$, $1 - \mathrm{prec}(\mathbf{h}, \mathbf{y})$ and $1 - \mathrm{acc}(\mathbf{h}, \mathbf{y})$. Note that the accuracy is a commonly used metric for risk-minimization, as it reduces to the popular Hamming loss in its complementary form. On the other hand, the recall and precision considered individually do not provide interesting loss functions, as their Bayes-optimal prediction would be respectively $\mathbf{h}^\star = \mathbf{1}$ and $\mathbf{h}^\star = \mathbf{0}$ regardless of the distribution $p(\mathbf{y})$.

F-measure

A popular loss function that combines recall and precision is the so-called F_β *loss*, which is usually introduced in its complementary form the F_β measure. Formally, the F_β measure is defined in terms of recall and precision as

$$F_\beta = \left(\frac{\alpha}{\text{Precision}} + \frac{1 - \alpha}{\text{Recall}} \right)^{-1} \in [0, 1],$$

where $\alpha = 1/(1 + \beta^2)$. According to Van Rijsbergen [Van79, Chapter 7], F_β was derived so that it "measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision". In the multi-label learning context, the formulation of the instance-wise F_β loss simplifies to

$$L_{F_\beta}(\mathbf{h}, \mathbf{y}) = 1 - \frac{(1 + \beta^2) \times \mathbf{h} \cdot \mathbf{y}}{\mathbf{h} \cdot \mathbf{h} + \beta^2 \times \mathbf{y} \cdot \mathbf{y}},$$

where $\cdot$ denotes the dot product operator and $0/0 = 1$ by definition. When $\beta = 1$ ($\alpha = 1/2$), F_1 reduces to the harmonic mean of precision and recall, which gives

equal importance to both terms. The resulting measure is known as the Dice coefficient [Dic45], which is most often simply called the F-measure, or F-score.

Recently, Dembczynski et al. [Dem+11] showed that solving equation (4.1) for F-measure maximization does not require to model the full joint distribution $p(\mathbf{y})$, but only a specific distribution $p(y_i, s_\mathbf{y})$ for every label, where $s_\mathbf{y} = \mathbf{y} \cdot \mathbf{y}$. As a result, the General F-measure Maximizer (GFM) algorithm for multi-label classification under the F-loss has a parameter complexity in $O(m^2)$, and an inference complexity in $O(m^3)$.

Jaccard index

Another well-known loss function in multi-label classification is the *Jaccard distance*, whose complementary form is the Jaccard index, a.k.a. Jaccard similarity coefficient. The expression of the Jaccard index closely resembles that of the F-measure, and is formally defined in terms of recall and precision as

$$\text{Jaccard} = \left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}} - 1\right)^{-1} \in [0, 1].$$

In the multi-label learning context, the formulation of the instance-wise Jaccard distance simplifies to

$$L_J(\mathbf{h}, \mathbf{y}) = 1 - \frac{\mathbf{h} \cdot \mathbf{y}}{\mathbf{h} \cdot \mathbf{h} + \mathbf{y} \cdot \mathbf{y} - \mathbf{h} \cdot \mathbf{y}},$$

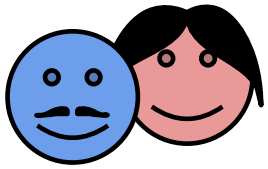
where $\cdot$ denotes the dot product operator and $0/0 = 1$ by definition.

It remains an open question whether or not a closed-form solution for the risk minimizer of the Jaccard similarity exists, but the problem seems far from straightforward [Wae+14], and one commonly believes that exact optimization is intractable in general [Chi+10].

4.1.3 Illustration

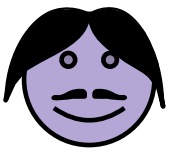
Let us now illustrate the influence of choosing a particular loss function in the general context of multivariate prediction.

Tab. 4.2. Alice and Bob.



a	b	$p(\mathbf{y}\|\mathbf{x})$	$\mathbb{E}_{\mathbf{y}}[L(\mathbf{h}(\mathbf{x}),\mathbf{y})]$				
			L_H	L_S	L_{F_1}	$\|\cdot\|_1^2$	$\|\cdot\|_2^1$
0	0	.02	.87	.99	.99	3.27	1.30
0	1	.11	.49	.88	.37	1.21	0.93
1	0	.12	.50	.89	.38	1.25	0.94
1	1	.76	**.12**	**0.24**	**.09**	**0.27**	**.24**

Tab. 4.3. Alice or Bob?



a	b	$p(\mathbf{y}\|\mathbf{x})$	$\mathbb{E}_{\mathbf{y}}[L(\mathbf{h}(\mathbf{x}),\mathbf{y})]$				
			L_H	L_S	L_{F_1}	$\|\cdot\|_1^2$	$\|\cdot\|_2^1$
0	0	.02	.53	.98	.98	1.22	1.01
0	1	.46	.49	**.54**	.49	1.86	**0.72**
1	0	.44	.51	.56	.51	1.94	0.75
1	1	.08	**.47**	.92	**.32**	**0.98**	0.93

In multi-label classification

Consider a face recognition problem, where the task is to determine the presence or absence of two people, Alice and Bob, in a picture. We formulate this problem as a supervised multi-label learning problem, with $\mathbf{X}$ the random variable representing the pictures and $\mathbf{Y} = (A, B)$ the random variable that indicates the presence or absence of Alice and Bob ($\mathbf{y} = (a, b) \in \{0, 1\}^2$). We will assume a perfect model of the conditional distribution $p(\mathbf{y}|\mathbf{x})$, from which we will infer Bayes-optimal predictions, and we will consider the two particular inputs presented in Tables 4.2 and 4.3. Each time we present the joint probability distribution of the labels given the input, $p(\mathbf{y}|\mathbf{x})$, and compute the risk of each label combination under several loss function, namely the Hamming loss L_H, the subset zero-one loss L_S, the F_1 loss L_{F_1}, the squared Manhattan distance $\|\cdot\|_1^2$ and the Euclidean distance $\|\cdot\|_2^1$. Each time the minimal risk is indicated in bold font.

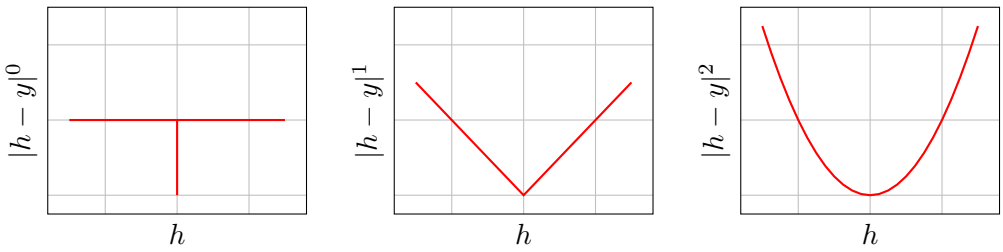
Given the first input, presented in Table 4.2, the uncertainty is rather small in the distribution of the labels $p(\mathbf{y}|\mathbf{x})$, and the mode $(1, 1)$ clearly gathers most of the probability density. In this situation the Bayes-optimal prediction is the same for all loss functions, that is $\mathbf{h}^*(\mathbf{x}) = (1, 1)$.

Given the second input, shown in Table 4.3, $p(\mathbf{y}|\mathbf{x})$ expresses a higher uncertainty, and the labels appear to share a mutual exclusion relationship. Obviously, either Alice or Bob is present in the picture, but not both at the same time. As expected, the Bayes-optimal prediction under L_S corresponds to the mode of the joint distribution, $(0, 1)$, while under L_H it corresponds to the mode of the marginal distribution of each label, $(1, 1)$. Indeed, we have that $p(a = 1|\mathbf{x}) = 0.52$ and $p(b = 1|\mathbf{x}) = 0.54$.

The Bayes-optimal prediction under L_{F_1} coincides with that of L_H, but we know this shall not always be the case. Finally, it appears that $\|\cdot\|_1^2$ coincides with L_H, while $\|\cdot\|_2^1$ coincides with L_S. Since $\lim_{q\to 0}\|\cdot\|$ is equivalent to L_S, we believe that somehow a loss function in the form $\|\cdot\|_p^q$ put more emphasis more on the marginal losses of the labels when $q > p$, and more on the joint loss of the labels when $q < p$.

We believe that both decomposable and non-decomposable loss functions are of interest, and any loss function is valid as long as it is meaningful for the problem at hand. For example, in the picture in Table 4.3 it is interesting to know that i) Alice or Bob may be present in the picture, and also ii) only one person is present in the picture. Under Hamming loss, we have information i) but not ii). Under subset zero-one loss we have part of information i) and part of information ii). Necessarily, when making a prediction one has to compress a whole distribution $p(\mathbf{y})$ into a single answer $\mathbf{h}^*$, which implies loosing some information. Somehow the loss function decides which information is the most important to keep.

What can we conclude from this example? Well, as we already stated, when the relation between $\mathbf{X}$ and $\mathbf{Y}$ is deterministic (i.e. $p(\mathbf{y}|\mathbf{x})$ is a Dirac), then choosing one loss function or another yields the same prediction, thus it makes sense to use the most convenient one (such as Hamming loss). When this relation is non-deterministic, you have two options. You can either i) consider that there is intrinsically a deterministic relationship in the problem at hand, but you only have noisy data. In that case you can work on your data to remove that noise and/or add additional input to make the relationship deterministic. Or ii) you can consider that the relationship is intrinsically non-deterministic, in which case the choice of a particular loss function is not harmless. We believe that in many machine learning situations this relation is non-deterministic by nature, and maybe people in the machine learning community should pay more attention to the loss function they use and its consequences, especially in the multi-variate setting. A popular challenge in machine learning recently is face completion [Den+09], where face images are split into two parts, and the goal is to reconstruct the right part of the image given the left part. Obviously there is not always enough information given by one side of an image to predict exactly the opposite side. Maybe then the consistency between the predicted pixels is more important that the marginal accuracy of each individual pixel?



(a) Zero-one error. **(b)** Absolute error. **(c)** Squared error.

Fig. 4.1. Illustration of different loss functions in univariate regression (y fixed).

In multivariate regression

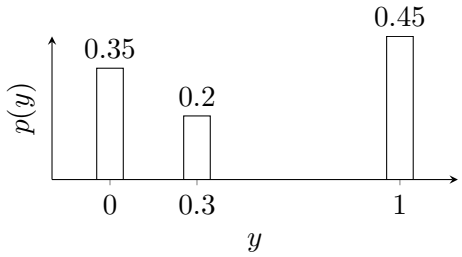
squ. error
abs. error
0/1 error

In uni-variate regression problems, the most common loss function is the *squared error* $|h-y|^2$, which is often chosen not because it is a meaningful distance measure, but because it has nice mathematical properties such as everywhere differentiability. However, one might consider other loss functions such as the *absolute error* $|h-y|^1$, or the *zero-one error* $|h-y|^0$, which yields 1 if $h=y$ and 0 otherwise[1].

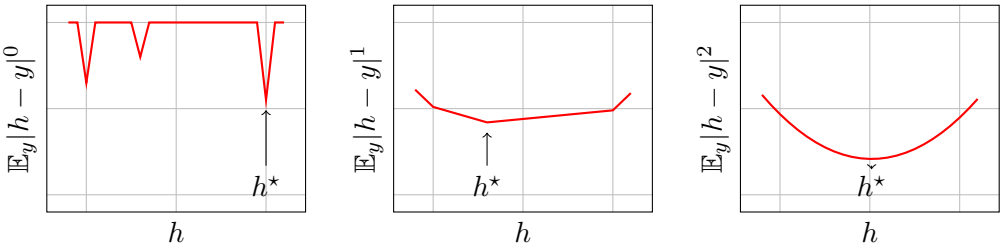
A common misconception is that the Bayes-optimal prediction h^* under any of these loss functions is the same. This is clearly not true. Indeed, with y fixed, the minimum of all these loss functions is found at the same value of h. However, in the standard supervised learning framework, it is the *expected* loss (4.1) over all the possible values of y that is minimized. Unless the relation between y and $\mathbf{x}$ is deterministic (i.e. $p(y|\mathbf{x})$ is a Dirac), the risk-minimizing prediction under different loss functions does not necessarily coincide. It turns out that the optimal prediction under the squared, absolute and zero-one error is respectively given by the mode, the median, and the mean value of Y. In probability distributions for which these characteristics coincide, e.g. Gaussian distributions, the Bayes-optimal predictions h^* will be the same under these loss functions. However this may not be the case in general distributions, as seen in Figure 4.2. Thus, even in the uni-variate output setting, choosing a particular loss function over another is not harmless and has a direct impact on the Bayes-optimal prediction.

In multivariate regression problems, a.k.a multi-output regression, the same situation occurs. Typically, the most common loss function is the squared Euclidean distance $\|\mathbf{h}-\mathbf{y}\|_2^2$ [BF97], which reduces to the Hamming loss in binary output spaces. However, other loss functions may be considered, such as $\|\mathbf{y}-\mathbf{h}\|_2^0$ which reduces to the subset zero-one loss in binary output spaces. And why not $\|\mathbf{y}-\mathbf{h}\|_1^2$ the squared Manhattan distance, or $\|\mathbf{y}-\mathbf{h}\|_2^1$ the Euclidean distance ? Any such loss function is

[1]Note that this convenient formulation of the zero-one error requires to define $0^0=0$, otherwise it is expressed as $\lim_{q\to 0}|h-y|^q$.



(a) True point-wise distribution of y.



(b) Expected zero-one error. **(c)** Expected absolute error. **(d)** Expected squared error.

Fig. 4.2. Illustration of different loss expectations in univariate regression. The corresponding Bayes-optimal prediction $h^\star$ is respectively given by the mode (1), the median (0.3) and the mean (0.51).

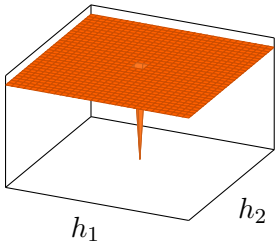
expressed as the q-th power of the L_p norm of $\mathbf{y} - \mathbf{h}$, which takes the general form of

$$\|\mathbf{y} - \mathbf{h}\|_p^q = \left(\sum_{j=1}^{m} |h_j - y_j|^p \right)^{\frac{q}{p}}.$$

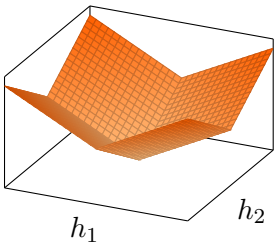
Clearly, in situations where $p = q$ such a loss function decomposes over the labels and does not require to model label dependencies. When $q = 0$, it does not decompose over the labels and explicitly requires to model $p(\mathbf{y})$, however it is very a difficult loss function to minimize. Interestingly, in other settings $(0 < p \neq q > 0)$ this loss function does not decompose over the labels either, and thus somehow must take into account label dependencies. To the best of our knowledge, such loss functions did not receive much attention so far, and we believe they may be worth studying. We also believe that finding risk-minimizing predictions under these loss functions must be more feasible that under subset zero-one loss. A straightforward minimization method may be the standard back-propagation algorithm to reach a local-minima of the risk, the partial derivative of these loss functions having the general form

$$\frac{\partial \|\mathbf{y} - \mathbf{h}\|_p^q}{\partial h_i} = q(h_i - y_i)|h_i - y_i|^{p-2} \left(\sum_{j=1}^{m} |h_j - y_j|^p \right)^{\frac{q}{p}-1}.$$

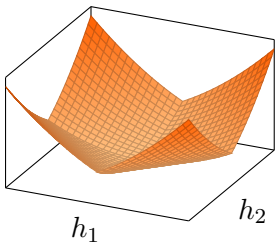
Figure 4.4 illustrates the impact of minimizing different loss functions in multivariate regression, with an image reconstruction task. Clearly choosing a decomposable



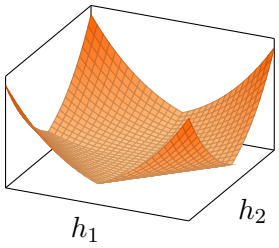
(a) $||\mathbf{h} - \mathbf{y}||_p^0$ (zero-one).



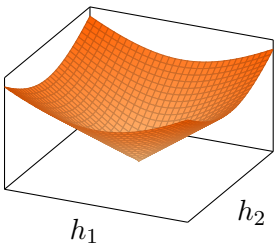
(b) $||\mathbf{h} - \mathbf{y}||_1^1$ (Manhattan).



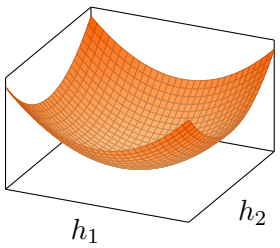
(c) $||\mathbf{h} - \mathbf{y}||_1^2$.



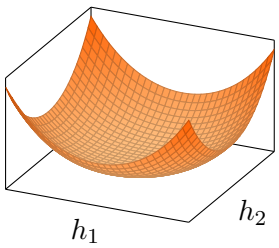
(d) $||\mathbf{h} - \mathbf{y}||_1^3$.



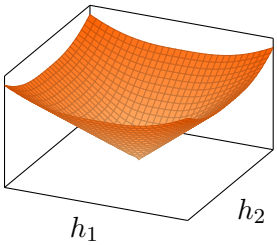
(e) $||\mathbf{h} - \mathbf{y}||_2^1$ (Euclidean).



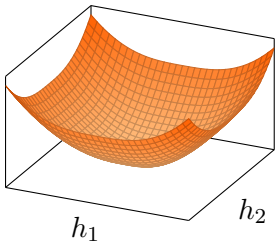
(f) $||\mathbf{h} - \mathbf{y}||_2^2$.



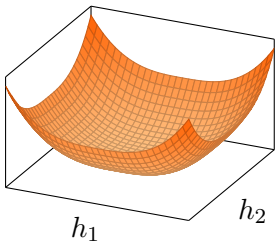
(g) $||\mathbf{h} - \mathbf{y}||_2^3$.



(h) $||\mathbf{h} - \mathbf{y}||_3^1$.



(i) $||\mathbf{h} - \mathbf{y}||_3^2$.



(j) $||\mathbf{h} - \mathbf{y}||_3^3$.

Fig. 4.3. Illustration of different loss functions in multivariate regression, with $m = 2$ ($\mathbf{y}$ fixed). Loss functions in the diagonal ((b), (f) and (j)) are label-wise decomposable.

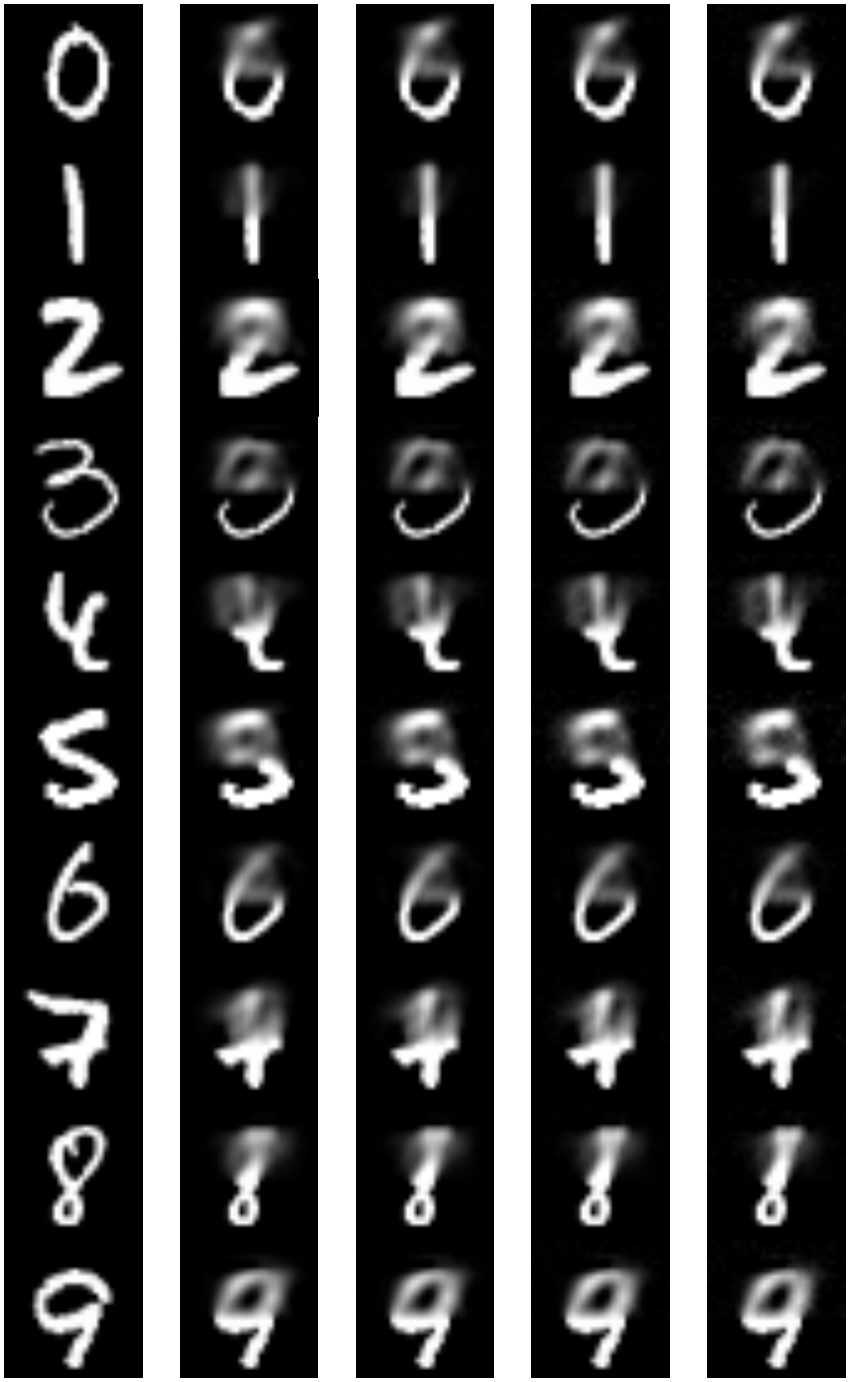


Fig. 4.4. Illustration of different loss functions for image reconstruction on MNIST. The problem is cast as a multivariate regression problem, where the top 60% pixels of the image are learned from the remaining lower 40% (i.e. $\mathcal{X} = \mathbb{R}^{12\times 28}$, $\mathcal{Y} = \mathbb{R}^{16\times 28}$). From left to right we respectively display the true pixel values of 10 test images, and the pixels recovered by minimizing the $\|\cdot\|_2^2$, $\|\cdot\|_2^1$, $\|\cdot\|_2^{0.5}$ and $\|\cdot\|_2^{0.1}$ loss functions with a linear model and batch stochastic gradient-descent, i.e. tanh perceptrons. Clearly the loss function has an effect on the global consistency between the predicted pixels, with the decomposable MSE loss function $\|\cdot\|_2^2$ producing highly blurred images, while the $\|\cdot\|_2^{0.1}$ loss function, closer to zero-one loss, produces less blurry images.

loss function such as the MSE is questionable in this setting, as the conditional dependency between the pixels is not taken into account in the output. Deciding on an alternative loss function in this context is far from trivial [WB09]. An interesting approach was recently proposed by Goodfellow et al. [Goo+14], which consists in training simultaneously two adversarial models: a first model which generates images, and a second models which discriminates these images from the original images in the dataset. As training progresses, the second model learns a specific loss function, which the first model seeks to minimize. This approach is very convenient as it solves at once both the problem of generating images and defining a loss function that makes sense in this context. This generative approach (i.e. sample images from $p(\mathbf{y})$) is easily extended to discriminative learning (i.e. sample images from $p(\mathbf{y}|\mathbf{x})$), and shows promising results [LKC15; RMC15].

4.2 Meta-learning approaches

Many approaches to multi-label classification intend to exploit label dependence by combining several models learned independently over different input/output spaces, which we call meta-learning[2]. Meta-learning approaches usually follow an intuitive scheme, such as chaining ($\mathcal{X} \mapsto \mathcal{Y}_1$, then $\mathcal{X} \times \mathcal{Y}_1 \mapsto \mathcal{Y}_2$, ...), stacking ($\mathcal{X} \mapsto \mathcal{Y}$, then $\mathcal{Y} \mapsto \mathcal{Y}$), ensemble learning, or a combination of these. Most often such approaches are motivated by intuitive rather than theoretical justifications, e.g. chaining must take into account label correlations, or stacking must correct wrong label co-occurrences. Moreover, a concrete connection between the resulting model and the loss to be minimized is rarely established, implicitly giving the misleading impression that the same method can be optimal for different loss functions [Dem+12]. And yet, meta-learning approaches can lead to competitive results in terms of several popular loss functions such as Hamming loss or subset zero-one loss, sometimes even better than the Bayes-optimal approach BR or LP. This observation is interesting, and we will give a brief discussion on this matter after we introduced the main meta-learning schemes.

4.2.1 Binary Relevance

BR

The most simple and intuitive approach to multi-label classification is the so-called *binary relevance* (BR) method, which decomposes the MLC problem into m independent binary problems, i.e. one independent classifier for every label,

$$\mathbf{h}_{BR}(\mathbf{x}) = \arg\max_{\mathbf{y}} \prod_{i=1}^{m} p(y_i \mid \mathbf{x}).$$

[2] In the literature these are sometimes called problem transformation methods[TK07]

The BR scheme is intuitive and conveniently simple to implement, and indisputably yields Bayes-optimal predictions under the Hamming loss. Still, it is often criticized for its strong independence assumption, i.e., it ignores label dependencies. This argument is often found as a ground truth in the literature, and comes with the idea that BR can be improved by incorporating label dependencies into the learning scheme [CMM12], without much consideration about which loss function is to be minimized. A comprehensive discussion on this matter can be found in [Dem+12].

However, in some experimental studies it can be found that BR is not such a weak approach, as it frequently yields competitive results in terms of complex loss functions such as the F-measure or the subset zero-one loss [Lua+12], although it is clearly not designed to minimize such loss functions. This surprising situation is somewhat similar to that of the Naive Bayes approach in the context of standard classification, which is also based on independence assumptions. In both cases, when the learning problem is too hard or when the training samples are too few, a strong biased estimator may be preferable to a weak unbiased one, i.e. $\max_y p(y) \prod_{x_i} p(x_i|y)$ instead of $\max_y p(y|\mathbf{x})$ for NB, and $\max_{\mathbf{y}} \prod_{y_i} p(y_i|\mathbf{x})$ instead of $\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$ for BR.

4.2.2 Label Powerset

A second intuitive approach to multi-label classification is the so-called *label powerset* (LP) method, which deals with the MLC problem as a standard classification problem, by considering each possible label combination as a particular class,

LP

$$\mathbf{h}_{LP}(\mathbf{x}) = \arg\max_{\mathbf{y}} p(\mathbf{y} \mid \mathbf{x}).$$

The LP scheme obviously corresponds to exact MAP inference, and therefore is tailored to minimize the subset zero-one loss. Although this approach may seem unfeasible due to a potentially exponential number of classes (2^m label combinations in the worst case), in practice it can perform reasonably well even on data sets with large label sets (up to the hundreds). This is because in usual multi-label data sets the number of positive labels per example is relatively low, resulting in many label combinations which never occur in practice, and an effective number of classes much lower than the exponential capacity.

4.2.3 Chaining

A very intuitive meta-learning scheme for MLC is the chaining approach, which gave rise to several popular instantiations. The most straightforward one is the so-called
CC *classifier chain* (CC) method proposed in [Rea+09; Rea+11], which consists in chaining m binary classifiers, one per label, in a fixed predefined order so that each model incorporates the previous labels as additional input features, i.e. $h_1(\mathbf{x})$, $h_2(\mathbf{x}, y_1)$, $h_3(\mathbf{x}, y_2, y_3)$, and so on. During the learning phase each classifier is built independently from the training samples to minimize the classification error, resulting in the following mappings,

$$h_{CC}^{learned}(\mathbf{x}, \mathbf{y}_{<i})_i = \arg\max_{y_i} p(y_i \mid \mathbf{x}, \mathbf{y}_{<i}),$$

where $\mathbf{y}_{<i}$ denotes the $i-1$ first labels in the chaining. At inference time the label terms $\mathbf{y}_{<i}$ are obviously not available, so the predictions are made iteratively according to the chaining order and the label values are substituted by the output of the previous classifiers in the chaining. The resulting $\mathcal{X} \mapsto \mathcal{Y}$ mapping is expressed as a recursive combination of the learned mappings, that is,

$$h_{CC}^{final}(\mathbf{x})_i = h_{CC}^{learned}(\mathbf{x}, \mathbf{h}_{CC}^{final}(\mathbf{x})_{<i})_i,$$

where $\mathbf{h}_{CC}^{final}(\mathbf{x})_{<i}$ is the output of the $i-1$ first classifiers in the chaining. Despite the simplicity of this approach, it is not clear which loss function is minimized at the end by CC. However, it is possible to compute tight upper-bounds of the worst-case regret of CC with respect to the Hamming loss and the subset zero-one loss, which is done in [DWH12]. As a result it appears that the regret is quite high in both cases, suggesting that CC can yield a poor performance for both loss functions. Nevertheless, CC seems to be more appropriate for the subset zero-one loss than the Hamming loss, with a lower worst-case regret.

BCC A first variant of CC is the so-called *Bayesian chain classifier* (BCC) model proposed in [Zar+11; Suc+14], where the chaining follows a particular BN structure learned from training data to encode a dependence structure among the labels. The chaining scheme is essentially the same, except that i) the chain ordering now follows the DAG ordering, and ii) each classifier incorporates only the parents of the target label in the DAG as additional features, which reduces the input space of each classifier compared to CC. The DAG skeleton is typically restricted to be a Chow-Liu tree [CL68], and a root node is picked at random to define a chaining order. This method shows competitive results compared to BR, without a significant improvement over CC. Admittedly the learned DAG structure can only represent unconditional label dependence, and just like CC the loss function minimized in the end remains unknown.

A very similar approach is followed by Zhang and Zhang [ZZ10], who learn a BN structure over the residuals of binary classifiers. The proposed method, called
LEAD *multi-label Learning by Exploiting lAbel Dependency* (LEAD), starts by building a BR model, and continues by learning a BN structure on the error residuals of the labels. Then, a the chain of classifiers is trained according to the DAG structure, as in BCC. The essential difference with LEAD is that the resulting DAG encodes a *conditional* independence model for $p(\mathbf{y} \mid \mathbf{x})$, where the DAG in BCC encodes a *marginal* dependence model for $p(\mathbf{y})$. The independence model resulting from the DAG in LEAD can be interpreted as a perfect map under two assumptions: i) $p(\mathbf{y} \mid \mathbf{x})$ is faithful to a DAG; and ii) the classification error on each label is independent of the input features. However, just like CC and BCC, the loss function minimized at the end by LEAD is unknown.

BR+ A different approach is proposed in [CMM12], called *binary relevance plus* (BR+). This approach does not depend on a particular chaining order, but can still be considered a chaining approach in our view. As in CC, m binary classifiers are learned from the training samples to minimize the classification error, but this time each model incorporates all the other labels as additional input features. Each classifier results in the following mapping,

$$h_{BR+}^{learned}(\mathbf{x}, \mathbf{y}_{-i})_i = \arg\max_{y_i} p(y_i \mid \mathbf{x}, \mathbf{y}_{-i}),$$

where $\mathbf{y}_{-i}$ denotes the $\mathbf{y}$ vector deprived of its i-th element the label y_i. At inference time the label terms $\mathbf{y}_{-i}$ are substituted by the output of a BR classifier, following a two-stage process. The resulting $\mathcal{X} \mapsto \mathcal{Y}$ mapping expresses a combination of the following mappings,

$$h_{BR+}^{final}(\mathbf{x})_i = h_{BR+}^{learned}\mathbf{x}, \mathbf{h}_{BR}(\mathbf{x})_{-i})_i,$$

where $\mathbf{h}_{BR}^{\star}(\mathbf{x})_{-i}$ is the output of the BR classifiers for every label except y_i. Again, the loss function minimized at the end by BR+ is unknown.

4.2.4 Stacking

A second intuitive scheme for meta-learning is the stacking approach, which resembles to chaining for inference, but follows a different learning scheme. The main idea of stacking is to train several layers of models on top of each other, e.g. $\mathbf{h}^1(\mathbf{x})$, then $\mathbf{h}^2(\mathbf{x}, \mathbf{h}^1(\mathbf{x}))$ and so on. One such approach is adopted in [GS04], where the output of a first BR classifier is reused as an input feature by a second BR classifier,

along with the full original feature set. This approach is sometimes called *meta stacking* (MS), and is expressed as follows,

MS

$$\mathbf{h}_{MS}(\mathbf{x}) = \arg\max_{\mathbf{y}} \prod_{i=1}^{m} p(y_i \mid \mathbf{x}, \mathbf{h}_{BR}(\mathbf{x})).$$

Note that MS closely resembles the BR+ approach, however in BR+ the label values used as input to train the second layer of classifiers come from the training set, i.e. $(\mathbf{x}, \mathbf{y}_{-i}) \rightarrow y_i$, while in MS they come from BR predictions, i.e. $(\mathbf{x}, \mathbf{h}_{BR}(\mathbf{x})) \rightarrow y_i$. Clearly MS can be considered as a BR classifier with additional feature functions, and minimizes the Hamming loss by design.

A different stacking approach is adopted in [Wan+14], where a Bayesian Network is stacked on top of BR predictions to perform MAP inference. This approach, which we call BN+, results in the following mapping,

BN+

$$\mathbf{h}_{BN+}(\mathbf{x}) = \arg\max_{\mathbf{y}} p(\mathbf{y} \mid \mathbf{h}_{BR}(\mathbf{x})).$$

The BN structure and parameters are learned from the both the label values and the BR predictions to model $p(\mathbf{y}, \mathbf{h}_{BR}(\mathbf{x}))$, and exact MAP inference is performed. Under the condition that $\mathbf{h}_{BR}(\mathbf{x})$ captures all the relevant information from the feature set to predict the labels, BN+ yields Bayes-optimal predictions under the subset zero-one loss. This condition is provably not guaranteed in general. Nevertheless, BN+ is clearly tailored towards subset zero-one loss minimization.

4.2.5 Ensemble learning

Essentially, the ensemble learning scheme consists in learning several $\mathcal{X} \mapsto \mathcal{Y}$ mappings, which are aggregated according to a predefined voting scheme to form a final mapping $\mathbf{h}^{ens}$. That is, $\{\mathbf{h}^{(j)}\}_{j=1}^{s}$ is an ensemble of mappings learned from data, and $\mathbf{f} : \mathcal{Y}^s \rightarrow \mathcal{Y}$ is an aggregation function, the voting scheme, such that $\mathbf{h}^{ens} = \mathbf{f}(\mathbf{h}^{(1)}, \ldots, \mathbf{h}^{(s)})$. In the context of multi-label classification, adopting a particular voting scheme can be seen as minimizing the risk of a particular loss function over the ensemble of predictors, that is,

$$\mathbf{h}^{ens}(\mathbf{x}) = \arg\min_{\hat{\mathbf{y}}} \sum_{j=1}^{s} L^{ens}(\hat{\mathbf{y}}, \mathbf{h}^{(j)}(\mathbf{x})).$$

This minimization problem is somewhat similar to the MLC problem itself, with $2^m - 1$ parameters to estimate in the worst case. Clearly, adopting a particular voting scheme without considering which loss function it minimizes is not harmless. Even if the individual mappings in $\{\mathbf{h}^{(j)}\}$ are learned to minimize a particular loss function, the actual loss function minimized at the end by the ensemble will also

depend on the voting scheme. Adopting a majority vote over each individual label will bring the final prediction to the marginals, in a BR fashion, while a majority vote over the global label combinations will bring the predictions to a MAP estimate, in a LP fashion. Using the same loss function for both training and voting seems a reasonable choice (e.g. BR voting over BR models), however a combination of two different loss functions may result in an ensemble model that is much harder to interpret (e.g. BR voting over LP models).

ECC

A straightforward instantiating of ensemble learning is adopted in Read et al. [Rea+09], who introduces the *ensemble of classifier chains* (ECC) method in order to strengthen CC. Essentially, ECC combines the prediction of several CC models trained over different random label orderings and different random training populations. The voting scheme adopted by ECC is a label-wise calibrated majority vote, that is,

$$h_{ECC}(\mathbf{x})_i = I\left(\frac{1}{s}\sum_{j=1}^{s} h_{CC}^{(j)}(\mathbf{x})_i \geq t\right),$$

where I$(\cdot)$ is the standard $\{False, True\} \rightarrow \{0, 1\}$ mapping and t is a fixed threshold in $[0, 1]$. While ECC significantly improves the performance of CC for several evaluation measures, it is even less clear which loss function it actually minimizes.

RAkEL

A second popular approach to MLC which uses ensemble learning is the so-called *RAndom k-labELsets* (RAkEL) method proposed in [TV07; TKV11]. RAkEL was initially proposed in order to take into account label correlations, while at the same time avoiding the computational burden of LP. Essentially, it consists in training several LP classifiers over randomly drawn and possibly overlapping label subsets of fixed size k, e.g. (Y_1, Y_3), (Y_2, Y_5), (Y_3, Y_4) and so on when $k = 2$. At inference time the ensemble of predictions are aggregated in the same way ECC does, with a marginal calibrated majority vote, by considering for each label only the classifiers which include that label as an output. It is rather difficult to state which loss function RAkEL minimizes at the end, however it constitutes a standard MLC methods which prove to be very effective under several evaluation measures, especially the Hamming loss.

ML-kNN

Interestingly, several other MLC methods rely on a voting scheme, although not in the context of ensemble learning. These include for example the *multi-label k-nearest neighbours* (ML-kNN) method from [ZZ07], which extends the standard k-nearest neighbour method to multi-label classification. Here a voting function is employed to aggregate the label combinations observed for k nearest neighbours in the training set, i.e. $\{\mathbf{y}^{(j)} | \mathbf{x}^{(j)} \in k\text{NN}(\mathbf{x})\}$. If we consider these label observations as an empirical estimate of $p(\mathbf{y}|\mathbf{x})$, then ML-kNN minimizes exactly the loss corresponding to the voting scheme. The default instantiation of ML-kNN proposed in [ZZ07]

adopts a marginal MAP voting scheme, and therefore is tailored for Hamming loss minimization.

Another set of approaches which rely on a voting scheme are those based on samples estimates, such as the *conditional dependency network* (CDN) model in [GG11], or the *probabilistic classifier chain* (PCC) model in [DWH12] with Monte-Carlo inference[3]. Both approaches learn a probabilistic model of $p(\mathbf{y}|\mathbf{x})$, and are able to produce at inference time a set of samples $\{\mathbf{y}^{(j)}\}$ drawn from this joint conditional distribution given a particular input $\mathbf{x}$. Once a sufficient number of samples is obtained, applying a voting scheme theoretically yields a Bayes-optimal prediction under the corresponding loss function. Admittedly, the major drawback of such approaches is that minimizing an arbitrary loss function requires a lot of samples to obtain a proper estimate of $p(\mathbf{y}|\mathbf{x})$, which in turn makes the voting process harder.

Monte-Carlo sampling

4.2.6 Discussion

As stated previously, the motivation behind meta-learning approaches most often comes from intuitive rather than theoretical justifications, sometimes without paying much attention to which loss function is to be minimized. However, such approaches can perform well in practice for several popular loss functions, sometimes better than a theoretically Bayes-optimal approach (e.g. BR for Hamming loss or LP for subset zero-one loss). A very common interpretation is then "approach A beats approach B by modeling label dependencies", which is rather ambiguous in our view. We may now give three potential explanations for why such approaches can result in improved classification performance, which do not relate to label dependence.

First, the (unknown) loss function minimized by a meta-learning approach can be interpreted as a surrogate for a more complex loss function, which would be too difficult to minimize directly. Given that the regret between the surrogate and the target loss is not too high, learning a biased but robust model can be preferable to learning a theoretically Bayes-optimal but weaker model. This can explain why such approaches sometimes outperform a theoretically Bayes-optimal approach for complex loss functions, such as the F-loss or the subset zero-one loss.

Second, constructing a high-level mapping from a composition of lower-level ones provides a larger hypothesis space and leads to richer models, a.k.a. deep models. While the BR approach is Bayes-optimal for the Hamming loss, in practice it involves finding an optimal mapping $\mathcal{X} \mapsto \mathcal{Y}$ among a limited hypothesis space, for example by considering only linear separations with logistic regression models. By stacking a second layer of mapping $\mathcal{Y} \mapsto \mathcal{Y}$ on top of BR, also with linear separations, the

[3] Both CDN and PCC are describe in detail in Section 4.3

hypothesis space considered in the final $\mathcal{X} \mapsto \mathcal{Y}$ mapping includes non-linear separations, resulting in a much richer model. Building rich functions from a composition of many simpler functions is the basic idea of deep learning [LBH15], which prove to be a very efficient approach [LT16] achieving state-of-the-art performance in many complex classification tasks.

Third, several meta-learning approaches follow an ensemble learning scheme with majority voting. Ensemble learning is known to improve classification performances when the voters are diverse, which is achieved in standard classification by learning several models from different input spaces and different training populations, e.g., random feature selection [Ho95] and random instance bagging [Bre96]. In multi-label classification the output space is multi-dimensional as well, so diversity can also be achieved by learning mappings to different output spaces (the RAkEL approach), or by following different chaining orders (the ECC approach).

4.3 Plug-in approaches

Plug-in approaches deal with the problem of finding $\mathbf{h}^\star$ with an explicit two-stage process: i) learning, which consists in training a probabilistic model of $p(\mathbf{y}|\mathbf{x})$; and ii) inference, which consists in applying a decision rule at test-time on the probability estimates to derive Bayes-optimal predictions $\mathbf{h}^\star(\mathbf{x}) = \arg\min_{\hat{\mathbf{y}}} \int_{\mathbf{y}} L(\hat{\mathbf{y}}, \mathbf{y}) p(\mathbf{y}|\mathbf{x}) \mathrm{d}\mathbf{y}$. This decomposition allows a principled analysis of the resulting mapping, and one can deal with each sub-problem independently, that is, any probabilistic model can be used to solve the estimation problem, and any inference algorithm can be used to solve the decision problem. However, the main drawback of such approaches is a potentially high cost for inference, that is, for each input $\mathbf{x}$ a new decision problem must be solved. To deal with this problem, plug-in approaches most often resort on biased probabilistic models for which exact inference is tractable (e.g. tree-structured models), or approximate inference algorithms (e.g. loopy belief propagation). In the following we will review several popular approaches which belong to the plug-in family, with a particular focus on the underlying probabilistic independence model.

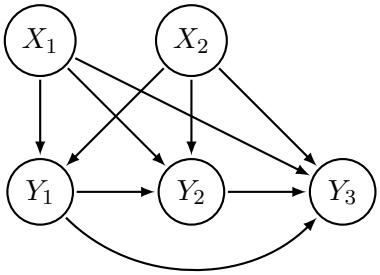


Fig. 4.5. DAG representation of a PCC model, which imposes no structural constraint on $p(\mathbf{y}|\mathbf{x})$.

4.3.1 Probabilistic classifier chains

PCC

Introduced by Dembczynski et al. [DCH10], *probabilistic classifier chain* (PCC) models simply apply the chain rule of probabilities to represent a joint conditional distribution $p(\mathbf{y}|\mathbf{x})$ from a set of marginal conditional distributions,

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{m} p(y_i \mid \mathbf{x}, \mathbf{y}_{<i}),$$

where $\mathbf{y}_{<i}$ denotes the $\mathbf{y}$ vector deprived of all elements indexed from i to m. Such a representation nicely decomposes the learning problem into m local distribution estimation problems, which may be solved independently, and obtaining probability estimates from $p(\mathbf{y}|\mathbf{x})$ is simple, with a linear complexity in the number of labels. However, the inference problem with PCC remains unchanged. In [DCH10], PCC is applied to multi-label classification with an exact MAP inference procedure in $O(2^m)$, thereby minimizing the subset zero-one loss. Due to the high complexity of inference, in practice such an approach can not be applied to problems with more than tens of labels. In [Kum+12; DWH12] approximate inference procedures are discussed, based on a sampling-voting scheme[4] with fixed sample size, or a beam-search on the tree structure of PCC for mode approximation with theoretical guarantees.

Interestingly, PCC models can be easily extended into what we will call *conditional Bayesian networks*. Consider a BN model of the joint distribution $p(\mathbf{x}, \mathbf{y})$, in which the label nodes are not allowed to appear as children of feature nodes. Then, the conditional distribution of the labels is clearly expressed as

$$p(\mathbf{y}|\mathbf{x}) = \prod_{Y_i \in \mathbf{Y}} p(y_i|\mathbf{pa}_{Y_i}),$$

which is not affected by the induced subgraph over $\mathbf{X}$. Therefore we may impose it to be empty without adding any constraint to $p(\mathbf{y}|\mathbf{x})$. By making this conditional BN fully-connected (except for the edges between features), we obtain an unconstrained conditional probabilistic model, PCC, which follows the ordering of the labels in the

[4] As discussed in Section 4.2.5.

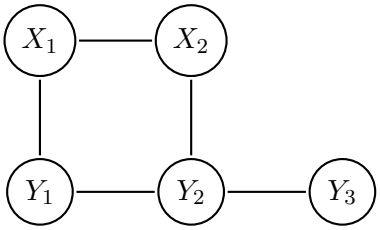


Fig. 4.6. A DN of the joint distribution $p(\mathbf{x}, \mathbf{y})$, whose local distributions are $p(x_1|x_2, y_1)$, $p(x_2|x_1, y_2)$, $p(y_1|x_1, y_2)$, $p(y_2|x_2, y_1, y_3)$, and $p(y_3|y_2)$.

DAG (see Figure 4.5). Of course, conditional BN models with sparse structures may also be considered in order to facilitate the inference problem.

4.3.2 Conditional dependency networks

DN

Introduced by Heckerman et al. [Hec+00], *dependency networks* (DNs) form an interesting family of graphical models, which encode a joint distribution $p(\mathbf{v})$ indirectly via Gibbs sampling. Formally, a DN consists in an undirected[5] graphical structure $\mathcal{G}$ over $\mathbf{V}$, together with a set of parameters that encode the conditional probability distribution of each variable given its neighbours, $p(v_i|\mathbf{nb}_{V_i})$. The independence model of a DN is exactly that of a Markov network, and is read from $\mathcal{G}$ by using the u-separation criterion.

Under the strict positivity condition ($p > 0$), a random Gibbs sampling procedure from the local distributions $p(v_i|\mathbf{nb}_{V_i})$ is guaranteed to converge to samples drawn from the probability distribution $p(\mathbf{v})$. Starting from any initial state $\mathbf{v}$, random Gibbs sampling consists in repeatedly picking a variable at random, say V_i, and updating its value by sampling from $p(v_i|\mathbf{nb}_{V_i})$. After a sufficient number of iterations, the final sample $\mathbf{v}$ is guaranteed to follow the joint distribution $p(\mathbf{v})$. The correctness DNs is due to the independence model encoded in $\mathcal{G}$, which implies that $p(v_i|\mathbf{v}_{-i}) = p(v_i|\mathbf{nb}_{V_i})$.

CDN

Dependency networks are easily extended to *conditional dependency networks* (CDNs), which model a conditional probability distribution $p(\mathbf{y}|\mathbf{x})$. It suffices to encode only the local conditional distributions of the nodes in $\mathbf{Y}$, which makes the adjacencies between the features $\mathbf{X}$ useless in $\mathcal{G}$. The Gibbs sampling procedure is then essentially the same, except that $\mathbf{x}$ is fixed and only $\mathbf{y}$ samples are produced, which are guaranteed to be drawn from the probability distribution $p(\mathbf{y}|\mathbf{x})$. The clear advantage of using a CDN for multi-label classification is that the learning process is relatively simple. Each local probability distribution can be estimated independently by

[5] Note that we consider here the family of *consistent* DNs, which can be described with undirected graphs. General DNs, including non-consistent ones, are usually described with directed graphs in which multiple edges and cycles are allowed, and local distributions in the form $p_i(v_i|\mathbf{pa}_{V_i})$.

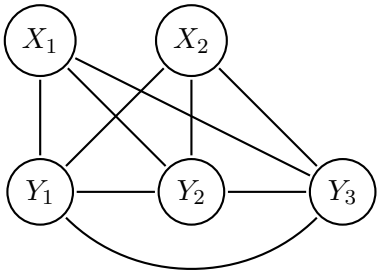


Fig. 4.7. A fully-connected CDN, which imposes no structural constraint on $p(\mathbf{y}|\mathbf{x})$.

solving a binary probabilistic classification problem, for which plenty off-the-shelf practical solutions exist. However, inference with CDNs is far from trivial. First, the computational cost of Gibbs sampling can be rather high, with a number of samples required to obtain a good estimate of $p(\mathbf{y}|\mathbf{x})$ exponential to the number of labels. Second, obtaining a risk-minimizing prediction $\mathbf{h}^\star(\mathbf{x})$ from these samples is very costly[6], and approximate voting schemes must be employed in practice.

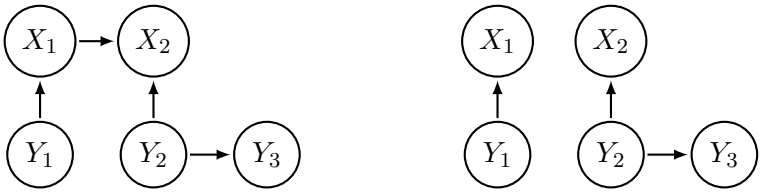
In [GG11], fully-connected CDN structures are employed, like the one in Figure 4.7, which impose no constraint on the expression of $p(\mathbf{y}|\mathbf{x})$. Standard binary logistic regression models are employed to learn the marginal conditional distributions $p(y_i|\mathbf{x}, \mathbf{y}_i)$, and a heuristic voting scheme is introduced to perform approximate MAP inference via Gibbs sampling.

4.3.3 Bayesian network classifiers

Introduced by Bielza et al. [BLL11], *multi-dimensional Bayesian network classifiers*
MBC (MBCs) are specific BNs models of the joint distribution $p(\mathbf{x}, \mathbf{y})$, in which the label nodes are not allowed to appear as children of feature nodes. Due to this constrained structure, a particular graph of interest is the so-called label-bridge subgraph, i.e. the original graph deprived of all the edges between the feature nodes. The label-bridge subgraph appears to be a sufficient structure to characterize the independence model of the labels conditioned on the feature set $\mathbf{X}$, and each of its maximal connected component characterizes a disjoint factor of the conditional joint distribution $p(\mathbf{y}|\mathbf{x})$. Given that such a factorization can be identified, the MAP inference problem can be solved much more efficiently by a decomposition of the learning and inference problems into simpler independent sub-problems. For example, considering the MBC structure in Figure 4.8 we have

$$\arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}) = \arg\left[\max_{y_1} p(y_1|x_1) \max_{y_2,y_3} p(y_2, y_3|x_2)\right].$$

[6] See Section 4.2.5 on ensemble learning and voting schemes complexity.



(a) MBC graph. **(b)** Label-bridge subgraph.

Fig. 4.8. A MBC graph and its label-bridge subgraph. It appears that $\{Y_1\} \perp\!\!\!\perp \{Y_2, Y_3\} \mid \mathbf{X}$, and therefore $p(\mathbf{y}|\mathbf{x})$ factorizes as $p(y_1|\mathbf{x})p(y_2, y_3|\mathbf{x})$. These two factors are identified by the maximal connected components of the label-bridge subgraph.



(a) General BN graph. **(b)** MBC-constrained graph.

Fig. 4.9. A BN graph where $Y_1 \perp\!\!\!\perp Y_2 \mid \mathbf{X}$, which can not be represented by a MBC graph.

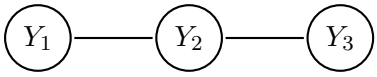
The property of label-bridge decomposability in MBCs relates closely to the work we have achieved during this thesis, as we will see in Chapter 5. However, label-bridge decomposition does not provide a general characterization of every possible disjoint factorization of $p(\mathbf{y}|\mathbf{x})$. This is shown in Figure 4.9, where the independence relation $Y_1 \perp\!\!\!\perp Y_2 \mid X_1$ encoded in the BN can not be encoded in a MBC without violating either $Y_1 \not\perp\!\!\!\perp X_1$ or $Y_2 \not\perp\!\!\!\perp X_1$. In Chapter 5 we will present several theoretical results to characterize the set of all irreducible disjoint label factors of $p(\mathbf{y}|\mathbf{x})$, without assuming the existence of any underlying probabilistic graphical structure.

In order to simplify both learning and inference, Antonucci et al. [Ant+13] consider a restricted class of MBC structures where the label subgraph (induced over $\mathbf{Y}$) is a tree and the feature subgraph (induced over $\mathbf{X}$) is empty. The first constraint imposes a low-treewidth in $p(\mathbf{y}|\mathbf{x})$, while the second constraint imposes that all features are conditionally independent given the labels, thus extending the naive Bayes assumption to the multi-dimensional output setting. Under these additional constraints, exact inference algorithms such as belief propagation can be used to solve efficiently the MAP inference problem.

4.3.4 Conditional Random Fields

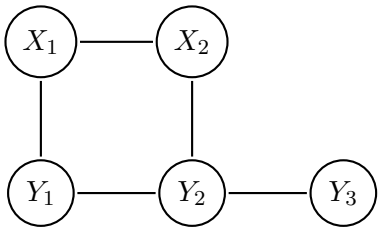
CRF (over $\mathbf{Y}$)

Introduced by Lafferty et al. [LMP01], *conditional random fields* (CRFs) form a family of undirected graphical models which represent a conditional joint distribution $p(\mathbf{y}|\mathbf{x})$. In the initial formulation, a CRF structure consists in an undirected graph $\mathcal{G}$



$$p(\mathbf{y}|\mathbf{x}) = \phi_1(y_1, y_2, \mathbf{x})\phi_2(y_2, y_3, \mathbf{x})$$

Fig. 4.10. A CRF over $\mathbf{Y}$, which encode the independence relation $\{Y_1\} \perp\!\!\!\perp \{Y_3\} \mid \{Y_2\} \cup \mathbf{X}$.



$$p(\mathbf{y}|\mathbf{x}) = \phi_1(x_1, x_2)\phi_2(x_1, y_1)\phi_3(x_2, y_2)\phi_4(y_1, y_2)\phi_5(y_2, y_3)$$

Fig. 4.11. A CRF over $\mathbf{X} \cup \mathbf{Y}$, which extends the one from Figure 4.10 with the independence relations $\{Y_1\} \perp\!\!\!\perp \{X_2, Y_3\} \mid \{X_1, Y_2\}$ and $\{Y_2, Y_3\} \perp\!\!\!\perp \{X_1\} \mid \{X_2, Y_1\}$ and $\{Y_3\} \perp\!\!\!\perp \mathbf{X} \cup \{Y_1\} \mid \{Y_2\}$.

defined over the node set $\mathbf{V} = \mathbf{Y}$, and the factorization of the conditional distribution $p(\mathbf{y}|\mathbf{x})$ is given by

$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{\mathbf{C}_i \in \mathcal{C}l_{\mathcal{G}}} \phi_i(\mathbf{c}_i, \mathbf{x}),$$

where $\mathcal{C}l_{\mathcal{G}}$ is the set of all cliques in $\mathcal{G}$. The semantics of such a CRF are then very similar to those of a Markov network, and just like Markov networks they must rely on the Hammersley-Clifford theorem for soundness, i.e. $p > 0$. The independence model induced by the graph is given by the u-separation criterion, and consists only in conditional independence relations in the form $\mathbf{A} \perp\!\!\!\perp \mathbf{B} \mid \mathbf{C} \cup \mathbf{X}$ with $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ disjoint subsets of $\mathbf{Y}$.

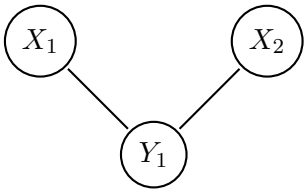
CRF (over $\mathbf{XY}$)

Most often, a second formulation of CRFs is found in the literature, with $\mathcal{G}$ an undirected graph defined over the node set $\mathbf{V} = \mathbf{X} \cup \mathbf{Y}$ and a factorization of $p(\mathbf{y}|\mathbf{x})$ given by

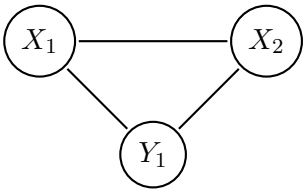
$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{\mathbf{C}_i \in \mathcal{C}l_{\mathcal{G}}} \phi_i(\mathbf{c}_i).$$

In such CRFs the graph also encodes interactions between $\mathbf{X}$ and $\mathbf{Y}$, which can reduce further the number of parameters required to model $p(\mathbf{y}|\mathbf{x})$.

However, under such an interpretation of CRFs, $\mathcal{G}$ does not necessarily provide a sound probabilistic independence model. An example is given in Figure 4.12, where the factorization encoded in first graph imposes a strong structural constraint on $p(\mathbf{y}|\mathbf{x})$, and yet does not induce any conditional independence relation. Clearly, here the (missing) adjacencies between features affect the factorization of $p(\mathbf{y}|\mathbf{x})$, without affecting the underlying independence model.

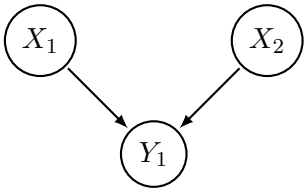


(a) $p(\mathbf{y}|\mathbf{x}) = \phi_1(x_1, y_1)\phi_2(x_2, y_1)$

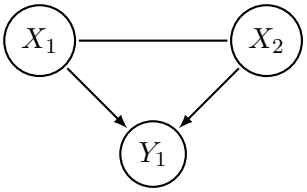


(b) $p(\mathbf{y}|\mathbf{x}) = \phi_1(x_1, x_2, y_1)$

Fig. 4.12. Two CRFs over $\mathbf{X} \cup \mathbf{Y}$ which encode $p(\mathbf{y}|\mathbf{x})$. The edges between features in $\mathbf{X}$ clearly constrain the joint distribution, without necessarily impacting the independence model.



(a) $p(\mathbf{y}|\mathbf{x}) = \phi_1(x_1, x_2, y_1)$

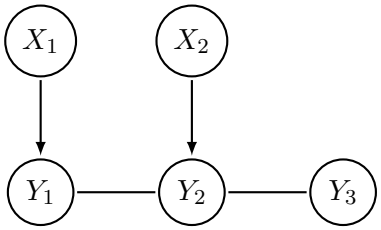


(b) $p(\mathbf{y}|\mathbf{x}) = \phi_1(x_1, x_2, y_1)$

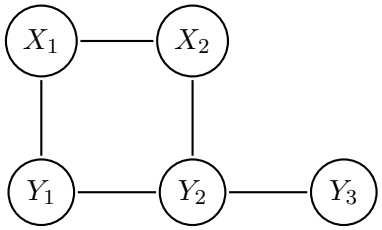
Fig. 4.13. Two LWF chain graphs which encode both $p(\mathbf{x})$ and $p(\mathbf{y}|\mathbf{x})$. The edges in the $\mathbf{X}$ components do not impact the factorization of $p(\mathbf{y}|\mathbf{x})$.

Interestingly, by making the edges between $\mathbf{X}$ and $\mathbf{Y}$ directed from the features to the labels, we obtain a sound probabilistic independence model under the LWF chain graph interpretation. As shown in Figure 4.13, in such a LWF chain graph the edges between the features impact only the factorization of $p(\mathbf{x})$ and have no effect on $p(\mathbf{y}|\mathbf{x})$.

By considering a restricted class of LWF chain graphs, with no edges between features and directed edges only from features to labels, we obtain a sound probabilistic independence model of $p(\mathbf{y}|\mathbf{x})$, with conditional independence relations in the form $\mathbf{A} \perp\!\!\!\perp \mathbf{B} \mid \mathbf{C} \cup \mathbf{X} \setminus (\mathbf{A} \cup \mathbf{B} \cup \mathbf{C})$ with $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ disjoint subsets of $\mathbf{X} \cup \mathbf{Y}$ and $(\mathbf{A} \cup \mathbf{B}) \cap \mathbf{Y} \neq \emptyset$.



(a) LWF chain graph over $\mathbf{X} \cup \mathbf{Y}$.



(b) Closure graph of $\mathbf{Y}$.

Fig. 4.14. An LWF chain graph along with the closure graph of the chain component $\mathbf{Y}$, which encodes a factorization of $p(\mathbf{y}|\mathbf{x})$ over its cliques.

While significant work has been done on structure learning for generative models, only a few papers address CRF structure learning [BG10]. In many MLC approaches based on CRFs the graph is tacitly assumed to be fixed, and MAP inference is performed, thereby minimizing the subset zero-one loss. Early applications of CRFs often assume loop-less structures such as chains or trees for practical reasons, to simplify both parameter learning and inference [LMP01]. For example, HMM-like CRFs define one potential for each label pair $\phi(y_i, y_{i+1})$, and one potential for each label-feature pair $\phi(y_i, x_i)$. Recent applications of CRFs have used more general graphical structures, however inference remains intractable in general, and approximate algorithms must be employed in practice. Interestingly, inference with CRFs can be formulated as a large-margin optimization problem that relates to Structural SVMs (S-SVMs) [Tso+05]. For a comprehensive review of CRFs, the reader is directed to Sutton and McCallum [SM12], and for S-SVMs to Joachims et al. [JFY09].

4.3.5 Sum product networks

SPN

Introduced by Poon and Domingos [PD11], *sum-product networks* (SPNs) are probabilistic graphical models of a joint distribution $p(\mathbf{v})$, though not in the classical sense. In classical PGMs, the graphical structure $\mathcal{G}$ is defined over a node set corresponding to the random variables in $\mathbf{V}$. In a SPN, the joint distribution $p(\mathbf{v})$ is constrained by a directed tree structure[7] with three types of nodes: sum, product and leaf. Each node N_i represents a local joint distribution $p_i(\mathbf{v}_i)$ over a subset of variables $\mathbf{V}_i \subseteq \mathbf{V}$ called its scope[8]. Note that each local distributions is underlined $p_i(\mathbf{v}_i)$ since it is only defined within the context of the current node N_i, and shall not be confused with the marginal distribution $p(\mathbf{v}_i)$. The joint distribution $p(\mathbf{v})$ encoded by the SPN corresponds to the local distribution of the root node, and decomposes recursively into products and weighted sums of local distributions according to the SPN structure.

Formally, the local distribution of a *product node* N_i is defined as a product of the local distributions of its children $\mathbf{CH}_{N_i}$,

$$p_i(\mathbf{v}_i) = \prod_{N_j \in \mathbf{CH}_{N_i}} p_j(\mathbf{v}_j), \tag{4.2}$$

[7] Note that SPNs can also be represented as DAGs, which are essentially compressed SPN trees [RG16].
[8] Note that we consider normal SPNs without loss of generality [ZMP15][Theorem 3], [Peh+15][Theorem 3].

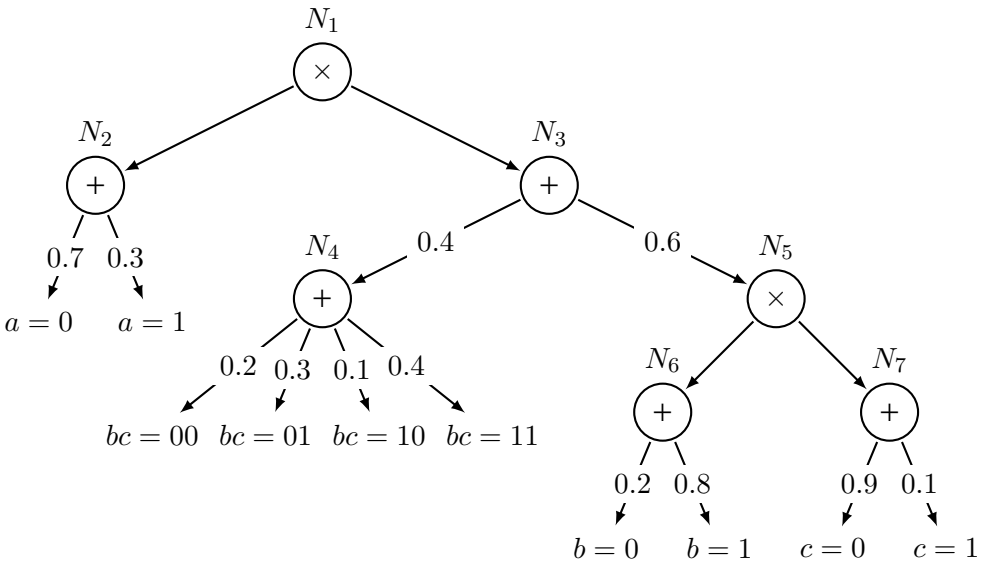


Fig. 4.15. A SPN model of a joint distribution $p(a, b, c)$ of three binary random variables.

where the scopes of the child nodes $\{\mathbf{V}_j \mid N_j \in \mathbf{CH}_{N_i}\}$ form a partition of the scope of the product node $\mathbf{V}_i$. On the other hand, the local distribution of a *sum node* N_i is defined as a weighted sum of the local distributions of its children $\mathbf{CH}_{N_i}$,

$$p_i(\mathbf{v}_i) = \sum_{N_j \in \mathbf{CH}_{N_i}} \theta_{i,j} p_j(\mathbf{v}_j), \tag{4.3}$$

where the weights are normalized positive values, $\sum_j \theta_{i,j} = 1$, and every child node shares the same scope as the sum node, $\mathbf{V}_j = \mathbf{V}_i$. Finally, a *leaf node* N_i represent a probability distribution $p_i(\mathbf{v}_i)$ by any mean, e.g. a probability table for discrete variables or a probability density function for continuous variables. An interesting SPN setting is when the leaf nodes represent deterministic distributions, i.e. $p_i(\mathbf{v}_i)$ equal to 1 for a particular value $\mathbf{v}_i$, and 0 elsewhere[9]. In this setting, a SPN encodes the full joint distribution $p(\mathbf{v})$ solely with the parameters Θ corresponding to the weights of the sum nodes. Such a SPN is presented in Figure 4.15.

Both structure and parameter learning of SPN models remain hard problems [Zha+16]. However, obtaining any joint, marginal or conditional probability in the form $p(\mathbf{y}|\mathbf{x})$ requires only a computational cost linear to the size of the SPN (i.e., the number of nodes). For example, consider the query $p(a = 1, b = 1|c = 0)$, in the SPN from Figure 4.15. To compute $p(a = 1, b = 1, c = 0)$, it suffices to set a value 1 to every leaf node which respect the evidence, like $a = 1$ or $bc = 10$, and a value 0 to every leaf node which violates it, like $a = 0$ or $bc = 11$. Then, by unrolling all the computations encoded in the SPN with a bottom-up pass up to the root node, we obtain

$$p(a = 1, b = 1, c = 0) = 0.3 \times (0.4 \times 0.1 + 0.6 \times 0.8 \times 0.9) = 0.1416.$$

[9] A Dirac distribution in the case of continuous variables.

Similarly, we may compute $p(c = 0)$ with one bottom-up pass

$$p(c = 0) = (0.7 + 0.3) \times (0.4 \times (0.2 + 0.1) + 0.6 \times (0.2 + 0.8) \times 0.9) = 0.66.$$

Finally, after only two passes through the SPN we get

$$p(a = 1, b = 1|c = 0) = \frac{0.1416}{0.66} = 0.2145455.$$

This easy marginalization procedure is an interesting property of SPNs, however MAP inference remains a hard problem in general. In their seminal paper, Poon and Domingos [PD11] proposed an exact MAP inference procedure with complexity linear to the size of the SPN, however this procedure was later shown to be wrong [Peh15].

The latent variable interpretation

A convenient way of interpreting a SPN structure is by re-expressing each local distribution $p_i(\mathbf{v}_i)$ with respect to the global distribution $p(\mathbf{v})$. Let us first consider a product node N_i. Because the scopes of the children form a partition of $\mathbf{V}_i$, their local distributions can be interpreted as marginals of the parent distribution and the expression (4.2) becomes

$$p_i(\mathbf{v}_i) = \prod_{N_j \in \mathbf{CH}_{N_i}} p_i(\mathbf{v}_j).$$

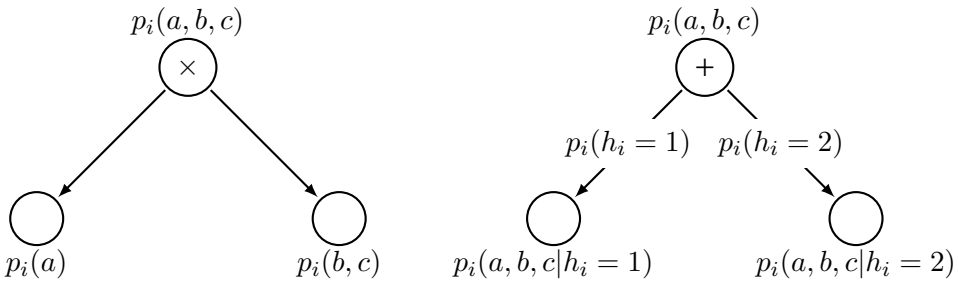
contextual ind.

Therefore, each product node encodes a local independence model of p_i with relations in the form $\mathbf{V}_j \perp\!\!\!\perp \mathbf{V}_i \setminus \mathbf{V}_j$ for every children $N_j \in \mathbf{CH}_{N_i}$. Such relations are sometimes called *contextual independence relations*, and do not necessarily hold in p, unless the product node is the root of the SPN and $p_i = p$. Second, let us consider a sum node N_i. Because the weights of each sum node are normalized, these may be interpreted as the probability distribution of a hidden variable H_i taking values in $\{j \mid N_j \in \mathbf{CH}_{N_i}\}$, such that $\theta_{i,j} = p_i(h_i = j)$. The local distribution of each child node $p_j(\mathbf{v}_j)$ can be then seen as the conditional distribution $p_i(\mathbf{v}_i|h_i = j)$, and the expression (4.3) becomes

$$p_i(\mathbf{v}_i) = \sum_{N_j \in \mathbf{CH}_{N_i}} p_i(h_i) p_i(\mathbf{v}_i|h_i).$$

mixture model

Therefore, each sum node corresponds to a *mixture model* $p_i(\mathbf{v}_i) = \sum_{h_i} p_i(\mathbf{v}_i, h_i)$ with a hidden variable H_i. Finally, the local distribution of each node in the SPN can be expressed as a component of the local distribution of its parent node, up to the root node of the SPN which encode the whole distribution $p(\mathbf{v}) = \sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h})$, with $\mathbf{H}$ the set of all the hidden variables of the SPN. The local probability distribution



(a) A SPN product node N_i. **(b)** A SPN sum node N_i.

Fig. 4.16. Two SPN nodes, a product and a sum node, which respectively decompose a multivariate probability distribution $p_i(a,b,c)$ into a product over disjoint factors, $p_i(a,b,c) = p_i(a)p_i(b,c)$, or a mixture model with a hidden variable, $p_i(a,b,c) = \sum_{h_i} p_i(a,b,c,h_i)$.

encoded by each intermediate node N_i can now be expressed with respect to the global distribution $p(\mathbf{v},\mathbf{h})$

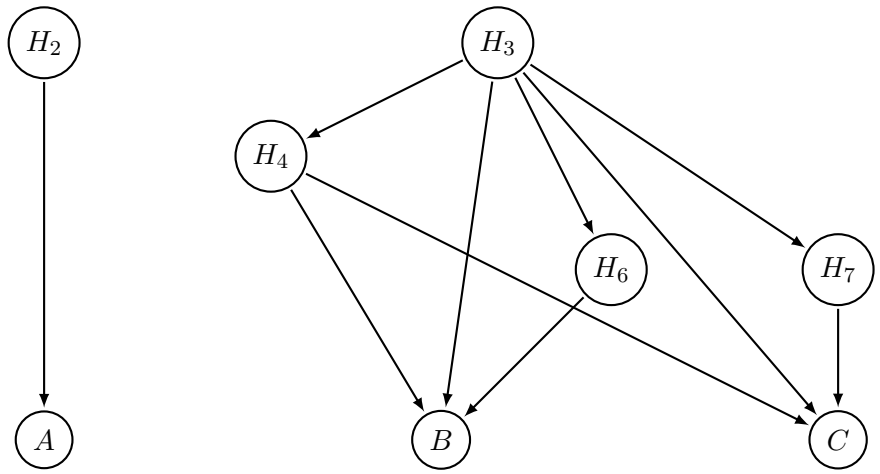
$$p_i(\mathbf{v}_i) = p(\mathbf{v}_i|\mathbf{h}^{(i)}_{\mathbf{AN}_i}),$$

where $\mathbf{H}_{\mathbf{AN}_i}$ is the set of all hidden variables introduced by the ancestors of N_i in the tree, and $\mathbf{h}^{(i)}_{\mathbf{AN}_i}$ is the particular value these hidden variables take in the context of node N_i. Likewise, the local distribution of each hidden variable H_i is expressed as

$$p_i(h_i) = p(h_i|\mathbf{h}^{(i)}_{\mathbf{AN}_i}).$$

In the end, we may represent the joint distribution $p(\mathbf{v},\mathbf{h})$ of both the observed and hidden variables as a Bayesian network with partial probability tables for the hidden variables in $\mathbf{H}$, which represent the local mixture models encoded by the sum nodes of the SPN, and compact probability tables for the observed variables in $\mathbf{V}$, which represent the local independence relations encoded by the product nodes of the SPN. Such a BN representation is given in Figure 4.17.

Moreover, since we are interested in modeling the marginal distribution $\sum_{\mathbf{h}} p(\mathbf{v},\mathbf{h})$, Zhao et al. [ZMP15] notice that we may fill up the missing values of the conditional probability tables of the BN by repeating the same probability distribution, so that $p(h_i|\mathbf{h}^{(i)}_{\mathbf{AN}_i}) = p(h_i)$. As a result, all the hidden variables are made (unconditionally) independent of each other, which simplifies the BN structure to a bipartite graph. The resulting BN is given in Figure 4.18, with conditional probability tables represented as Algebraic Decision Diagrams (ADDs). In such a representation, both the BN structure and the probability distribution $p(\mathbf{v}|\mathbf{h})$ with ADDs are specified by the SPN structure, while the probability distribution of the hidden variables $p(\mathbf{h})$ is specified by the SPN weights Θ. Notice that due to the BN structure $p(\mathbf{h})$ factorizes into $\prod_{h_i} p(h_i)$, while $p(\mathbf{v}|\mathbf{h})$ factorizes into $\prod_{v_i} p(v_i|\mathbf{h})$.



(a) The BN structure of the SPN.

H_2	
1	2
0.7	0.3

(b) $p(h_2)$

H_3	
1	2
0.4	0.6

(c) $p(h_3)$

H_3	H_4 = 1	2	3	4
1	0.2	0.3	0.1	0.4
2	na	na	na	na

(d) $p(h_4|h_3)$

H_3	H_6 = 1	2
1	na	na
2	0.2	0.8

(e) $p(h_6|h_3)$

H_3	H_7 = 1	2
1	na	na
2	0.9	0.1

(f) $p(h_7|h_3)$

H_2	A = 0	1
1	1	0
2	0	1

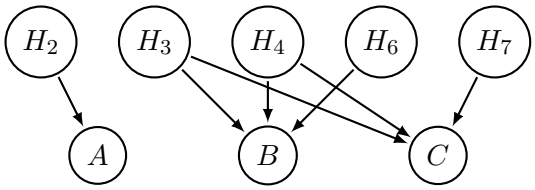
(g) $p(a|h_2)$

H_3	H_4	H_6	B = 0	1
1	1	{1,2}	1	0
	2		1	0
	3		0	1
	4		0	1
2	{1,2,3,4}	1	1	0
		2	0	1

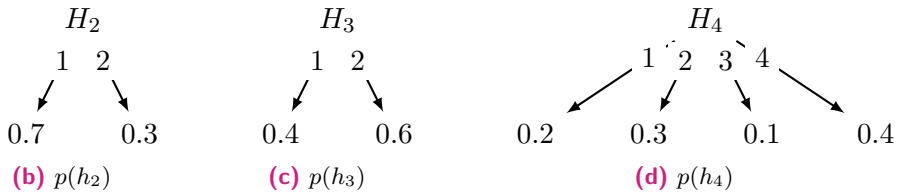
H_3	H_4	H_7	C = 0	1
1	1	{1,2}	1	0
	2		0	1
	3		1	0
	4		0	1
2	{1,2,3,4}	1	1	0
		2	0	1

(h) $p(b|h_3, h_4, h_6)$ **(i)** $p(c|h_3, h_4, h_7)$

Fig. 4.17. The SPN from Figure 4.15 represented as a BN over $\mathbf{V} \cup \mathbf{H}$, with $\mathbf{H}$ a set of hidden variables corresponding to the sum nodes of the SPN. The joint distribution $p(\mathbf{v}, \mathbf{h})$ is not completely specified due to the missing values (na) in the probability tables of $p(\mathbf{h})$. However the marginal distribution $p(\mathbf{v}) = \sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h})$ is fully specified, due to the compact representation of the probability tables for $p(\mathbf{v}|\mathbf{h})$.



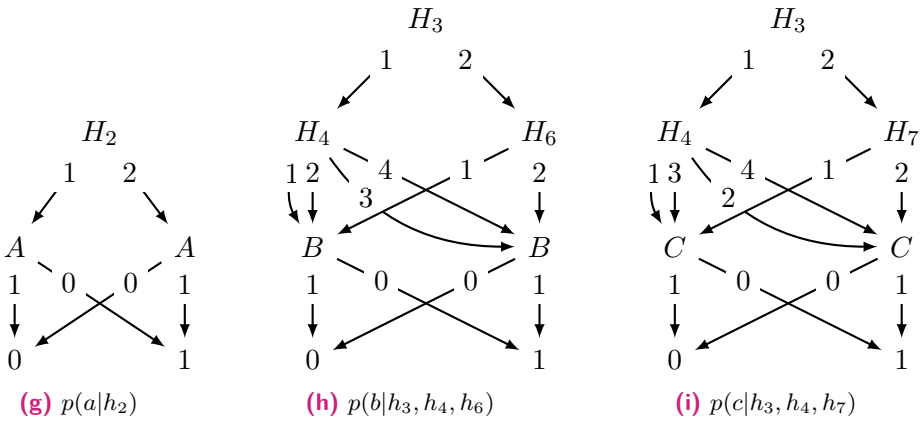
(a) The bipartite BN structure of the SPN.



(b) $p(h_2)$ **(c)** $p(h_3)$ **(d)** $p(h_4)$



(e) $p(h_6)$ **(f)** $p(h_7)$



(g) $p(a|h_2)$ **(h)** $p(b|h_3, h_4, h_6)$ **(i)** $p(c|h_3, h_4, h_7)$

Fig. 4.18. The BN from Figure 4.15 completed so that $p(\mathbf{v}, \mathbf{h})$ is completely specified, and represents the same marginal distribution $\sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h})$. The probability tables are now represented as Algebraic Decision Diagrams (ADDs), which encode the local independence relations of the product nodes in the SPN.

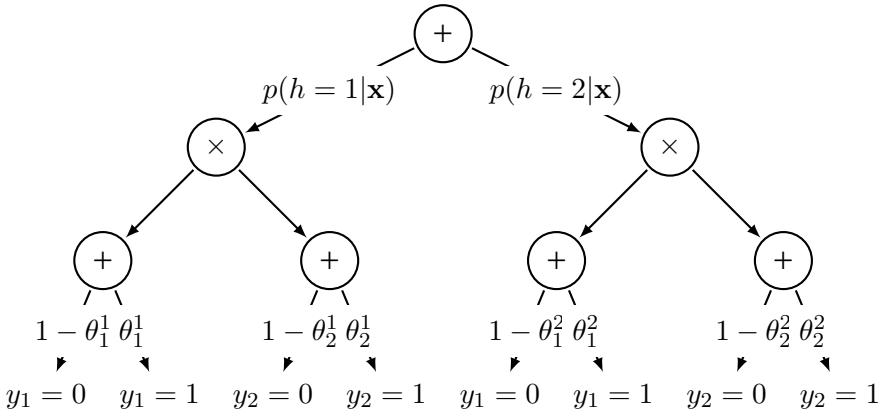


Fig. 4.19. A SPN model of a conditional joint distribution $p(y_1, y_2|\mathbf{x})$ with two binary labels, equivalent to a mixture of conditional Bernoulli distributions as in [Li+16]. Here we have only two components ($k = 2$), and θ_i^j denotes $p(y_i = 1|\mathbf{x}, h = j)$.

SPNs for multi-label classification

The closest application of SPNs to the MLC problem can be found in [Li+16], where the joint conditional distribution $p(\mathbf{y}|\mathbf{x})$ is represented as a mixture of k conditional Bernoulli distributions. In such a model a single hidden variable H is introduced, which takes values in $\{1, \ldots, k\}$, and within each mixture component the conditional distribution of labels is assumed to factorize over the whole label set. Therefore, the conditional joint distribution is expressed as

$$p(\mathbf{y}|\mathbf{x}) = \sum_{j=1}^{k} p(h = j|\mathbf{x}) \prod_{i=1}^{n} p(y_i|\mathbf{x}, h = j).$$

Such a model can be expressed as a three-layer SPN as the one in Figure 4.19, in which the weights of the sum nodes are not fixed but inferred from a set of probabilistic models. More precisely, the $p(h = j|\mathbf{x})$ parameters can be obtained from a multinomial probabilistic regression, and the $p(y_i|\mathbf{x}, h = j)$ parameters from a series of binary probabilistic regressions. Due to the constrained structure of this SPN, both the learning with a standard expectation-maximization (EM) procedure and exact MAP inference can be performed, thereby modeling relatively complex probability distributions at a reasonable cost.

4.3.6 Discussion

While plug-in approaches provide a principled way to deal with the multi-label classification problem, in practice each approach has to deal with the exponential blow-up of the output space, either during the learning phase to model $p(\mathbf{y}|\mathbf{x})$, or during the inference phase to obtain $\mathbf{h}^\star = \arg\min_{\mathbf{h}} \sum_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}) L(\mathbf{h}, \mathbf{y})$. Still,

graphical models seem to be a particularly well suited tool in this context, as they are able to express structural constraints which reduce both the parameter complexity and the computational complexity of the problem. In the next chapter we will introduce a particular structural constraint of $p(\mathbf{y}|\mathbf{x})$, that is, its decomposition into a product of marginal probability distributions called *irreducible disjoint label factors*.

5 Irreducible label factors

> *Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius — and a lot of courage to move in the opposite direction.*
>
> **— Ernst F. Schumacher**
> 1973

In this chapter we propose a generic approach to identify the unique partition of the label set into irreducible label factors (ILFs), that is, the irreducible factorization $p(\mathbf{y}|\mathbf{x})$ into disjoint marginal distributions,

$$p(\mathbf{y}|\mathbf{x}) = \prod_{\mathbf{Y}_F \in \mathcal{F}} p(\mathbf{y}_F|\mathbf{x}),$$

where $\mathcal{F}$ is a partition of the label set, and each $\mathbf{Y}_F$ is called a label factor. Our approach draws strongly on the constraint-based structure learning methods discussed in Chapter 3, and constitutes the major contribution of this Thesis.

In Section 5.1 we introduce formally the concept of irreducible label factors, and present a series of theoretical results to address the ILF decomposition problem. We show that a generic procedure exists in the general case with only $O(m^2)$ pairwise conditional independence tests between the labels, then we consider reasonable assumptions about the underlying probability distribution in order to derive three practical procedures: 1) ILF-DAG when p is faithful to a DAG; 2) ILF-Inter when p supports the Intersection property; and 3) ILF-Compo p supports the Composition property. As a subsidiary result, we show that each of these procedures is also able to tackle the feature subset selection problem in a principled way.

In Section 5.2 we apply the ILF approach to multi-label classification (MLC) problem for subset $0/1$ loss minimization, with a simple decomposition of the LP scheme. In Section 5.3 we do the same for F-loss minimization, a.k.a. F-measure maximization, by decomposing the Bayes-Optimal GFM method. Our conclusions are supported by carefully designed experiments on synthetic and benchmark data [GAE15; GA16a].

5.1 Characterizations

In this section, we first introduce formally the concept of *irreducible label factors*, and then address the problem of identifying the ILF decomposition a joint conditional distribution. Since this problem closely relates to the Markov boundary discovery problem, in a second part we also discuss the feature subset selection problem in the context of MLC. The resulting procedures for discovering both the ILFs and their (minimum) feature subsets, termed ILF-DAG, ILF-Inter and ILF-Compo, will then be subject to experimental validation.

We shall assume throughout that $\mathbf{X}$ is the feature set, $\mathbf{Y}$ is the label set, $\mathbf{U} = \mathbf{X} \cup \mathbf{Y}$ is the union of both and p is the underlying joint distribution. The proofs of the Theorems and Lemmas presented here are deferred to the Appendix.

5.1.1 Preliminary materials

We shall now introduce formally the concept of *label factor* that will play a pivotal role in the factorization of $p(\mathbf{y}|\mathbf{x})$.

Def. 5.1 *A label factor is a subset $\mathbf{Y}_F \subseteq \mathbf{Y}$ such that $\mathbf{Y}_F \perp\!\!\!\perp \mathbf{Y} \setminus \mathbf{Y}_F \mid \mathbf{X}$. Additionally, an irreducible label factor is non-empty and has no non-empty label factor as proper subset.*

The key idea is then to decompose the joint conditional distribution of the labels into a product of disjoint marginal conditional distributions,

$$p(\mathbf{y}|\mathbf{x}) = \prod_{\mathbf{Y}_F \in \mathcal{F}_I} p(\mathbf{y}_F|\mathbf{x}).$$

Algebraic structure

Label factors can be characterized as an algebraic structure satisfying certain axioms. Let $\mathcal{F}$ denote the set of all label factors (LFs for short), and $\mathcal{F}_I \subset \mathcal{F}$ the set of all irreducible label factors (ILFs for short). It is easily shown that $\{\mathbf{Y}, \emptyset\} \subseteq \mathcal{F}$. More specifically, $\mathcal{F}$ can be ordered via subset inclusion to obtain a lattice bounded by $\mathbf{Y}$ itself and the null set, while $\mathcal{F}_I$ forms a partition of $\mathbf{Y}$.

Thm. 5.1 (LF algebraic structure) *If $\mathbf{Y}_{F_i}, \mathbf{Y}_{F_j} \in \mathcal{F}$, then $\mathbf{Y}_{F_i} \cup \mathbf{Y}_{F_j} \in \mathcal{F}$, $\mathbf{Y}_{F_i} \cap \mathbf{Y}_{F_j} \in \mathcal{F}$ and $\mathbf{Y}_{F_i} \setminus \mathbf{Y}_{F_j} \in \mathcal{F}$. Moreover, $\mathbf{Y}$ breaks down into a unique partition of irreducible components, $\mathcal{F}_I$.*

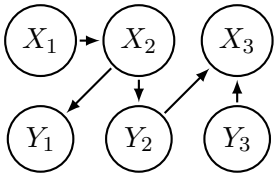


Fig. 5.1. An example DAG.

It follows that the factorization of $p(\mathbf{y}|\mathbf{x})$ into minimal disjoint marginal distributions is unique.

Relation to PGMs

To illustrate the concept of ILF decomposition, consider the following example.

Ex. 5.1 *Suppose p is faithful to the DAG from Figure 5.1. From the d-separation criterion we have that $\{Y_1\} \perp\!\!\!\perp \{Y_2, Y_3\} \mid \mathbf{X}$, so both $\{Y_1\}$ and $\{Y_2, Y_3\}$ are label factors. However, we have $\{Y_2\} \not\perp\!\!\!\perp \{Y_1, Y_3\} \mid \mathbf{X}$ and $\{Y_3\} \not\perp\!\!\!\perp \{Y_1, Y_2\} \mid \mathbf{X}$, so $\{Y_2\}$ and $\{Y_3\}$ are not label factors. Therefore $\{Y_1\}$ and $\{Y_2, Y_3\}$ are the irreducible label factors, and $p(\mathbf{y}|\mathbf{x}) = p(y_1|\mathbf{x}) \times p(y_2, y_3|\mathbf{x})$.*

Note that the concept of irreducible label factors bears a close resemblance to the maximal connected components in so-called "class-bridge decomposable" multi-dimensional Bayesian network classifiers (MBCs) [BLL11], discussed in Section 4.3.3. Still, MBCs are not able to provide a general characterization of every possible ILF decomposition, as was illustrated in Figure 4.9.

In [GAE14] we follow a similar approach, where we learned a generic Bayesian network structure from data in order to identify the ILFs. In the present work we present an even more generic approach, and show that the ILFs can be characterized efficiently without assuming the existence of any underlying probabilistic graphical model.

Feature subset selection

The concept of Markov blanket offers a principled solution to the *feature subset selection* (FSS) problem [KS96], which can be formulated in terms of conditional independence. A feature subset of $\mathbf{Y}$ is by definition a subset $\mathbf{M} \subseteq \mathbf{X}$ such that $\mathbf{Y} \perp\!\!\!\perp \mathbf{X} \setminus \mathbf{M} \mid \mathbf{M}$, a.k.a. a Markov blanket[1] of $\mathbf{Y}$ in $\mathbf{X}$. Similarly, a *minimal feature*

[1] The formal definition of a Markov blanket and a Markov boundary can be found in Section 3.3.3, Definition 3.1.

minimal feature subset

subset is a Markov boundary of $\mathbf{Y}$ in $\mathbf{X}$. In the context of MLC, the problem of identifying a minimal feature subset for a set of labels boils down to a Markov boundary discovery problem. In the following, we will provide necessary and sufficient conditions to characterize the irreducible label factors, $\{\mathbf{Y}_F\}$, and their respective Markov boundaries (or at least Markov blankets) in $\mathbf{X}$, $\{\mathbf{M}_F\}$,

$$p(\mathbf{y}|\mathbf{x}) = \prod_{\mathbf{Y}_F \in \mathcal{F}_I} p(\mathbf{y}_F|\mathbf{m}_F).$$

An important theoretical result states that the minimal feature subset is unique in distributions satisfying the Intersection property [Pea89].

Thm. 5.2 *Consider* $\mathbf{V}$, $\mathbf{W}$ *two subsets of* $\mathbf{U}$. *Then,* $\mathbf{V}$ *has a unique Markov boundary in* $\mathbf{W}$ *if* p *supports the Intersection property.*

Still, Theorem 5.2 says nothing about distributions that do not satisfy the Intersection property, therefore it might very well be the case that the labels we are seeking to predict possess multiple minimal feature subsets [SLA13; Peñ+07].

5.1.2 Irreducible label factors

The problem of identifying the ILFs is non-trivial, as we shall see, so we may consider three assumptions about the underlying distribution p, namely the DAG-faithfulness, the Intersection, and the Composition assumptions. We first show that the ILF decomposition may be read off from a DAG in $O(m)$ operations, under the assumption that p is faithful to the graph. Then we derive two convenient results to characterize the ILFs under the Intersection and Composition assumptions with $O(m^2)$ independence tests. Finally, we show that a on-trivial characterization in $O(m^2)$ exists also for general distributions, which does not necessarily translate into a practical procedure. In order to illustrate each of these characterizations, we introduce four DAGs in Figure 5.2 that will be used as example CI models all along our analysis.

All our results to characterize the ILFs will involve pairwise conditional independence relations between the labels in the form $Y_i \perp\!\!\!\perp Y_j \mid \mathbf{Z}$. Such independence relations may be directly read off from an independence model such as a DAG, or estimated from the data set $\mathcal{D}$ with a statistical CI test. As discussed in Section 3.3, the required sample size depends implicitly upon the degree of freedom of the test, which increases exponentially with the number of variables considered. Therefore, it is of practical interest to keep the conditioning set $\mathbf{Z}$ as small as possible in order for our theoretical results to translate into feasible solutions.

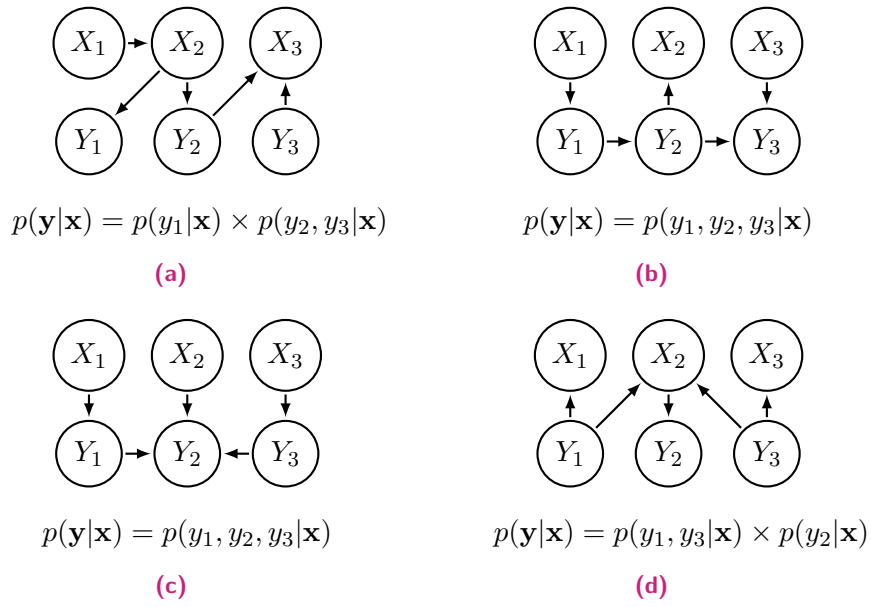


Fig. 5.2. Four DAG examples along with the ILF decomposition of $p(\mathbf{y}|\mathbf{x})$, given that p is faithful to $\mathcal{G}$.

Under the DAG-Faithfulness assumption

Let us first provide a characterization of the ILFs with $O(2^m)$ pairwise independence relations, which is intractable in general but will prove useful under the DAG-faithfulness assumption.

Thm. 5.3 *Let $\mathcal{G}$ be an undirected graph whose nodes correspond to the random variables in $\mathbf{Y}$ and in which two nodes Y_i and Y_j are adjacent* iff *there exists $\mathbf{Z} \subseteq \mathbf{Y} \setminus \{Y_i, Y_j\}$ such that $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid (\mathbf{X} \cup \mathbf{Z})$. Then, each connected component in $\mathcal{G}$ is an ILF.*

Theorem 5.3 offers an elegant graphical approach to characterize the ILFs by mere inspection of the connected components in a graph $\mathcal{G}$, which can be done efficiently in $O(m)$ using a breadth-first search algorithm. For illustration purposes, let us show how such a graph is constructed from the examples in Figure 5.2.

Ex. 5.2 *Consider the four DAGs in Figure 5.2. In DAG (a), from the d-separation criterion we have that $\{Y_1\} \perp\!\!\!\perp \{Y_2\} \mid \mathbf{X}$ and $\{Y_1\} \perp\!\!\!\perp \{Y_2\} \mid \mathbf{X} \cup \{Y_3\}$, so there is no edge between Y_1 and Y_2 in $\mathcal{G}$. Likewise, Y_1 and Y_3 are d-separated for every $\mathbf{Z} \in \{\emptyset, \{Y_2\}\}$, so these two are not adjacent either. Only the edge $Y_2 - Y_3$ is present, because we have $\{Y_2\} \not\perp\!\!\!\perp \{Y_3\} \mid \mathbf{X}$. By proceeding in the same way, in DAG (b) we have $\{Y_1\} \not\perp\!\!\!\perp \{Y_2\} \mid \mathbf{X}$, $\{Y_1\} \not\perp\!\!\!\perp \{Y_3\} \mid \mathbf{X}$ and $\{Y_2\} \not\perp\!\!\!\perp \{Y_3\} \mid \mathbf{X}$, so every pair of labels is adjacent. In DAG (c), we have $\{Y_1\} \not\perp\!\!\!\perp \{Y_2\} \mid \mathbf{X}$, $\{Y_1\} \not\perp\!\!\!\perp \{Y_3\} \mid \mathbf{X} \cup \{Y_2\}$ and $\{Y_2\} \not\perp\!\!\!\perp \{Y_3\} \mid \mathbf{X}$ so the graph is also complete. In DAG (d) we have $\{Y_1\} \perp\!\!\!\perp \{Y_3\} \mid \mathbf{X}$ due to the collider node X_2, so there is an edge between Y_1 and Y_3. There is no other edge in $\mathcal{G}$ because Y_1 and Y_2 are d-separated for every $\mathbf{Z} \in \{\emptyset, \{Y_3\}\}$, as well as Y_2 and Y_3 for every $\mathbf{Z} \in \{\emptyset, \{Y_2\}\}$. The*

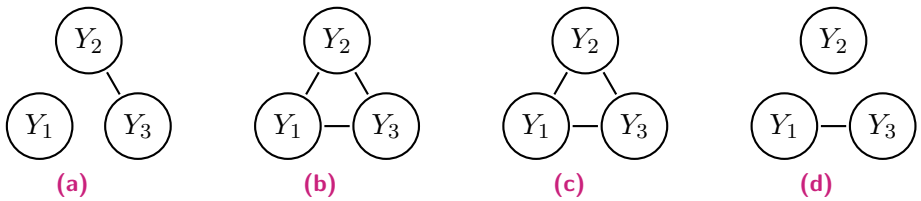


Fig. 5.3. Theorem 5.3 applied to our example DAGs.

graphs resulting from this procedure are displayed in Figure 5.3. We can now read off the ILFs directly from the connected components in these graphs.

Despite the simplicity of this graphical characterization, deciding upon whether $\exists \mathbf{Z} \subseteq \mathbf{Y} \setminus \{Y_i, Y_j\}$ such that $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid (\mathbf{X} \cup \mathbf{Z})$ is a challenging combinatorial problem as the number of possible combinations for $\mathbf{Z}$ grows exponentially with the number of labels, resulting in $O(2^m)$ conditional independence tests. However, when p is faithful to a DAG, the ILFs can be directly read off from the DAG in $O(m)$,

Thm. 5.4 *Suppose p is faithful to a DAG $\mathcal{G}$. Then, two labels Y_i and Y_j belong to the same ILF* iff *there exists a path in $\mathcal{G}$ between them such that all intermediate nodes are either (i) a label, or (ii) a collider.*

Theorem 5.4 directly follows from Theorem 5.3, and identifies all ILFs with a single breadth-first search through the DAG. Note that a similar result can be given for UGs (Markov networks), with the mere condition that all intermediate nodes must be labels. Still, applying such a procedure requires the DAG-faithfulness assumption about p, and implies to recover a DAG first, which is known to be a hard problem [CHM04].

Under the Intersection assumption

When p supports the Intersection property, the ILFs can identified with only $O(m^2)$ statistical tests of independence in the form $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid \mathbf{X} \cup \mathbf{Z}$, with $\mathbf{Z}$ ranging from $\emptyset$ to $\mathbf{Y} \setminus \{Y_i, Y_j\}$,

Thm. 5.5 *Consider $<$ a strict total order of $\mathbf{Y}$. Let $\mathcal{G}$ be an undirected graph whose nodes correspond to the random variables in $\mathbf{Y}$ and in which two nodes Y_i and Y_j ($Y_i < Y_j$) are adjacent* iff $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid \mathbf{X} \cup \{Y | Y > Y_i\} \setminus \{Y_j\}$. *Then, each connected component in $\mathcal{G}$ is an ILF if p supports the Intersection property.*

The quadratic complexity of Theorem 5.5 is very convenient, but the problem of performing a statistical test with a large conditioning set remains. A practical solution is given in Theorem 5.6, where the neighbourhood of each label Y_i in $\mathcal{G}$

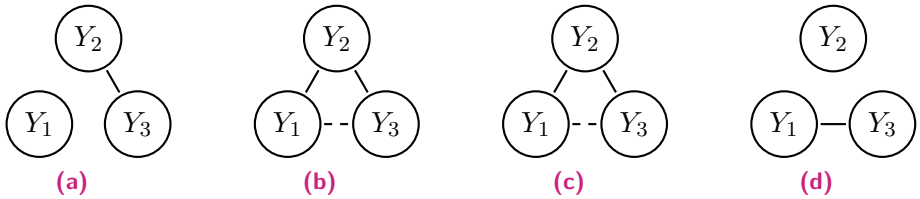


Fig. 5.4. Theorem 5.5 applied to our example DAGs (Intersection assumption). Dashes indicate edges that may be present or absent depending on the label ordering used.

boils down to a mere inspection of $\mathbf{M}_i$, a Markov boundary of Y_i in $\mathbf{X} \cup \{Y | Y > Y_i\}$. Thus, this new procedure does not explicitly rely on the result of a statistical test, but on our ability to infer Markov boundaries, for which there exists plenty of practical exact and approximate algorithmic solutions. Also, Theorem 5.6 exhibits the appealing property of identifying correct label factors in every situation (although not necessarily minimal), even when p does not obey the Intersection property.

Thm. 5.6 *Consider $<$ a strict total order of $\mathbf{Y}$, and let $\mathbf{M}_i$ denote an arbitrary Markov boundary of Y_i in $\mathbf{X} \cup \{Y | Y > Y_i\}$. Let $\mathcal{G}$ be an undirected graph whose nodes correspond to the random variables in $\mathbf{Y}$ and in which two nodes Y_i and Y_j ($Y_i < Y_j$) are adjacent* iff *Y_j belongs to $\mathbf{M}_i$. Then, each connected component in $\mathcal{G}$ is a LF, and an ILF if p supports the Intersection property.*

Note that under the Intersection assumption very similar procedures exists, with independence tests in the form $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid \mathbf{X} \cup \mathbf{Y} \setminus \{Y_i, Y_j\}$ in Theorem 5.5 and Markov boundaries in $\mathbf{X} \cup \mathbf{Y} \setminus \{Y_i\}$ in Theorem 5.6. Despite being conceptually simpler (no label ordering involved), such procedures have no theoretical advantage over those presented above, with larger conditioning sets compared to Theorem 5.5 and without the desirable correctness property of Theorem 5.6 when p does not support Intersection.

The Intersection assumption might be too restrictive in many practical scenarios. In fact, many real-life distributions (e.g., engineering systems such as digital circuits and engines that contain deterministic components) violate the Intersection property. As noted in [SLA13], high-throughput molecular data, known as the "multiplicity" of molecular signatures (i.e., different gene/biomarker sets perform equally well in terms of predictive accuracy of phenotypes) also suggests existence of multiple Markov boundaries. It is usually unknown to what degree the Intersection assumption holds in distributions encountered in practice. The following examples illustrate two cases where the Intersection property does not hold, and the characterizations provided in Theorems 5.5 and 5.6 do not necessarily yield an ILF decomposition.

Ex. 5.3 *Consider $\mathbf{Y} = \{Y_1, Y_2, Y_3\}$ three random variables such that $Y_1 = Y_2 = Y_3$, and $\mathbf{X} = \emptyset$. Clearly the Intersection property does not hold, and every pair of labels is conditionally independent given the third one. If we build the graph in Theorem 5.5 with the natural*

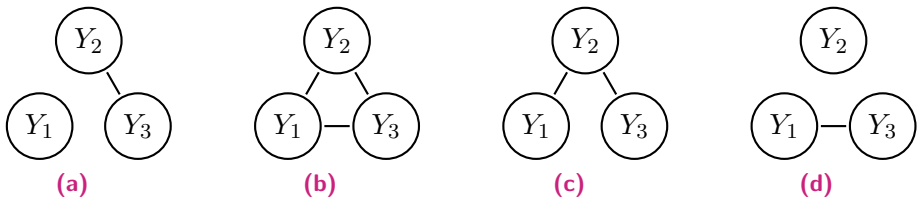


Fig. 5.5. Theorem 5.7 applied to our example DAGs (Composition assumption).

ordering (Y_1, Y_2, Y_3) *we end up with two label factors* $\{Y_1\}$ *and* $\{Y_2, Y_3\}$*, which is clearly wrong.*

Ex. 5.4 *Consider* $\mathbf{Y} = \{Y_1, Y_2\}$ *and* $\mathbf{X} = \{X_1\}$*, with* Y_1, Y_2, X_1 *three random variables such that* $Y_1 = Y_2 = X_1$*. Clearly the Intersection property does not hold. If we apply Theorem 5.6 with the ordering* $\{Y_1, Y_2\}$*, then we have* $\mathbf{M}_1 \in \{\{X\}, \{Y_2\}\}$ *and* $\mathbf{M}_2 = \{X\}$*, and the procedure either ends up with the irreducible label factors* $\{Y_1\}$ *and* $\{Y_2\}$*, or the label factor* $\{Y_1, Y_2\}$ *depending on which Markov boundary is chosen for* Y_1*.*

Under the Composition assumption

When p supports the Intersection property, the ILFs can identified with only $O(m^2)$ statistical tests of independence in the form $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid \mathbf{X}$,

Thm. 5.7 *Let* $\mathcal{G}$ *be an undirected graph whose nodes correspond to the random variables in* $\mathbf{Y}$ *and in which two nodes* Y_i *and* Y_j *are adjacent* iff $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid \mathbf{X}$*. Then, each connected component in* $\mathcal{G}$ *is an ILF if* p *supports the Composition property.*

From Theorem 5.7 we obtain a quadratic procedure which reduces the conditioning set of the statistical test to only the feature set $\mathbf{X}$ compared to $\mathbf{X} \cup \mathbf{Z}$ in Theorem 5.5. Still, performing that statistical test remains problematic for high dimensional input spaces. A practical solution is given in Theorem 5.8, where the conditioning set is further reduced to $\mathbf{M}_i$ a Markov boundary of Y_i in $\mathbf{X}$. Again, the search for ILFs now relies on our ability to infer Markov boundaries, for which there exists a wealth of practical algorithmic solutions. Therefore, Theorem 5.8 offers a convenient way to recover the ILFs when the Composition assumption holds. It should be emphasized that, under the Composition assumption, if $\{Y_i\} \perp\!\!\!\perp \{Y_j\} \mid \mathbf{M}_i$ (respectively $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid \mathbf{M}_i$) holds for a particular Markov boundary, then it holds for every Markov blanket of Y_i in $\mathbf{X}$, and for every Markov blanket of Y_j in $\mathbf{X}$ (see the proof).

Thm. 5.8 *For each label* Y_i*, let* $\mathbf{M}_i$ *be an arbitrary Markov boundary of* Y_i *in* $\mathbf{X}$*. Let* $\mathcal{G}$ *be an undirected graph whose nodes correspond to the random variables in* $\mathbf{Y}$ *and in which two nodes* Y_i *and* Y_j *are adjacent* iff $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid \mathbf{M}_i$*. Then, each connected component in* $\mathcal{G}$ *is an ILF if* p *supports the Composition property.*

It is usually unknown to what degree the Composition assumption holds in distributions encountered in practice. Some special distributions are known to satisfy the Composition property, for example the multivariate Gaussian distribution [Stu05]. The following example provides a case where the Composition property does not hold.

Ex. 5.5 *Consider* $\mathbf{Y} = \{Y_1, Y_2, Y_3\}$ *and* $\mathbf{X} = \emptyset$*, with* Y_1, Y_2, Y_3 *three binary variables such that* $Y_1 = Y_2 \oplus Y_3$ *(*$\oplus$ *denotes the exclusive OR operator). The Markov boundary in* $\mathbf{X}$ *of each factor is necessarily* $\mathbf{M}_i = \emptyset$*. If we build the graph from Theorem 5.8, we may deduce that* $\{Y_1\}$ *is a label factor because* $\{Y_1\} \perp\!\!\!\perp \{Y_2\} \mid \emptyset$ *and* $\{Y_1\} \perp\!\!\!\perp \{Y_3\} \mid \emptyset$*. Clearly this is wrong because* $\{Y_1\} \not\perp\!\!\!\perp \{Y_2, Y_3\} \mid \mathbf{X}$*.*

For general distributions

We show next that, for any probability distribution, the ILFs can also be identified with $O(m^2)$ statistical tests of independence. Compared to the previous characterizations, the tests must now be made in sequential order and are in the form $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid \mathbf{X} \cup \mathbf{Z}$, with $\mathbf{Z}$ a subset of $\mathbf{Y} \setminus \{Y_i, Y_j\}$.

Thm. 5.9 *Consider* $<$ *a strict total order of* $\mathbf{Y}$*. Let* $\mathcal{G}$ *be an undirected graph whose nodes correspond to the labels, obtained from the following procedure:*

1: $\mathcal{G} \leftarrow (\mathbf{Y}, \emptyset)$ *(empty graph)*
2: ***for all*** $Y_i \in \mathbf{Y}$ ***do***
3: $\quad \mathbf{Y}^i_{ind} \leftarrow \emptyset$
4: $\quad$ ***for all*** $Y_j \in (Y | Y > Y_i)$ *(processed in* $<$ *order)* ***do***
5: $\quad\quad$ ***if*** $Y_i \perp\!\!\!\perp Y_j \mid \mathbf{X} \cup \{Y | Y < Y_i\} \cup \mathbf{Y}^i_{ind}$ ***then***
6: $\quad\quad\quad \mathbf{Y}^i_{ind} \leftarrow \mathbf{Y}^i_{ind} \cup \{Y_j\}$
7: $\quad\quad$ ***else***
8: $\quad\quad\quad$ *Insert a new edge* (i, j) *in* $\mathcal{G}$

Then, each connected component in $\mathcal{G}$ *is an ILF.*

Note that the graph obtained from the explicit procedure in Theorem 5.9 may vary when different label orderings are considered. Still, what matters is that the resulting graph always exhibits the same set of connected components, and thereby the same ILF decomposition. For the sake of illustration, let us apply Theorem 5.9 to our DAG examples, and confirm that we obtain the same ILFs as previously.

Ex. 5.6 *Consider the three DAGs in Figure 5.2, and let us identify the ILFs with Theorem 5.9. Take the natural ordering* $\{Y_1, Y_2, Y_3\}$ *and consider the DAG (a), we have* $\{Y_1\} \perp\!\!\!\perp \{Y_2\} \mid \mathbf{X}$ *so* Y_1 *and* Y_2 *are not adjacent in* $\mathcal{G}$*. Then,* Y_2 *is added to the conditioning set, and we have* $\{Y_1\} \perp\!\!\!\perp \{Y_3\} \mid \mathbf{X} \cup \{Y_2\}$ *so* Y_1 *and* Y_3 *are not adjacent either. We proceed the*

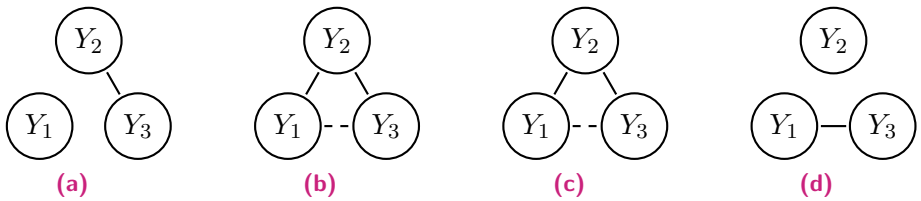


Fig. 5.6. Theorem 5.9 applied to our example DAGs (no assumption). Dashes indicate edges that may be present or absent depending on the label ordering used.

same way with the second label to obtain $Y_2 \not\perp\!\!\!\perp \{Y_3\} \mid \mathbf{X} \cup \{Y_1\}$*, which indicates the presence of an edge between* Y_2 *and* Y_3*. In the DAG (b), we have* $\{Y_1\} \not\perp\!\!\!\perp \{Y_2\} \mid \mathbf{X}$*, then* $\{Y_1\} \not\perp\!\!\!\perp \{Y_3\} \mid \mathbf{X}$ *and* $\{Y_2\} \perp\!\!\!\perp \{Y_3\} \mid \mathbf{X} \cup \{Y_1\}$ *so the graph* $\mathcal{G}$ *is complete. Had we taken another ordering, we would have ended up with a different graph. For example, with the ordering* $\{Y_2, Y_1, Y_3\}$*, the edge* $Y_1 - Y_3$ *is absent in* $\mathcal{G}$*. In the DAG (c), we have* $\{Y_1\} \not\perp\!\!\!\perp \{Y_2\} \mid \mathbf{X}$*, then* $\{Y_1\} \perp\!\!\!\perp \{Y_3\} \mid \mathbf{X}$ *and* $\{Y_2\} \not\perp\!\!\!\perp \{Y_3\} \mid \mathbf{X} \cup \{Y_1\}$ *so only the edge* $Y_1 - Y_3$ *is missing. With the ordering* $\{Y_2, Y_1, Y_3\}$*, the edge* $Y_1 - Y_3$ *is present. Finally, in the DAG (d), we have* $\{Y_1\} \perp\!\!\!\perp \{Y_2\} \mid \mathbf{X}$*, then* $\{Y_1\} \not\perp\!\!\!\perp \{Y_3\} \mid \mathbf{X} \cup \{Y_2\}$ *and* $\{Y_2\} \perp\!\!\!\perp \{Y_3\} \mid \mathbf{X} \cup \{Y_1\}$ *so only the edge* $Y_1 - Y_3$ *is present. We notice that, while the structure of* $\mathcal{G}$ *may differ according to the chosen label ordering, the connected components remain unchanged as expected. The resulting graphs are displayed in Figure 5.6.*

Here again, the number of conditional independence tests required in Theorem 5.9 is quadratic in the number of labels. Compared to the previous characterizations under the Intersection and the Composition properties, this new result has the desirable advantage of requiring no assumption about the underlying distribution p. However, it suffers from two limitations: i) the conditioning set at line 5 ranges from $\mathbf{X}$ in the first iteration to $\mathbf{X} \cup \mathbf{Y} \setminus \{Y_i, Y_j\}$ in the last iteration, which is problematic in high-dimensional data sets; and ii) the whole procedure is prone to error propagation, since each iteration depends on the result of the previous tests to constitute the $\mathbf{Y}^i_{ind}$ set. These problems remain serious impediments to the practical application of this characterization. Also, the procedure to build the graph can not be fully run in parallel, contrary to the ones in Theorems 5.5 and 5.7.

At this point is is worth pausing to summarize the results obtained so far for characterizing the ILFs. We established that finding the ILFs boils down to searching for connected components in an undirected graph, that can be constructed edge by edge with a sequence of statistical independence tests. It appears that by considering several (reasonable) assumptions about the underlying probability distribution p, practical procedures may be derived which imply learning a Bayesian network structure (under the DAG-faithfulness assumption), or learning a Markov boundary for every label (under the Intersection and Composition assumptions).

5.1.3 Minimal feature subsets

The second fundamental problem that we wish to address involves finding optimal feature subsets in the multi-label context, with respect to an Information Theory criterion [KS96]. The feature subset selection problem in MLC has received some attention in the last years [LK15; GEA14; LK13; Spo+13]. However, to our knowledge, the empirical work developed in these studies has not yet been underpinned by theoretical results. In this section we establish useful connections between marginal and joint Markov blankets (or Markov boundaries when applicable), under several assumptions about the underlying probability distribution.

Under the DAG-Faithfulness assumption

When p is faithful to a DAG $\mathcal{G}$, the joint Markov boundary of a label set can be depicted graphically in terms of the parent, child and spouse nodes of the labels,

Thm. 5.10 *Let $\mathbf{Y}_S = \{Y_1, Y_2, \ldots, Y_n\}$ be any label subset. Then, when p is faithful to a DAG $\mathcal{G}$, the Markov boundary of $\mathbf{Y}_S$ in $\mathbf{U}$ is given by $\mathbf{M} = \bigcup_{i=1}^{n} (\mathbf{PC}_{Y_i} \cup \mathbf{SP}_{Y_i}) \setminus \mathbf{Y}_S$.*

Note that when $\mathbf{M}$ contains no features, then it is also a Markov boundary of $\mathbf{Y}_S$ in $\mathbf{X}$. Therefore, when p is faithful to a DAG, the joint Markov boundary of $\mathbf{Y}$ in $\mathbf{X}$ consists in the set of all features which appear as a parent, child or spouse node of a label in $\mathbf{Y}$. Also, due to the characterization of ILFs in Theorem 5.4, it can be shown that the set of parent, child and spouse nodes of a label in an ILF does not contain any label outside those in the same ILF. Therefore, when p is faithful to a DAG, the joint Markov boundary of an ILF $\mathbf{Y}_F$ in $\mathbf{X}$ consists in the set of all features which appear as a parent, child or spouse node of a label in $\mathbf{Y}_F$.

Under the Intersection assumption

When p supports the Intersection property, the joint Markov boundary of a label set can be recovered as follows,

Thm. 5.11 *Let $\mathbf{Y}_S$ be any label subset, $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ a partition of $\mathbf{Y}_S$, and $\mathbf{M}_i$ a Markov boundary of $\mathbf{Y}_i$ in $\mathbf{U} \setminus \bigcup_{j=1}^{i} \mathbf{Y}_j$. Then, $\mathbf{M} = \bigcup_{i=1}^{n} \mathbf{M}_i \setminus \mathbf{Y}_S$ is a Markov blanket for $\mathbf{Y}_S$ in $\mathbf{U}$, and a Markov boundary when p supports the Intersection property.*

Again, if $\mathbf{M}$ contains no features, then it is also a Markov boundary (resp. Markov blanket) in $\mathbf{X}$. Therefore, when p supports the Intersection property, the joint Markov boundary of $\mathbf{Y}$ in $\mathbf{X}$ consists in all the features which appear in the marginal

Markov boundary of each single label $Y_1, Y_2, \ldots, Y_m$ respectively in $\mathbf{U}$, $\mathbf{U} \setminus \{Y_1\}, \ldots, \mathbf{U} \setminus \{Y_1, \ldots, Y_{m-1}\}$. Also, due to the characterization of ILFs in Theorem 5.6, it can be shown that the same procedure applied to an ILF $\mathbf{Y}_F$ necessarily results in the joint Markov boundary of $\mathbf{Y}_F$ in $\mathbf{X}$ when p supports the Intersection property. Still, Theorem 5.11 requires the Intersection assumption, as illustrated in the following example.

Ex. 5.7 *Consider $\mathbf{Y} = \{Y_1, Y_2\}$ and $\mathbf{X} = \{X_1, X_2\}$, with Y_1, Y_2, X_1, X_2 four random variables such that $Y_1 = Y_2 = X_1 = X_2$. Clearly p does not support the Intersection property. Suppose we apply Theorem 5.11 with $\mathbf{Y}_1 = \{Y_1\}$ and $\mathbf{Y}_2 = \{Y_2\}$. Then multiple marginal Markov boundaries exist, and $\mathbf{M}_1$ can be one of $\{\{X_1\}, \{X_2\}, \{Y_2\}\}$ and $\mathbf{M}_2$ one of $\{\{X_1\}, \{X_2\}\}$. Depending on which marginal Markov boundaries are use, $\mathbf{M}$ can be one of $\{\{X_1\}, \{X_2\}, \{X_1, X_2\}\}$. Clearly $\{X_1, X_2\}$ is not a Markov boundary of $\mathbf{Y}$.*

Under the Composition assumption

When p supports the Composition property, the joint Markov blanket of a label set can be recovered as follows,

Thm. 5.12 *Let $\mathbf{Y}_S$ be any label subset, $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ a partition of $\mathbf{Y}_S$, and $\mathbf{M}_i$ a Markov boundary of $\mathbf{Y}_i$ in $\mathbf{X}$. Then, $\mathbf{M} = \bigcup_{i=1}^{n} \mathbf{M}_i$ is a Markov blanket for $\mathbf{Y}_S$ in $\mathbf{X}$ when p supports the Composition property, and a Markov boundary when p also supports the Intersection property.*

As a result, it appears that multi-label feature selection based on single label feature selection requires i) the Composition property to ensure a correct feature subset, and ii) the Intersection property to ensure a minimal feature subset. This is illustrated in the following example.

Ex. 5.8 *Consider $\mathbf{Y} = \{Y_1, Y_2\}$ and $\mathbf{X} = \{X_1\}$, with $Y_1 = X_1 \oplus Y_2$ ($\oplus$ denotes the exclusive OR operator). Clearly the Composition property does not hold, yet the Intersection does. The marginal Markov boundaries are $\mathbf{M}_1 = \emptyset$ and $\mathbf{M}_2 = \emptyset$, so Theorem 5.12 yields $\mathbf{M} = \emptyset$ and fails to identify the dependency between X_1 and $\{Y_1, Y_2\}$. Consider Theorem 5.11 instead. We have $\mathbf{M}_1 = \{X_1, Y_2\}$ and $\mathbf{M}_2 = \emptyset$, which yields the correct joint Markov boundary $\mathbf{M} = \{X_1\}$.*

For general distributions

Finally, for any probability distribution p, the joint Markov blanket of a label set can be recovered according to Theorem 5.11.

5.1.4 Algorithms

In view of the theoretical analysis conducted in the last section, we shall propose three generic procedures for recovering a partition of (irreducible) label factors and their respective (minimal) feature subsets. These strategies are termed *ILF-DAG*, *ILF-Inter* and *ILF-Compo* in reference to their respective assumptions about the underlying probability distribution, that is, DAG-faithfulness, Intersection and Composition.

Generic procedures

- **ILF-DAG** (Algorithm 11): When p is faithful to a DAG. The procedure goes as follows: 1) learn the BN structure; 2) start from any label, say Y_i, recover its parents, children and spouses in the DAG, and keep doing it for every label recovered this way. At convergence, the set of all the recovered labels forms an ILF according to Theorem 5.4, and the set of all the recovered features forms its minimal feature subset according to Theorem 5.10. 3) repeat the last step with any label that has not been recovered yet, until all ILFs have been recovered. This approach has already been successfully used in Gasse et al. [GAE14].

- **ILF-Inter** (Algorithm 12): When p supports the Intersection property. The procedure goes as follows: 1) initialize the set of all previously processed labels, $\mathbf{Y}_{done} = \emptyset$; 2) start from any label, say Y_i, add it to $\mathbf{Y}_{done}$, recover its Markov boundary in $\mathbf{U} \setminus \mathbf{Y}_{done}$, and keep doing it for every label recovered this way. At convergence, the set of all the recovered labels forms an ILF according to Theorem 5.6, and the set of all the recovered features forms its minimal feature subset according to Theorem 5.11. 3) repeat the last step with any label that has not been recovered yet, until all ILFs have been recovered. Note that even if the Intersection property is violated, the recovered label sets are still label factors, and the recovered feature sets are valid (not necessarily minimal) feature subsets.

- **ILF-Compo** (Algorithm 13): When p supports the Composition property. The procedure goes as follows: 1) for every label Y_i recover $\mathbf{M}_i$ its Markov boundary in $\mathbf{X}$; 2) start from any label, say Y_i, recover any label such that $Y_i \not\perp\!\!\!\perp Y_j \mid \mathbf{M}_i$ or $Y_i \not\perp\!\!\!\perp Y_j \mid \mathbf{M}_j$ with a statistical independence test, and keep doing it for every label recovered this way. At convergence, the set of all the recovered labels forms an ILF according to Theorem 5.8, and the union of their Markov boundaries forms a (not necessarily minimal) feature subset according

Algorithm 11 ILF-DAG

Require: $\mathcal{D}$ a data set, $\mathbf{X}$ the set of features, $\mathbf{Y}$ the set of labels, BN_{alg} a Bayesian network structure learning algorithm.
Ensure: $\mathcal{P}_{\mathbf{Y}}$ a partition of $\mathbf{Y}$ and $\mathcal{S}_{\mathbf{X}}$ a family of subsets of $\mathbf{X}$.

Initialize $\mathcal{P}_{\mathbf{Y}} \leftarrow \emptyset$, $\mathcal{S}_{\mathbf{X}} \leftarrow \emptyset$, $\mathbf{Y}_{done} \leftarrow \emptyset$
Compute $\mathcal{G}$ the Bayesian network structure which covers $\mathbf{X} \cup \mathbf{Y}$ using BN_{alg}
while $\mathbf{Y} \setminus \mathbf{Y}_{done} \neq \emptyset$ **do**
    Select arbitrarily one label Y_i in $\mathbf{Y} \setminus \mathbf{Y}_{done}$
    Initialize $\mathbf{Y}_F \leftarrow \{Y_i\}$, $\mathbf{M}_F \leftarrow \emptyset$
    while $\mathbf{Y}_F \setminus \mathbf{Y}_{done} \neq \emptyset$ **do**
        Select arbitrarily one label Y_j from $\mathbf{Y}_F \setminus \mathbf{Y}_{done}$
        Recover $\mathbf{M}_j$ the set of parent, child and spouse nodes of Y_j in $\mathcal{G}$
        Add $\mathbf{M}_j \cap \mathbf{X}$ to $\mathbf{M}_F$
        Add $\mathbf{M}_j \cap \mathbf{Y}$ to $\mathbf{Y}_F$
        Add Y_j to $\mathbf{Y}_{done}$
    Add $\mathbf{Y}_F$ to $\mathcal{P}_{\mathbf{Y}}$
    Add $\mathbf{M}_F$ to $\mathcal{S}_{\mathbf{X}}$

Algorithm 12 ILF-Inter

Require: $\mathcal{D}$ a data set, $\mathbf{X}$ the set of features, $\mathbf{Y}$ the set of labels, MB_{alg} a Markov boundary learning algorithm.
Ensure: $\mathcal{P}_{\mathbf{Y}}$ a partition of $\mathbf{Y}$ and $\mathcal{S}_{\mathbf{X}}$ a family of subsets of $\mathbf{X}$.

while $\mathbf{Y} \setminus \mathbf{Y}_{done} \neq \emptyset$ **do**
    Select arbitrarily one label Y_i from $\mathbf{Y} \setminus \mathbf{Y}_{done}$
    Initialize $\mathbf{Y}_F \leftarrow \{Y_i\}$, $\mathbf{M}_F \leftarrow \emptyset$
    while $\mathbf{Y}_F \setminus \mathbf{Y}_{done} \neq \emptyset$ **do**
        Select arbitrarily one label Y_j from $\mathbf{Y}_F \setminus \mathbf{Y}_{done}$
        Compute $\mathbf{M}_j$ a Markov boundary of Y_j in $\mathbf{U} \setminus \mathbf{Y}_{done}$ using MB_{alg}
        Add $\mathbf{M}_j \cap \mathbf{X}$ to $\mathbf{M}_F$
        Add $\mathbf{M}_j \cap \mathbf{Y}$ to $\mathbf{Y}_F$
        Add Y_j to $\mathbf{Y}_{done}$
    Add $\mathbf{Y}_F$ to $\mathcal{P}_{\mathbf{Y}}$
    Add $\mathbf{M}_F$ to $\mathcal{S}_{\mathbf{X}}$

to Theorem 5.12. 3) repeat the last step with any label that has not been recovered yet, until all ILFs have been recovered. Note that if p also supports the Intersection property, then the recovered feature subsets are minimal. This approach has recently been discussed in Gasse et al. [GAE15].

While it is important to decompose $p(\mathbf{y}|\mathbf{x})$ as much as possible, it is even more important not to decompose it when such a factorization does not exist. Table 5.1 gives an overview of the theoretical capabilities of each of our procedures, according to the validity of our assumptions about the underlying distribution p. At first sight, ILF-Inter is conceptually the most promising procedure as it does guarantee to output correct label factors and feature subsets in every situation (but not necessarily irreducible ones). In contrast, ILF-Compo is not guaranteed to output correct label

Algorithm 13 ILF-Compo

Require: $\mathcal{D}$ a data set, $\mathbf{X}$ the set of features, $\mathbf{Y}$ the set of labels, MB_{alg} a Markov boundary learning algorithm, $(\cdot \perp\!\!\!\perp \cdot \mid \cdot)$ a statistical test of conditional independence.

Ensure: $\mathcal{P}_{\mathbf{Y}}$ a partition of $\mathbf{Y}$ and $\mathcal{S}_{\mathbf{X}}$ a family of subsets of $\mathbf{X}$.

Initialize $\mathcal{P}_{\mathbf{Y}} \leftarrow \emptyset$, $\mathcal{S}_{\mathbf{X}} \leftarrow \emptyset$, $\mathbf{Y}_{done} \leftarrow \emptyset$
for all $Y_i \in \mathbf{Y}$ **do**
    Compute $\mathbf{M}_i$ a Markov boundary of Y_i in $\mathbf{X}$ using MB_{alg}
while $\mathbf{Y} \setminus \mathbf{Y}_{done} \neq \emptyset$ **do**
    Select arbitrarily one label Y_i from $\mathbf{Y} \setminus \mathbf{Y}_{done}$
    Initialize $\mathbf{Y}_F \leftarrow \{Y_i\}$, $\mathbf{M}_F \leftarrow \mathbf{M}_i$
    while $\mathbf{Y}_F \setminus \mathbf{Y}_{done} \neq \emptyset$ **do**
        Select arbitrarily one label Y_j from $\mathbf{Y}_F \setminus \mathbf{Y}_{done}$
        for all $Y_k \in \mathbf{Y} \setminus (\mathbf{Y}_{done} \cup \mathbf{Y}_F)$ **do**
            if $\{Y_j\} \not\perp\!\!\!\perp_{\mathcal{D}} \{Y_k\} \mid \mathbf{M}_j$ **or** $\{Y_j\} \not\perp\!\!\!\perp_{\mathcal{D}} \{Y_k\} \mid \mathbf{M}_k$ **then**
                Add Y_k to $\mathbf{Y}_F$
                Add $\mathbf{M}_k$ to $\mathbf{M}_F$
        Add Y_j to $\mathbf{Y}_{done}$
    Add $\mathbf{Y}_F$ to $\mathcal{P}_{\mathbf{Y}}$
    Add $\mathbf{M}_F$ to $\mathcal{S}_{\mathbf{X}}$

Tab. 5.1. Theoretical capabilities of each method when p satisfies the DAG faithfulness, Intersection and Composition assumptions. The label sets returned by each method may be guaranteed to be irreducible label factors (ILF), or correct but not necessarily irreducible label factors (LF). Likewise, the feature subset returned for each label factor may be either a Markov boundary in $\mathbf{X}$ (MB), or a Markov blanket in $\mathbf{X}$ (M).

DAG-faith.	Intersection	Composition	ILF-DAG		ILF-Inter		ILF-Compo	
X	X	X	ILF	MB	ILF	MB	ILF	MB
-	X	X	-	-	ILF	MB	ILF	MB
-	X	-	-	-	ILF	MB	-	-
-	-	X	-	-	LF	M	ILF	M
-	-	-	-	-	LF	M	-	-

factors and feature subsets when the Composition property is violated. Still, in the particular setting where only the Composition property holds, ILF-Compo may be preferable to ILF-Inter to identify the true ILF decomposition. An empirical comparison of these three procedures will be performed in the next section, on a set of carefully designed synthetic experiments.

The Intersection and Composition axioms are not a universally valid properties of probabilistic independence models (Section 1.1.3), and neither does one imply the other. They appear to be essential properties though, as they often loosen the computational burden involved in statistical queries [Peñ+06; KT05]. It is considered reasonable to assume Intersection whenever there is uncertainty about the data, due for instance to measurement noise [Pea89] (i.e., all assignments of

the domain variables have a non zero probability, $p > 0$). Typical problems violating the positivity condition involve noise-free data such as logic propositions. While most data distributions encountered in practical regression or classification tasks are strictly positive, this is not the case in MLC problems as pairwise *positive entailment* relationships (e.g., river $\rightarrow$ water and car $\rightarrow$ vehicle) and/or mutually exclusive labels (e.g., four seasons: autumn, winter, spring and summer) often exist among the labels [PTT15]. On the other hand, the Composition axiom is violated when, for instance, the variables exhibit an exclusive-or relationship (Examples 5.5 and 5.8).

Implementation

The three generic procedures *ILF-DAG*, *ILF-Inter* and *ILF-Compo* are mathematically sound when the assumptions about p are met. They however rely on specific algorithmic procedures as subroutines for learning a BN structure (ILF-DAG), recovering a Markov boundary (ILF-Inter, ILF-Compo) or performing a conditional independence test (ILF-Compo). While there exists a wealth of such procedures in the literature, in most cases these offer only asymptotic guarantees, and are fallible with limited data sets. In addition, in typical MLC problems the distribution of the labels is known to be highly unbalanced, which also contributes to degrading the accuracy of statistical tests. With those considerations in mind, we propose a straightforward implementation of our three procedures in order to corroborate our theoretical findings empirically.

We chose to implement the three procedures upon the *bnlearn* R package from Scutari [Scu10] that offers plenty of practical procedures for Bayesian network structure learning, Markov boundary discovery and statistical independence testing. For a fair comparison of our three generic strategies, within ILF-DAG we employ the Bayesian network structure learning algorithm proposed by Margaritis and Thrun [MT99], which also relies on a Markov boundary discovery algorithm as a subroutine. Then, in our three procedures we employ the KIAMB algorithm, a powerful constraint-based method proposed by Peña et al. [Peñ+07] that is able to return any Markov boundary with non-zero probability (when multiple Markov boundaries exist). Finally, we employ a discrete semi-parametric permutation-based mutual information independence test, as advocated by Tsamardinos and Borboudakis [TB10]. We run the test with 100 permutations, and binarized continuous variables. Compared to classical asymptotic tests, permutation tests are better calibrated, that is, the actual Type I error is closer to the significance level α set by the user. Note that the main caveat in our implementation choice is that KIAMB relies on the Composition property for correctness, so our implementation of ILF-Inter, although "optimal" in a certain sense under Intersection, will suffers from certain limitations.

In fact, the correctness of all constraint-based Markov boundary learning algorithms that appeared in the recent literature (i.e., IAMB [TAS03a], Grow-Shrink [MT99], MMMB [TAS03b], MBOR [MA10a], PCMB [Peñ+07]) also rely on the Composition property, as well as any forward approach to feature subset selection [GE03]. Therefore, it appears rather difficult to implement ILF-Inter efficiently without also assuming the Composition property.

Regarding running-time efficiency, the actual complexity of our proposed methods is closely tied to the complexity of the approximation algorithms which are used as subroutines. Markov boundary and BN structure learning are both NP-hard problems with respect to the number of variables considered, while performing a parametric statistical test of independence requires a complete pass through the data set and therefore scales linearly with the number of samples. In the end, the time complexity of ILF-DAG is $O(m + C_{BN}(m + d))$, where $C_{BN}(m + d)$ is the time complexity of algorithm BN_{alg} with $m+d$ variables. The time complexity of ILF-Inter is $O(mC_{MB}(m + d))$, where $C_{MB}(m + d)$ is the time complexity of algorithm MB_{alg} with $m + d$ variables, and that of ILF-Compo is $O(m^2 s + mC_{MB}(d))$.

Experimental validation

Let us now corroborate our theoretical findings by means of empirical evidence. As it is unknown to what degree the DAG-Faithfulness, Intersection and Composition properties hold in distributions encountered in practice, we set up a simple toy problem to investigate the detrimental side effects that may arise in the learning process of ILF-DAG, ILF-Inter and ILF-Compo when one or several assumptions are not valid. Note that we do not investigate empirically the feature subset selection problem, even though our proposed procedures are theoretically able to perform feature subset selection.

Consider the directed acyclic graph $\mathcal{G}$ depicted in Figure 5.7, which consists of six binary variables X_1, X_2, ..., X_6 and six labels Y_1, Y_2, ..., Y_6. The initial probability distribution we consider is faithful to $\mathcal{G}$, with the conditional probability tables presented in Table 5.8a. Now, we may easily violate the DAG-faithfulness assumption (and yet retain the Intersection and Composition properties), by considering X_4 as a hidden variable. Indeed, there exists no DAG that encodes faithfully all the remaining conditional independence relations. We shall also violate the Composition assumption by setting up a non-deterministic XOR relationship between variables Y_5, X_6 and Y_6. To do so, we replace the conditional probability table of X_6 with that of Table 5.8b. It can be observed now that $Y_5 \perp\!\!\!\perp X_6$, $Y_5 \perp\!\!\!\perp Y_6$, and $Y_5 \not\perp\!\!\!\perp \{X_6, Y_6\}$ which is inconsistent with the Composition property. Finally, we shall also violate

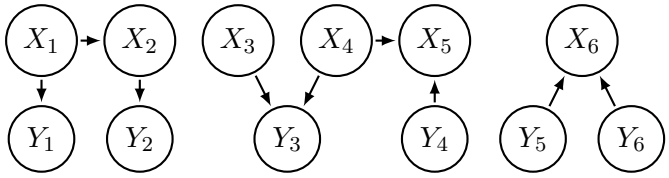


Fig. 5.7. DAG of toy problem 1.

Tab. 5.2. Pairwise F_2 measure (mean $\pm$ std in percent) of the decomposition output by each method versus the optimal decomposition, over 1000 runs with 5000 samples (higher is better).

Scenario	ILF-DAG	ILF-Inter	ILF-Compo
DAG-Faithfulness	100.0 ± 0.5	99.9 ± 1.4	99.9 ± 1.5
Inters. + Compos.	77.4 ± 22.2	90.5 ± 18.2	93.1 ± 16.1
Intersection	28.2 ± 27.8	45.2 ± 21.7	47.8 ± 19.3
Composition	77.3 ± 22.2	88.3 ± 18.0	93.1 ± 16.1
Worst case	27.6 ± 27.8	43.6 ± 22.0	46.8 ± 20.3

the Intersection assumption by establishing a deterministic relationship between nodes X_1, Y_1 and X_2. To do so, we shall replace the conditional probability table of Y_1 and X_2 with that in Table 5.8c. It can be observed now that $Y_1 \perp\!\!\!\perp X_2 \mid \{X_1, Y_2\}$, $Y_1 \perp\!\!\!\perp X_1 \mid \{X_2, Y_2\}$ and $Y_1 \not\perp\!\!\!\perp \{X_1, X_2\} \mid Y_2$ which is inconsistent with the Intersection property.

To summarize, we alter the initial model in order to obtain five different distributions satisfying the following scenarios: i) p is DAG-faithful; ii) p is not DAG-faithful but still satisfies the Intersection and the Composition properties; iii) p only satisfies the Intersection property; iv) p only satisfies the Composition property; and v) p neither satisfies Intersection nor Composition (termed "Worst case" in the tables). To increase the difficulty of the Markov boundary discovery task, 14 irrelevant random binary variables $X_7, \ldots, X_{20}$ are added to each data set. We sample 1000 training sets of 5000 observations each from each scenario, and compare the decomposition output by ILF-DAG, ILF-Inter and ILF-Compo with the true ILF decomposition of each scenario, that is, $\mathcal{F}_I = \{\{Y_1\}, \{Y_2\}, \{Y_3\}, \{Y_4\}, \{Y_5, Y_6\}\}$ in the DAG-faithful scenario, and $\mathcal{F}_I = \{\{Y_1\}, \{Y_2\}, \{Y_3, Y_4\}, \{Y_5, Y_6\}\}$ in all the remaining scenarios. In this experiment we employ CI tests with significance level $\alpha = 0.01$.

To evaluate the ILF decomposition quality, we report in Table 5.2 the pairwise F_2 measure of each method compared to the ground truth decomposition. The idea is to view the ILF decomposition as a series of $m(m-1)/2$ decisions, one for each pair of labels, as to whether the pair belongs to the same ILF or not. Ideally, one would like to assign two labels to the same ILF if and only if they are in the true ILF. However, an excessive factorization should be penalized more than a missed factorization, therefore we report the pairwise F_2 measure to penalize false negatives more strongly than false positives.

	1	0
$p(X_1)$	0.5	0.5
$p(X_2 \mid X_1 = 0)$	0.2	0.8
$p(X_2 \mid X_1 = 1)$	0.8	0.2
$p(X_3)$	0.5	0.5
$p(X_4)$	0.5	0.5
$p(X_5 \mid X_4 = 0, Y_4 = 0)$	0.1	0.9
$p(X_5 \mid X_4 = 0, Y_4 = 1)$	0.6	0.4
$p(X_5 \mid X_4 = 1, Y_4 = 0)$	0.6	0.4
$p(X_5 \mid X_4 = 1, Y_4 = 1)$	0.8	0.2
$p(X_6 \mid Y_5 = 0, Y_6 = 0)$	0.1	0.9
$p(X_6 \mid Y_5 = 0, Y_6 = 1)$	0.6	0.4
$p(X_6 \mid Y_5 = 1, Y_6 = 0)$	0.6	0.4
$p(X_6 \mid Y_5 = 1, Y_6 = 1)$	0.8	0.2
$p(Y_1 \mid X_1 = 0)$	0.8	0.2
$p(Y_1 \mid X_1 = 1)$	0.2	0.8
$p(Y_2 \mid X_2 = 0)$	0.8	0.2
$p(Y_2 \mid X_2 = 1)$	0.2	0.8
$p(Y_3 \mid X_3 = 0, X_4 = 0)$	0.1	0.9
$p(Y_3 \mid X_3 = 0, X_4 = 1)$	0.6	0.4
$p(Y_3 \mid X_3 = 1, X_4 = 0)$	0.6	0.4
$p(Y_3 \mid X_3 = 1, X_4 = 1)$	0.2	0.8
$p(Y_4)$	0.5	0.5
$p(Y_5)$	0.5	0.5
$p(Y_6)$	0.5	0.5

(a) Faithful distribution.

	1	0
$p(X_6 \mid Y_5 = 0, Y_6 = 0)$	0.1	0.9
$p(X_6 \mid Y_5 = 0, Y_6 = 1)$	0.9	0.1
$p(X_6 \mid Y_5 = 1, Y_6 = 0)$	0.9	0.1
$p(X_6 \mid Y_5 = 1, Y_6 = 1)$	0.1	0.9

(b) XOR relationship between Y_5, X_6 and Y_6. The Composition property is violated.

	1	0
$p(X_2 \mid X_1 = 0)$	0.0	1.0
$p(X_2 \mid X_1 = 1)$	1.0	0.0
$p(Y_1 \mid X_1 = 0)$	0.0	1.0
$p(Y_1 \mid X_1 = 1)$	1.0	0.0

(c) Deterministic relationship between Y_1, X_1 and X_2. The Intersection property is violated.

Fig. 5.8. Conditional probability tables in our toy problem.

As expected, the global performance globally decreases as we move from the easiest scenario (DAG-Faithfulness) to the admittedly most difficult one (i.e., neither Intersection nor Composition properties hold). Nonetheless, the detrimental side effects in the learning process are more pronounced when the Composition property is violated. Indeed, a drastic drop of performance in terms of F_2 measure is observed for the Intersection and the worst case scenarios. Second, while all methods perform equivalently under the Faithfulness assumption, ILF-DAG fails spectacularly when the Composition is not valid. Learning a DAG prior to extracting the ILFs is apparently not the best strategy when the distribution is not faithful to such a DAG. In the worst case, ILF-DAG is superseded by ILF-Inter and ILF-Compo in terms of F_2 measure. Third, ILF-Compo compares favorably to ILF-Inter and ILF-DAG in all cases. The deceiving performance of ILF-Inter when the Composition is not valid is most likely due to KIAMB being incorrect for this subclass of distributions.

Overall, this toy problem clearly highlights the limitations of each method when some of the probabilistic assumptions on which they are based are not valid. Usually, practitioners have no indication about the probabilistic properties underlying their data at hand and whether the assumptions are testable. So, we single out ILF-Compo for its robustness and overall performance on this toy problem.

5.2 Application to subset zero-one loss minimization

This section presents a number of experimental studies to evaluate our ILF decomposition approach for MLC with subset $0/1$ loss minimization, using both synthetic and benchmark data. Our aim is not to perform a thorough comparison against state-of-the-art MLC algorithms but instead to corroborate our theoretical findings by means of empirical evidence. We first investigate on a toy problem the extent to which ILF-Compo (the best performing algorithm in our previous experiments) can help to solve the MLC problem under subset $0/1$ loss, for different label independence structures. Finally, we assess the ability of ILF-Compo to reduce the empirical subset $0/1$ loss on 8 real-world MLC benchmark data sets.

5.2.1 Factorized LP

The subset $0/1$ loss, introduced in Section 4.1.2, is a commonly applied performance metric in MLC studies. The point-wise risk-minimizing prediction is given by the mode of the joint distribution of the labels, $\arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$, a.k.a. the maximum a-posteriori estimate (MAP), or most probable expectation (MPE). Then, a straight-

forward approach is to cast the MLC problem as a single multinomial classification problem, by considering each label combination as a distinct meta-class. This scheme, introduced in Section 4.2.2, is called Label Powerset (LP) [TV07; TKV11], and is guaranteed to perform MAP inference in any situation. However, the number of meta-classes in LP is potentially exponential to the number of labels, which leads to tractability and robustness issues.

Given that an ILF decomposition exists, the MLC problem under subset $0/1$ loss decomposes nicely into a series of simpler sub-problems that can be solved independently, one for each label factor,

$$\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}) = \prod_{\mathbf{Y}_F \in \mathcal{F}_I} \max_{\mathbf{y}_F} p(\mathbf{y}_F|\mathbf{x}). \tag{5.1}$$

F-LP

In light of (5.1), applying the LP scheme on each ILF is guaranteed to results in MAP inference. We refer to this approach as *Factorized LP* (F-LP). The theoretical advantages are two-fold: i) stronger probability estimates, due to a reduced number of free parameters in $p(\mathbf{y}|\mathbf{x})$; and ii) smaller prediction times, due to the decomposition of the inference problem (5.1).

It should be emphasized that, when the ILFs are reduced to singletons, $p(\mathbf{y}|\mathbf{x})$ factorizes as the product of m marginal distributions $\prod_{i=1}^{m} p(y_i|\mathbf{x})$, and the problem of obtaining a MAP estimate of the labels boils down to training a separate binary classifier for each label. This simple scheme, introduced in Section 4.2.1, is called Binary relevance (BR) [Lua+12], and is well suited in this situation. In any other situation, applying the BR scheme is no longer guaranteed to result in MAP inference.

The F-LP procedure goes as follows: 1) run the ILF-Compo algorithm to obtain the ILF decomposition; 2) apply the LP scheme on each ILF to solve the MLC problem under subset $0/1$ loss. Overall, F-LP balances between BR and LP when $p(\mathbf{y}|\mathbf{x})$ accepts respectively a full decomposition or no decomposition at all. While theoretically sound, this whole procedure may not necessarily translate into a gain in performance (i.e., reduced subset $0/1$ loss on the test set) if the MLC problem is not decomposable, or if the true decomposition is not correctly identified by ILF-Compo.

5.2.2 Toy problem

The aim of this first experiment is to illustrate the impact of the ILF decomposition to minimize the *subset* $0/1$ *loss*. As F-LP includes BR and LP as special cases, we examine whether the decomposition translates to improved performances with respect to these two baseline algorithms. Consider the DAG depicted in Figure 5.9, which consists in a (single) discrete variable X with 16 modalities, and 5 labels

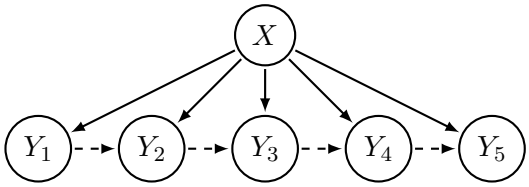


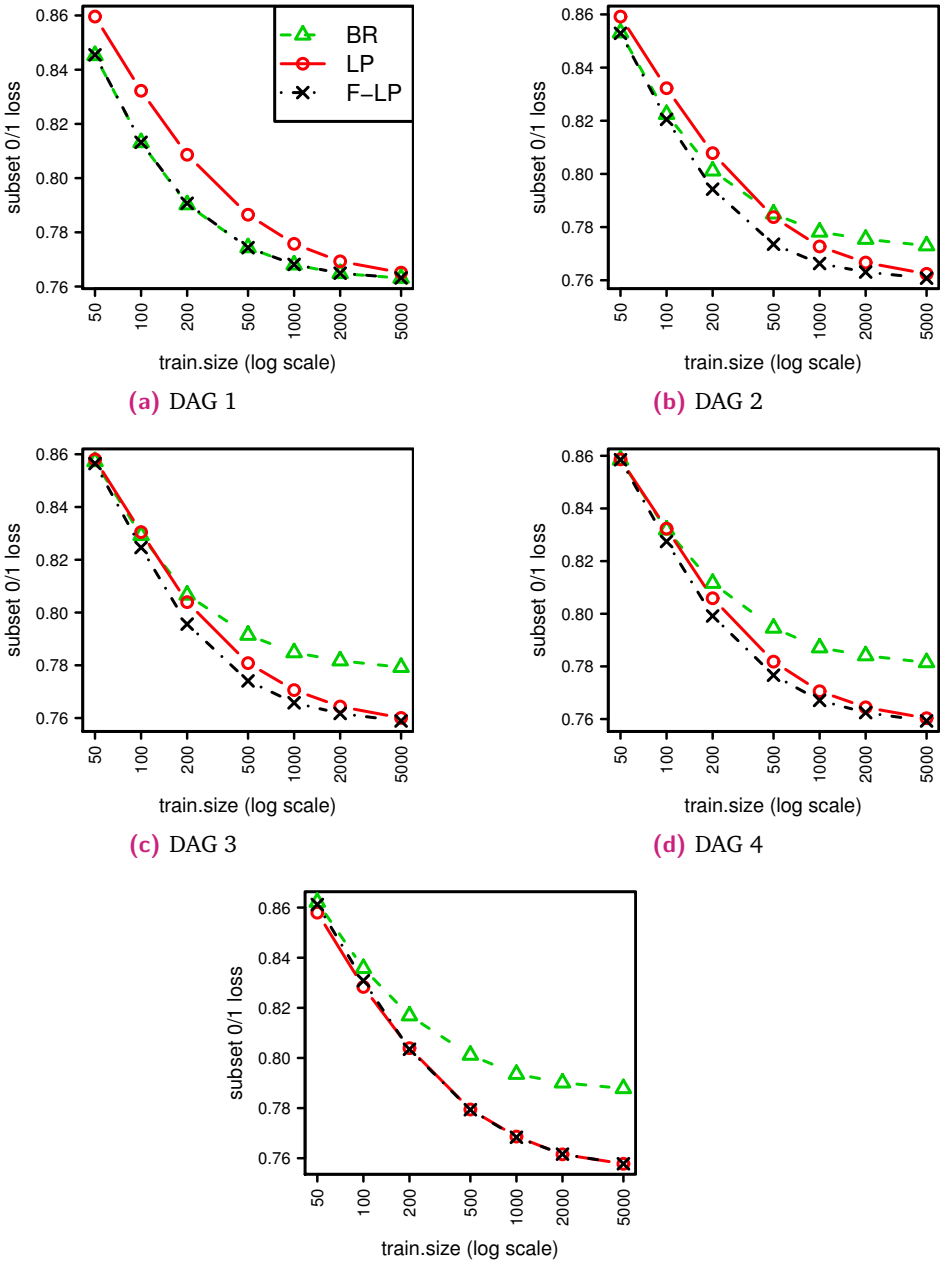
Fig. 5.9. BN structure of our toy problem. Dashed lines indicate possibly missing edges.

$Y_1, Y_2, \ldots, Y_5$. Due to the DAG structure, each label Y_i can only be made directly dependent of X and its neighbouring labels Y_{i-1} and Y_{i+1}. By removing intentionally certain short-dashed edges in the DAG, several groups of labels can be made made conditionally independent given X, thereby imposing a particular ILF decomposition. In order to evaluate the behaviour of F-LP on different scenarios, we consider five distinct ILF decompositions:

- DAG 1: $\mathcal{F}_I = \{\{Y_1\}, \{Y_2\}, \{Y_3\}, \{Y_4\}, \{Y_5\}\}$;
- DAG 2: $\mathcal{F}_I = \{\{Y_1, Y_2\}, \{Y_3, Y_4\}, \{Y_5\}\}$;
- DAG 3: $\mathcal{F}_I = \{\{Y_1, Y_2, Y_3\}, \{Y_4, Y_5\}\}$;
- DAG 4: $\mathcal{F}_I = \{\{Y_1, Y_2, Y_3, Y_4\}, \{Y_5\}\}$;
- DAG 5: $\mathcal{F}_I = \{\{Y_1, Y_2, Y_3, Y_4, Y_5\}\}$.

For each scenario, we generate random probability distributions by sampling uniformly the conditional probability table of each node in the DAG from a unit simplex, as discussed in Smith and Tromble [ST04]. The process is repeated 1000 times for each DAG, to obtain 5×1000 random probability distributions. From each distribution, we draw 7 training samples with respectively 50, 100, 200, 500, 1000, 2000 and 5000 instances and one testing sample with 5000 instances. F-LP, LP and BR are then run on each training set using the same multi-class base learner, a Random Forest classifier [LW02], and we evaluate performance of each method on the test set. In this experiment we run ILF-Compo with significance level $\alpha = 0.01$.

The MLC performance of F-LP, BR and LP in terms of subset $0/1$ loss is reported in Figure 5.10. As expected, LP (red curve) is asymptotically optimal with the sample size of the training set. Nonetheless, it happens that BR (green curve) outperforms LP on small sample sizes (< 500), as shown in Fig. 5.10b. The reason is that LP needs more observations to safely estimate $p(\mathbf{y} \mid \mathbf{x})$ than BR to estimate each $p(y_i \mid \mathbf{x})$. Second, the asymptotic difference between LP and BR is more pronounced as we move from DAG 1 to DAG 5. In fact, when all the labels are conditionally independent of each other (DAG 1), the ILFs are reduced to singletons and BR is optimal in terms of subset $0/1$ loss. In any other situation BR is no longer optimal, and the higher the conditional dependence between the labels is, the more BR and LP diverge asymptotically. Finally, F-LP (black curve) compares favorably to both BR (green curve) and LP (red curve) in all scenarios. When all successive labels are



(a) DAG 1 (b) DAG 2

(c) DAG 3 (d) DAG 4

(e) DAG 5

Fig. 5.10. Mean subset zero-one loss of each method on each DAG, with 7 different training sizes (50, 100, 200, 500, 1000, 2000, 5000) displayed on a logarithmic scale, averaged over 1000 repetitions with random distributions (lower is better).

Tab. 5.3. Pairwise F_2 measure (mean $\pm$ std in percent) of the decomposition output by ILF-Compo versus the true ILF decomposition, over 1000 runs (higher is better).

Sample size	DAG 1		DAG 2	DAG 3	DAG 4	DAG 5
50	99.3	8.3	2.6 ± 11.4	2.1 ± 7.5	1.6 ± 5.4	1.8 ± 4.6
100	95.3	21.2	28.9 ± 32.4	26.0 ± 21.1	19.6 ± 15.1	16.2 ± 11.6
200	92.3	26.7	76.6 ± 28.8	64.1 ± 18.9	47.8 ± 14.1	39.3 ± 10.5
500	89.7	30.4	98.1 ± 7.6	84.2 ± 10.2	64.9 ± 11.4	55.2 ± 9.2
1000	89.8	30.3	99.3 ± 2.7	91.2 ± 10.2	75.3 ± 12.2	65.0 ± 10.2
2000	90.5	29.3	99.1 ± 2.7	96.7 ± 7.4	84.8 ± 9.9	74.3 ± 8.7
5000	90.8	28.9	99.0 ± 2.9	99.3 ± 3.1	92.0 ± 7.5	82.3 ± 7.9

pairwise dependent then ILF-Compo boils down to LP (DAG 5), while at the opposite extreme it boils down to BR (DAG 1). Between these two extreme cases (DAGs 2, 3, 4), the decomposition found by ILF-Compo seems always beneficial to F-LP as it outperforms both BR and LP in terms of subset $0/1$ loss. Overall, these results are in nice agreement with our theoretical expectations.

Since we know the ground truth of the ILF decomposition for each scenario, here again we can evaluate the quality of the decomposition returned by ILF-Compo in terms F_2 measure, reported in Table 5.3. It is worth noting that, in the extreme case where almost all the labels are conditionally dependent (e.g., DAG 4 and 5), ILF-Compo can fail spectacularly to identify the correct ILFs with small sample sizes (< 200). This is clearly due to the lack of robustness of the statistical test employed with a limited amount of samples. However, this has little impact in terms of subset $0/1$ loss for small sample sizes since BR and LP perform poorly as well. Still, as the sample size increases the quality of the ILF decomposition becomes significantly better.

5.2.3 Real-world benchmark

We now report on an experiment performed on 8 real-world multi-label data sets. These come from different domains including text, biology, music and vision, with a number of labels ranging from 5 to 53. All data sets can be found on the Mulan[2] repository, except for *image* which comes from Zhou[3] [MR98]. Table 5.4 presents the main characteristics of each data set $\mathcal{D}$, where $|\mathcal{D}|$ indicates the number of examples, $dim(\mathcal{D})$ the number of features, $L(\mathcal{D})$ the number of labels, $F(\mathcal{D})$ the feature type and $DL(\mathcal{D})$ the number of distinct label combinations appearing in the data set. Of course, we have no idea about the true ILF decomposition for each of these data sets, therefore we also repeat the experiment on *augmented* data sets to ensure that the

[2] `http://mulan.sourceforge.net/datasets.html`
[3] `http://lamda.nju.edu.cn/data_MIMLimage.ashx`

Tab. 5.4. Benchmark data sets characteristics

name	domain	$\lvert\mathcal{D}\rvert$	$dim(\mathcal{D})$	$F(\mathcal{D})$	$L(\mathcal{D})$	$DL(\mathcal{D})$
emotions	music	593	72	cont.	6	27
image	images	2000	135	cont.	5	20
scene	images	2407	294	cont.	6	15
yeast	biology	2417	103	cont.	14	198
slashdot	text	3782	1079	disc.	22	156
genbase	biology	662	1186	disc.	27	32
medical	text	978	1449	disc.	45	94
enron	text	1702	1001	disc.	53	753

underlying conditional distribution $p(\mathbf{y}|\mathbf{x})$ factorizes into (at least) two ILFs. The augmented data sets are created as follows: we create a copy of the original data set, permute its rows, and merge it side-by-side with the original data set. So by design, these augmented data sets have twice as many features and twice as many labels as the original ones, for the same sample size. Using this method, we maintain the probabilistic structure of the original and duplicated parts, while imposing their mutual independence.

Comparative methods

In order to assess the effectiveness of the ILF decomposition scheme, in this experiment we compare F-LP to three baseline approaches for subset $0/1$ loss minimization, namely LP, PCC and its relaxed variant MCC. For information purposes we also measure the performance in terms of subset $0/1$ loss of a variety of other approaches commonly used in the MLC literature, namely BR, RAkEL, HOMER, CC and two of its variants ECC and LEAD. As we will see, each of these approaches relies on a particular decomposition of the MLC problem, which we believe is interesting to compare to our ILF decomposition scheme. Note that we do not include in this experiment other approaches to MLC such as CRF, S-SVM, MBC or CDN discussed in Chapter 4, since these rely on specific inference procedures and thus are difficult to compare to F-LP without using the same base learner. We now give a short overview of the compared methods, which is not intended to convey a detailed understanding of the algorithms but rather give a flavour of the various stages involved.

- **RAkEL** The *RAdom k labELsets* approach [TV07; TKV11], discussed in Section 4.2.5, is an ensemble method that has been proposed in order to overcome the computational burden of LP. It consists of several LP classifiers over randomly drawn subsets of labels, and is parametrized by the size of the random label subsets and the number of subsets to draw. The global prediction is ob-

tained by combining the LP predictions of all the label subset with a label-wise majority vote, that is, by counting for each label how many times it is predicted positive. Despite its intuitive appeal and competitive performance, RAKEL is still not well understood from a theoretical point of view. While for particular settings it reduces to BR (m subsets of size 1) or LP (1 subset of size m), and it is not clear which loss function it intends to minimize in general.

- **HOMER** The *Hierarchy Of Multi-label classiERs* approach [TKV08] is a hierarchical method that has been proposed as an effective and computationally efficient solution to MLC. Basically, HOMER constructs a tree that decomposes the label set hierarchically into disjoint subsets, from the root node that contains the whole label set to the leaf nodes that contain single labels. The splitting criterion relies on a clustering algorithm that partitions the current label set into k disjoint subset, so that similar labels are placed together and dissimilar apart. The classification scheme then follows the tree structure, with a baseline multi-label classifier that predicts for each node which of its child nodes contains one or more positive labels. According to the authors, the benefit of this method is a sub-linear prediction time with respect to the number of labels, due to the sparseness of most MLC problems. Still, the theoretical properties of the HOMER structure is not well understood, and it is not clear which loss function it intends to minimize in the end.

- **LEAD** The *multi-label Learning by Exploiting lAbel Dependency* approach [ZZ10], discussed in Section 4.2.3, learns a DAG structure from the error residuals of a BR classifier, and then adopts a CC scheme that follows the DAG structure. To some extent LEAD exploits conditional label dependencies to learn the DAG structure, however it inherits the greedy inference scheme of CC and it is not clear which loss function it minimizes.

- **PCC** The *Probabilistic Classifier Chain* approach [DCH10] was discussed in detail in Section 4.3.1. Basically, PCC learns a chain of binary probabilistic classifiers to model $p(\mathbf{y}|\mathbf{x})$, and performs exact MAP inference with an exhaustive search in $O(2^m)$ for the most probable label combination. Clearly, PCC is tailored for subset $0/1$ minimization, but in practice is limited to problems with small to moderate number of labels, typically not more than about 15.

- **MCC** The *Monte-Carlo Classifier Chain* approach [RML14] is basically a relaxed version of PCC, which decides on the best chaining order at training time with a random exploration scheme, and replaces the exhaustive search of PCC for inference with a random search. The inference procedure in MCC loosens the computational complexity of PCC, and is able to deal with larger-sized

problems. Still, while MCC is clearly tailored for subset $0/1$ loss, it can perform only approximate MAP inference.

- **CC** The *Classifier Chain* approach [Rea+09], discussed in Section 4.2.3, is a greedy version of PCC where exact MAP inference is replaced by a greedy approximation according to the chain order, which reduces the inference complexity to $O(m)$. The appealing property of CC is that it can account for label dependencies at the same cost as BR, and in practice it was shown to perform well in many cases. However, the greedy approximation scheme can lead to a high regret with respect to both Hamming loss and subset $0/1$ loss [DWH12], and is rather sensitive the ordering of the labels in the chain.

- **ECC** The *Ensemble of Classifier Chains* approach [Rea+09], discussed in Section 4.2.5, reduces the influence of the label ordering in CC, by averaging the multi-label predictions over a (randomly chosen) set of orderings, with a label-wise majority vote. Although ECC was shown to be more competitive than CC in terms of several evaluation metrics, the actual impact of the ensemble averaging for Hamming loss or subset $0/1$ loss remains unknown. We believe, as suggested in [DCH10], that the averaging used in ECC may bring the predictions closer to the marginals.

In this experiment we used the implementations from the *Mulan*[4] library [Tso+11] for F-LP, BR, LP, RAkEL and HOMER, and from the *Meka*[5] [Rea+16] library for PCC, CC, ECC and MCC. Both the Mulan and Meka libraries being based on *Weka*[6] [Hal+09], within each of these approaches we employed the same base learner, an SMO linear SVM, which we also employed within F-LP. For LEAD, we used the author's implementation[7] with a linear SVM classifier as base learner, and the K2 algorithm to learn the BN structure. We used the default parameters of each method without any tuning, and a significance value $\alpha = 0.0001$ within ILF-Compo. All experiments were performed on an Intel(R) Pentium CPU @3.60 GHz 8GB RAM.

Results

The performance of each compared method in terms of subset $0/1$ loss is reported in Table 5.5, averaged over a 5x2-fold cross-validation for each of the 16 data sets (original and duplicated). Note that several results are missing on large data sets for PCC, the most time demanding procedure, when it exceeded 4 hours of computations.

[4] http://mulan.sourceforge.net
[5] http://meka.sourceforge.net
[6] http://www.cs.waikato.ac.nz/ml/weka
[7] http://cse.seu.edu.cn/PersonalPage/zhangml

Tab. 5.5. Subset zero-one loss (mean $\pm$ std in percent) achieved by comparative methods on the original and the duplicated benchmark, over 5x2-CV. Best results are bold-faced (lower is better).

method	emotions	image	scene	yeast	slashdot	genbase	medical	enron
F-LP	**66.2 ± 2.5**	**53.7 ± 1.0**	31.8 ± 1.5	**75.1 ± 1.0**	59.1 ± 1.2	3.4 ± 1.0	32.2 ± 1.9	85.3 ± 1.2
LP	**66.2 ± 2.5**	**53.7 ± 1.0**	**31.5 ± 1.0**	**75.1 ± 1.0**	**55.0 ± 0.7**	3.8 ± 1.1	33.0 ± 1.3	**83.8 ± 1.1**
PCC	70.7 ± 1.9	59.7 ± 1.9	39.8 ± 1.0	79.6 ± 0.7	na	na	na	na
MCC	67.9 ± 2.0	57.3 ± 1.3	37.2 ± 1.4	79.8 ± 0.9	61.9 ± 0.7	3.4 ± 1.1	33.4 ± 1.9	88.1 ± 1.2
BR	73.6 ± 1.8	76.4 ± 1.7	49.0 ± 1.2	85.5 ± 0.9	66.2 ± 0.8	3.4 ± 1.1	35.9 ± 1.6	89.3 ± 1.2
RAkEL	69.3 ± 1.6	57.8 ± 0.9	39.4 ± 1.1	81.6 ± 0.7	65.3 ± 0.7	3.2 ± 1.0	35.6 ± 1.6	89.0 ± 1.1
HOMER	71.7 ± 1.8	68.4 ± 0.9	49.4 ± 3.2	86.9 ± 1.1	64.9 ± 0.8	3.4 ± 1.2	37.9 ± 3.3	89.7 ± 1.1
CC	71.6 ± 2.4	57.9 ± 1.1	37.0 ± 1.4	80.7 ± 0.7	62.0 ± 0.9	3.3 ± 1.2	32.7 ± 1.9	88.0 ± 1.2
ECC	70.6 ± 1.8	59.7 ± 1.2	37.7 ± 1.5	79.8 ± 0.6	60.3 ± 0.9	**3.1 ± 1.1**	**31.7 ± 2.2**	86.9 ± 1.1
LEAD	76.2 ± 2.0	70.2 ± 1.6	49.9 ± 1.9	85.4 ± 0.7	69.2 ± 0.7	3.8 ± 1.5	37.4 ± 1.0	91.8 ± 0.8

method	emotions2	image2	scene2	yeast2	slashdot2	genbase2	medical2	enron2
F-LP	**91.8 ± 1.4**	**82.0 ± 1.4**	**58.6 ± 1.3**	**95.0 ± 0.6**	**83.9 ± 0.7**	**6.8 ± 1.7**	**62.4 ± 2.5**	**98.4 ± 0.4**
LP	94.9 ± 1.1	87.6 ± 1.0	62.8 ± 1.7	97.5 ± 0.4	90.3 ± 0.4	33.7 ± 3.1	86.6 ± 1.6	99.3 ± 0.2
PCC	93.1 ± 1.6	85.9 ± 0.8	71.0 ± 0.5	na	na	na	na	na
MCC	93.6 ± 1.4	85.6 ± 1.1	67.9 ± 1.3	96.4 ± 0.4	86.6 ± 0.6	7.1 ± 1.7	64.4 ± 2.5	98.9 ± 0.5
BR	94.7 ± 1.3	93.7 ± 0.7	79.0 ± 1.1	98.0 ± 0.4	89.9 ± 0.6	**6.8 ± 1.7**	67.0 ± 2.8	99.1 ± 0.3
RAkEL	93.7 ± 0.8	89.7 ± 0.5	72.0 ± 1.4	97.8 ± 0.4	89.3 ± 0.6	**6.8 ± 1.8**	67.2 ± 2.9	99.2 ± 0.2
HOMER	95.5 ± 0.8	91.8 ± 0.9	79.9 ± 1.3	98.8 ± 0.5	97.0 ± 0.6	27.0 ± 3.7	82.1 ± 2.5	99.6 ± 0.3
CC	95.1 ± 1.0	83.9 ± 0.9	66.9 ± 1.2	96.5 ± 0.4	86.5 ± 0.5	7.1 ± 1.9	64.4 ± 2.8	99.0 ± 0.4
ECC	93.6 ± 1.6	84.8 ± 0.9	66.5 ± 1.8	97.0 ± 0.4	86.1 ± 0.4	7.2 ± 1.8	64.4 ± 2.8	98.7 ± 0.3
LEAD	95.9 ± 1.4	93.0 ± 0.6	80.5 ± 1.6	98.1 ± 0.4	91.3 ± 0.4	8.9 ± 1.6	65.5 ± 2.5	99.6 ± 0.2

The total running time of each method (for both training and testing) is reported in Table 5.6, while the running time spend by ILF-Compo to learn the decompositions is reported in Table 5.7. For clarity we defer to the appendix Figures A.1 to A.8, which display typical decomposition graphs learned by ILF-Compo on each data set (as defined in Theorem 5.8). For completeness we also report in the appendix several commonly applied evaluation measures, namely the Hamming loss, the micro-F_1 score and the macro-F_1 score as in [TKV10].

On the original benchmark the best performing approaches are clearly LP and F-LP. The two other approaches tailored for subset $0/1$ loss, PCC and its relaxed variant MCC, perform consistently well, while their greedy versions CC and ECC perform reasonably. As expected BR performs poorly in every situation. Among the remaining approaches, it is worth noting that the RAkEL approach performs always better than BR, while the HOMER and LEAD approaches perform rather poorly. Since LEAD follows the CC scheme for inference, the performance gap between both approaches may be imputed either to the learned DAG structure that may be too restrictive, or to the linear SVM that may be implemented differently than that of CC. On the other hand HOMER shares the same base learner as all the other methods, so its bad performance can only be imputed to its hierarchical decomposition scheme.

We may now observe the decomposition graphs displayed in Figures A.3 to A.8, and relate these to the performance measures in Table 5.5. Several graphs like

scene, *image*, *yeast* and *emotions* are densely connected, while others are surprisingly sparse, like *genbase* and *medical*. On *scene*, *image*, *yeast* and *emotions* F-LP boils down to the LP method, with a single ILF consisting of all the labels. This is confirmed when inspecting Table 5.5, as F-LP and LP exhibit a similar subset $0/1$ loss on these data sets. The opposite was observed on *genbase* and *medical*, where F-LP merely boils down to the BR method. On these two data sets the performance gap between BR and LP is small, and F-LP performs slightly better than both approaches. This is in nice agreement with our previous experiments with sparse structures (toy problem 2, Figures 5.10a and 5.10b). On the remaining data sets, *slashdot* and *enron*, the ILF decomposition exhibits one dominant label factor consisting of most of the labels, and the remaining labels as singletons. In both cases the decomposition does not seem to benefit to F-LP, which may indicate an erroneous ILF decomposition, due to either numerical problems within ILF-Compo or the Composition property being violated.

On the duplicated data sets, the decomposition graphs clearly exhibit two label factors with the original and duplicated label sets, as was expected. In such a situation the decomposition scheme greatly benefits to F-LP, while LP seems to bear some difficulties to efficiently perform MAP inference. Indeed, in typical MLC problems the effective number of label combinations is rather low by nature (see $DL(\mathcal{D})$ in Table 5.4), while here that number grows quadratically due to our particular duplication scheme. As a consequence, on the largest data sets LP is systematically overtaken by simpler approaches such as BR, RAkEL, CC or ECC, even though these do not yield Bayes-optimal predictions. The other Bayes-optimal approaches PCC and MCC do not seem to suffer as much as LP from this additional complexity, and perform consistently well across all data sets. In this setting F-LP outmatches all the other approaches, and therefore seems particularly well suited to scenarios that exhibit distinct label factors of important size.

Regarding the complexity of ILF-Compo, the running time for learning the decomposition seems loosely related to the dimensionality of the data sets, as reported in Table 5.7. However, we believe that in practice it is much more dependent to the dependency structure between the labels and the features, such as the size of the minimal feature subsets of the labels, which is very specific to each data set. Admittedly, the datasets used in this experiment only contain a small to moderate number of labels (up to 53), and the additional computational burden of ILF-Compo to learn the decomposition may become prohibitive for larger data sets.

Tab. 5.6. Total running time (mean in seconds) of the comparative methods for both training and testing on the original and the duplicated benchmark, over 5x2-CV.

method	emotions	image	scene	yeast	slashdot	genbase	medical	enron
F-LP	4	3	27	122	2010	232	769	895
LP	2	3	3	43	160	8	40	872
PCC	1	3	17	1064	na	na	na	na
MCC	0	15	1	59	865	89	299	1519
BR	1	2	4	4	60	4	9	26
RAkEL	2	6	11	17	331	13	40	145
HOMER	2	3	4	4	46	6	11	24
CC	1	3	4	4	48	6	11	26
ECC	2	12	27	31	525	29	87	229
LEAD	15	10	20	48	440	2305	1316	1825

method	emotions2	image2	scene2	yeast2	slashdot2	genbase2	medical2	enron2
F-LP	6	12	103	146	11 754	2097	7103	5674
LP	68	91	62	2063	7104	93	778	6096
PCC	116	115	1604	na	na	na	na	na
MCC	14	131	145	2164	8253	411	973	7066
BR	2	12	18	21	284	12	33	123
RAkEL	10	53	97	89	1489	56	154	699
HOMER	4	15	35	19	183	14	25	53
CC	4	10	16	14	189	13	36	125
ECC	12	71	140	127	1737	125	659	1110
LEAD	97	110	165	271	5080	1373	2696	5029

Tab. 5.7. Running time to learn the ILF-Compo decomposition graph (mean $\pm$ std in seconds) on the original and the duplicated benchmark, over 5x2-CV.

dataset	original	duplicated
emotions	1 ± 0	3 ± 0
image	1 ± 0	8 ± 1
scene	25 ± 10	94 ± 14
yeast	5 ± 1	26 ± 1
slashdot	1789 ± 1470	$10\,752 \pm 12006$
genbase	119 ± 3	1116 ± 12
medical	438 ± 12	4297 ± 92
enron	396 ± 9	3580 ± 61

5.3 Application to F-measure maximization

In this section, we investigate the ILF decomposition approach for MLC under the F-loss function, a.k.a. F-measure maximization. The F-measure, introduced in Section 4.1.2, is a standard performance metric in information retrieval which is used in a variety of prediction problems including binary classification, multi-label classification and structured output prediction. Formally, the F-measure of a binary vector $\mathbf{h} = (h_1, \ldots, h_m)$ compared to a label vector $\mathbf{y} = (y_1, \ldots, y_m)$ is given by

$$F(\mathbf{y}, \mathbf{h}) = \frac{2(\mathbf{y} \cdot \mathbf{h})}{\mathbf{y} \cdot \mathbf{y} + \mathbf{h} \cdot \mathbf{h}}, \tag{5.2}$$

where $\cdot$ denotes the dot product operator[8] and $0/0 = 1$ by definition.

Optimizing the F-measure is a statistically and computationally challenging problem, since no closed-form solution exists and few theoretical studies of the F-measure were carried out. Some efficient approaches for F-measure maximization have been proposed [Jan07; Ye+12], which explicitly rely on the assumption of conditional independence of the labels (somewhat similarly to BR for subset $0/1$ loss). Under this restricted assumption, only $O(m)$ parameters are required, that is, the marginal probability $p(y_i|\mathbf{x})$ of each label, and inference can be made in $O(m^2)$. While such an assumption naturally holds in standard binary classification (labels are i.i.d.), in domains like MLC it is in general not tenable any more.

GFM

Recently, Dembczynski et al. [Dem+11] presented an exact algorithm for F-measure maximization, named *General F-measure Maximizer* (GFM), which requires $O(m^2)$ parameters and can infer Bayes-optimal predictions in $O(m^3)$. While computationally expensive, this method is statistically consistent and results in state-of-the art performance in multi-label classification [Wae+14]. Here, we will show how our ILF decomposition approach can be applied to the F-measure maximization problem, by reducing the number of parameters required by GFM to $O(m^2/n)$ (in the best case, assuming the label set can be partitioned into n conditionally independent subsets), with an inference complexity contained within $O(m^3)$. In the following we

F-GFM

introduce the *Factorized-GFM* (F-GFM) method [GA16a], and evaluate its empirical performance on a carefully designed set of experiments.

[8] In a binary setting the dot product $\mathbf{h} \cdot \mathbf{y}$ offers a convenient notation to count the number of positives values common to both $\mathbf{h}$ and $\mathbf{y}$.

5.3.1 Factorized GFM

We start by reviewing the General F-measure Maximizer method presented in Dembczynski et al. [Dem+11]. To keep our notation uncluttered, without loss of generality, the conditioning on $\mathbf{X}$ will be made implicit in the remainder of this work, so that $\mathbf{h}(\mathbf{x}) = \mathbf{h}$ and $p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y})$. Assuming the underlying probability distribution p is known, the optimal prediction $\mathbf{h}^*$ that maximizes the expected F-measure is given by

$$\mathbf{h}^* = \underset{\mathbf{h}\in\{0,1\}^m}{\arg\max}\, \mathbb{E}_{\mathbf{y}}[F(\mathbf{y},\mathbf{h})] = \underset{\mathbf{h}\in\{0,1\}^m}{\arg\max} \sum_{\mathbf{y}\in\{0,1\}^m} p(\mathbf{y})F(\mathbf{y},\mathbf{h}). \tag{5.3}$$

Jansche [Jan07] noticed that (5.3) can be solved via outer and inner maximization. The inner maximization step is

$$\mathbf{h}^{(k)} = \underset{\mathbf{h}\in\mathcal{H}_k}{\arg\max}\, \mathbb{E}_{\mathbf{y}}[F(\mathbf{y},\mathbf{h})], \tag{5.4}$$

where $\mathcal{H}_k = \{\mathbf{h} \in \{0,1\}^m | \mathbf{h}\cdot\mathbf{h} = k\}$, followed by an outer maximization

$$\mathbf{h}^* = \underset{\mathbf{h}\in\{\mathbf{h}^{(0)},\ldots,\mathbf{h}^{(m)}\}}{\arg\max}\, \mathbb{E}_{\mathbf{y}}[F(\mathbf{y},\mathbf{h})]. \tag{5.5}$$

The outer maximization (5.5) can be done in linear time by simply checking all $m+1$ possibilities. The main effort is then devoted to solving the inner maximization (5.4). For convenience, Waegeman et al. [Wae+14] introduce the following quantities:

$$s_{\mathbf{y}} = \mathbf{y}\cdot\mathbf{y}, \quad \Delta_{ik} = \sum_{\mathbf{y}\in\mathcal{Y}_i} \frac{2p(\mathbf{y})}{s_{\mathbf{y}}+k},$$

with $\mathcal{Y}_i = \{\mathbf{y} \in \{0,1\}^m | y_i = 1\}$. The first quantity is the number of ones in the label vector $\mathbf{y}$, while Δ_{ik} is a specific marginal value for the i-th label. Using these quantities, the maximizer in (5.4) becomes

$$\mathbf{h}^{(k)} = \underset{\mathbf{h}\in\mathcal{H}_k}{\arg\max} \sum_{i=1}^{m} h_i\Delta_{ik},$$

which boils down to selecting the k labels with the highest Δ_{ik} value. In the special case of $k = 0$, we have $\mathbf{h}^{(0)} = \mathbf{0}$ and $\mathbb{E}_{\mathbf{y}}[F(\mathbf{y},\mathbf{h}^{(0)})] = p(\mathbf{y} = \mathbf{0})$. As a result, it is not required to estimate the 2^m parameters of the whole distribution $p(\mathbf{y})$ to find the F-measure maximizer $\mathbf{h}^*$, but only m^2+1 parameters: the values of Δ_{ik} which take the form of an $m\times m$ matrix $\mathbf{\Delta}$, plus the value of $p(\mathbf{y} = \mathbf{0})$. Once these parameters are known, obtaining the optimal F-measure prediction can be done in $O(m^2)$ with the GFM algorithm (see [Wae+14] for details).

In order to combine GFM with a training algorithm, Waegeman et al. [Wae+14] decompose the $\mathbf{\Delta}$ matrix as follows. Consider the probabilities

$$p_{is} = p(y_i = 1, s_{\mathbf{y}} = s), \quad i, s \in \{1, \ldots, m\}$$

that constitute an $m \times m$ matrix $\mathbf{P}$, along an $m \times m$ matrix $\mathbf{W}$ with elements

$$w_{sk} = \frac{2}{s+k},$$

then it can be easily shown that

$$\mathbf{\Delta} = \mathbf{PW}. \tag{5.6}$$

If the matrix $\mathbf{P}$ is taken as an input by GFM then its computational complexity is dominated by the matrix multiplication (5.6), which is solved naively in $O(m^3)$.

In view of this result, Dembczynski et al. [Dem+11] establish that modeling pairwise or higher degree dependences between labels is not necessary to obtain an optimal solution, only a proper estimation of marginal quantities p_{is} is required to take the number of co-occurring labels into account. With our ILF decomposition approach, we will show that modeling high degree dependences between the labels can help to obtain better estimates of p_{is}, and thereby better predictions within the GFM framework.

F-GFM parameters

Assuming an ILF decomposition of the label set, the p_{is} parameters can be reconstructed from a smaller number of parameters estimated locally within each label factor, at a computational cost of $O(m^3)$.

Let m_k denote the number of labels in a particular label factor, we introduce for every label factor $\mathbf{Y}_{F_k} = \{Y_1, \ldots, Y_{m_k}\}$ the following terms,

$$p^k_{is} = p(y_i = 1, s_{\mathbf{y}_{F_k}} = s), \quad i, s \in \{1, \ldots, m_k\},$$

which constitute an $m_k \times m_k$ matrix $\mathbf{P}^k$.

Given a factorization of the label set into label factors, our proposed method F-GFM requires to estimate, for each label factor, a local matrix $\mathbf{P}^k$ of size ${m_k}^2$, and then combine these to reconstruct the global matrix $\mathbf{P}$ of size m^2. The total number of parameters is therefore reduced from m^2 to $\sum_{k=1}^{n} {m_k}^2$. It is easily shown that, in the best case, the total number of parameters is m^2/n when $m_k = m/n$ for every

label factor, and the worst case is $(n-1)+(m-n+1)^2$ when all the label factors, but one, are singletons. In both cases the number of parameters is reduced, which results in better probability estimates and a better robustness of the model.

Prior to recovering $\mathbf{P}$, we must introduce some extra parameters $\mathbf{d}$. Consider, for each label factor $\mathbf{Y}_{F_k}$, the following probabilities,

$$d_s^k = p(s_{\mathbf{y}_{F_k}} = s), \quad s \in \{0, \ldots, m_k\},$$

which form a vector $\mathbf{d}^k$ of size $m_k + 1$. These do not constitute additional free parameters, as each $\mathbf{d}^k$ vector can be recovered from a $\mathbf{P}^k$ matrix in ${m_k}^2$ operations.

Recovering $\mathbf{d}^k$

Note that the same method holds to recover $\mathbf{d}^k$ from $\mathbf{P}^k$ or $\mathbf{d}$ from $\mathbf{P}$, therefore in the following we will drop the superscript k to keep our notations uncluttered. Consider the following expression for p_{is} and d_s,

$$\begin{aligned} p_{is} &= \sum_{\mathbf{y} \in \{0,1\}^m} p(\mathbf{y}) \cdot \mathbb{I}[s_{\mathbf{y}} = s] \cdot \mathbb{I}[y_i = 1], \\ d_s &= \sum_{\mathbf{y} \in \{0,1\}^m} p(\mathbf{y}) \cdot \mathbb{I}[s_{\mathbf{y}} = s]. \end{aligned}$$

Notice that, for a particular $\mathbf{y} \in \{0,1\}^m$, the following equality holds,

$$\mathbb{I}[s_{\mathbf{y}} = s] \cdot \sum_{i=1}^{m} \mathbb{I}[y_i = 1] = s \cdot \mathbb{I}[s_{\mathbf{y}} = s].$$

Therefore, when $s > 0$, d_s can be expressed as

$$d_s = \sum_{\mathbf{y} \in \{0,1\}^m} p(\mathbf{y}) \cdot \mathbb{I}[s_{\mathbf{y}} = s] \cdot \frac{1}{s} \sum_{i=1}^{m} \mathbb{I}[y_i = 1].$$

This expression can be further simplified in order to express d_s as a composition of p_{is} terms,

$$d_s = \frac{1}{s} \sum_{i=1}^{m} p_{is}, \quad \forall s \in \{1, \ldots, m\}.$$

We may recover d_0 from

$$d_0 = 1 - \sum_{s=1}^{m} d_s.$$

As a result, each vector $\mathbf{d}^k$ can be obtained from $\mathbf{P}^k$ in ${m_k}^2$ operations. Interestingly, because $p(\mathbf{y} = \mathbf{0}) = d_0$, this additional parameter can actually be inferred from $\mathbf{P}$ at the expense of m^2 operations, thereby reducing the number of parameters required by GFM to m^2 instead of $m^2 + 1$.

Recovering $\mathbf{P}$

We may now describe the global procedure to recover $\mathbf{P}$ and $p(\mathbf{y} = \mathbf{0})$ from the individual $\mathbf{P}^k$ matrices, in $O(m^3)$.

When $n = 2$. Let us first assume that there are only two label factors $\mathbf{Y}_{F_1}$ and $\mathbf{Y}_{F_2}$. Consider a label Y_i that belongs to $\mathbf{Y}_{F_1}$, from the marginalization rule p_{is} may be decomposed as follows,

$$p_{is} = \sum_{s'} p(y_i = 1, s_{\mathbf{y}} = s, s_{\mathbf{y}_{F_1}} = s'). \tag{5.7}$$

The inner term of this sum factorizes because of the label factor assumption. First, recall that $s_{\mathbf{y}} = s_{\mathbf{y}_{F_1}} + s_{\mathbf{y}_{F_2}}$, which allows us to write

$$p(y_i = 1, s_{\mathbf{y}} = s, s_{\mathbf{y}_{F_1}} = s') = p(y_i = 1, s_{\mathbf{y}_{F_1}} = s', s_{\mathbf{y}_{F_2}} = s - s').$$

Second, due to the label factor assumption, i.e. $\mathbf{Y}_{F_1} \perp\!\!\!\perp \mathbf{Y}_{F_2}$, we have

$$p(y_i, s_{\mathbf{y}}, s_{\mathbf{y}_{F_1}}) = p(y_i, s_{\mathbf{y}_{F_1}}) \cdot p(s_{\mathbf{y}_{F_2}}). \tag{5.8}$$

We may combine (5.8) and (5.7) to obtain

$$p_{is} = \sum_{s'} p(y_i = 1, s_{\mathbf{y}_{F_1}} = s') \cdot p(s_{\mathbf{y}_{F_2}} = s - s'). \tag{5.9}$$

Finally, we have necessarily $s' \leq s$ and $s' \leq m_1$, which implies $s' \leq min(s, m_1)$. Also, $s - s' \leq m_2$ and $s' \geq 1$ because $y_i = 1$, which implies $s' \geq max(1, s - m_2)$. So we can re-write (5.9) as follows,

$$p_{is} = \sum_{s'=max(1,s-m_2)}^{min(s,m_1)} p^1_{is'} \cdot d^2_{s-s'}. \tag{5.10}$$

In the case where $Y_i \in \mathbf{Y}_{F_2}$, we obtain a similar result. In the end, given that both $\mathbf{P}^k$ and $\mathbf{d}^k$ are known for $\mathbf{Y}_{F_1}$ and $\mathbf{Y}_{F_2}$, (5.10) allows us to recover all term in $\mathbf{P}$ in $(m_2 + 1){m_1}^2 + (m_1 + 1){m_2}^2$ operations. Assuming that only the $\mathbf{P}^k$ matrices are

known, we must add up the additional cost for recovering the $\mathbf{d}^k$ vectors, which brings the total computational burden to $(m_2+2){m_1}^2+(m_1+2){m_2}^2$.

For any n. The same procedure can be used iteratively to merge $\mathbf{P}^1$ and $\mathbf{P}^2$ into a matrix $\mathbf{P}'$ of size $(m_1+m_2)^2$, then combine this matrix with $\mathbf{P}^3$ to form a new matrix of size $(m_1+m_2+m_3)^2$, and so on until every label factor is merged into a matrix of size m^2. In the end we obtain $\mathbf{P}$ in a total number of operations equal to

$$\sum_{i=2}^{n}(m_i+2)(\sum_{j=1}^{i-1}m_j)^2+m_i^2(2+\sum_{j=1}^{i-1}m_j).$$

To avoid tedious calculations, we can easily compute a tight upper bound of the number of computations, i.e.

$$\max_{m_1,\ldots,m_n}\quad\sum_{i=2}^{n}(m_i+2)\left(\sum_{j=1}^{i-1}(m_j+2)\right)\left(\sum_{j=1}^{i}(m_j+2)\right)\quad\text{s.t.}\sum_{i=1}^{n}m_i=m.$$

Solving $\nabla\mathcal{L}(m_1,\ldots,m_n,\lambda)=0$ yields

$$m_i=\Big((m+2n)^2-\lambda\Big)^{1/2}+2n,\quad\forall i\in\{1,\ldots,n\},$$

which implies that all the label factors have equal size. As a result, with $m_i=m/n$ for every label factor we obtain an upper bound on the worst case number of operations equal to $(\frac{m}{n}+2)^3(n^2-1)$. Thus, the overall complexity to recover $\mathbf{P}$ is bounded by $O(m^3)$.

Given that the label factors are known and that every $\mathbf{P}^k$ matrix has been estimated, the whole F-GFM procedure for recovering $\mathbf{P}$ and inferring a Bayes-optimal prediction is presented in Algorithm 14, with an overall complexity within $O(m^3)$, just as GFM.

Parameter estimation

Our proposed method F-GFM requires to estimate for each label factor $\mathbf{Y}_{F_k}$ the $m_k\times m_k$ matrix $\mathbf{P}^k$, instead of the whole $m\times m$ matrix $\mathbf{P}$ in GFM. Still, the problem of parameter estimation in GFM and F-GFM is essentially the same, that is, estimating the matrix $\mathbf{P}$ (resp. $\mathbf{P}^k$) for a particular input $\mathbf{x}$, given a set of training samples $(\mathbf{x},\mathbf{y})$ (resp. $(\mathbf{x},\mathbf{y}_{F_k})$).

Dembczynski et al. [Dem+13] propose a solution to estimate the p_{is} terms directly by solving m multinomial logistic regression problems with $m+1$ classes, with one

Algorithm 14 Factorized-GFM

Require: $\mathbf{Y}$ the label set, $\mathbf{Y}_{F_1}, \ldots, \mathbf{Y}_{F_n}$ the label factors, $m_1, \ldots, m_n$ their size and $\mathbf{P}^1, \ldots, \mathbf{P}^n$ their matrix of $p^k_{i,s}$ parameters.
Ensure: $\mathbf{h}^*$ the F-measure maximizing prediction.

Initialize $m \leftarrow 0$, $\mathbf{P} \leftarrow \emptyset$, $\mathbf{d} \leftarrow \{1\}$
for all $k \in \{1, \ldots, n\}$ **do**
  $m' \leftarrow m$, $\mathbf{P}' \leftarrow \mathbf{P}$, $\mathbf{d}' \leftarrow \mathbf{d}$, $m \leftarrow m' + m_k$
  Initialize $\mathbf{d}^k = \{d_0, \ldots, d_{m_k}\}$ a vector of size $m_k + 1$
  for all $s \in \{1, \ldots, m_k\}$ **do** ▷ 1) recover $\mathbf{d}^k$ from $\mathbf{P}^k$
    $d^k_s \leftarrow s^{-1} \sum_{i=1}^{m_k} p^k_{i,s}$
  $d^k_0 \leftarrow 1 - \sum_{s=1}^{m_k} d^k_s$
  Initialize $\mathbf{P}$ a zero matrix of size $m \times m$
  for all $i \in \{1, \ldots, m_k\}$ **do** ▷ 2) merge $\mathbf{P}^k$ and $\mathbf{d}'$ into $\mathbf{P}$
    for all $s1 \in \{1, \ldots, m_k\}$ **do**
      for all $s2 \in \{0, \ldots, m'\}$ **do**
        $p_{i,s1+s2} \leftarrow p_{i,s1+s2} + p^k_{i,s1} \cdot d'_{s2}$
  for all $i \in \{1, \ldots, m'\}$ **do** ▷ 3) merge $\mathbf{P}'$ and $\mathbf{d}^k$ into $\mathbf{P}$
    for all $s1 \in \{1, \ldots, m'\}$ **do**
      for all $s2 \in \{0, \ldots, m_k\}$ **do**
        $p_{i+m_k,s1+s2} \leftarrow p_{i+m_k,s1+s2} + p'_{i,s1} \cdot d^k_{s2}$
  Initialize $\mathbf{d}$ a zero vector of size $m + 1$
  for all $s \in \{1, \ldots, m\}$ **do** ▷ 4) recover $\mathbf{d}$ from $\mathbf{P}$
    $d_s \leftarrow s^{-1} \sum_{i=1}^{m} p_{i,s}$
  $d_0 \leftarrow 1 - \sum_{s=1}^{m} d_s$
$\mathbf{h}^* \leftarrow GFM(\mathbf{P}, d_0)$ ▷ 5) obtain $\mathbf{h}^*$ from $\mathbf{P}$ and d_0
Rearrange $\mathbf{h}^*$ to match the order of the labels in $\mathbf{Y}$.

mapping $(\mathbf{x}, \mathbf{y}) \rightarrow (\mathbf{x}, y = y_i \cdot s_{\mathbf{y}})$ for each label. However, we observed that the parameters estimated with this approach are inconsistent, that is, they often result in a negative probability for d_0 when trying to recover $\mathbf{d}$ from $\mathbf{P}$. To overcome this numerical problem, we found a straightforward and effective approach. Instead of estimating the p_{is} terms directly, we can proceed in two steps. From the chain rule of probabilities, we have that

$$\overbrace{p(y_i, s_{\mathbf{y}}|\mathbf{x})}^{p_{is}} = \overbrace{p(s_{\mathbf{y}}|\mathbf{x})}^{d_s} \cdot \overbrace{p(y_i|s_{\mathbf{y}}, \mathbf{x})}^{p_{i|s}}. \tag{5.11}$$

The idea is to estimate each of these two terms independently. First, the d_s terms are obtained from a multinomial logistic regression with $m + 1$ classes, using the mapping $(\mathbf{x}, \mathbf{y}) \rightarrow (\mathbf{x}, y = s_{\mathbf{y}})$. Second, for each label Y_i we estimate the $p_{i|s}$ terms with a binary logistic regression model, using the mapping $(\mathbf{x}, \mathbf{y}) \rightarrow ((\mathbf{x}, s_{\mathbf{y}}), y = y_i)$. To summarize, for each label factor, one multinomial logistic regression model with $m_k + 1$ classes, and m_k binary logistic regression models are trained. In order to estimate the p^k_{is} terms, we combine the outputs of the multinomial and the binary models according to (5.11). This approach has the desirable advantage of producing calibrated $\mathbf{P}^k$ matrices and consistent $\mathbf{d}^k$ vectors, which appears to be crucial for the success of F-GFM. Notice that in our experiments this approach was also beneficial to GFM in terms of MLC performance.

5.3.2 Toy problem

In this section, we compare GFM and F-GFM on a synthetic toy problem to assess the effective improvement in classification performance due to the label factorization. The code to reproduce this experiment is available online[9].

Setup details

Consider $\mathbf{Y} = \{Y_1, \ldots, Y_8\}$ 8 labels and $\mathbf{X} = \{X_1, \ldots, X_6\}$ 6 binary random variables. The true joint distribution $p(\mathbf{x}, \mathbf{y})$ is encoded in a Bayesian network (one example is displayed in Figure 5.11) which imposes different label factor decompositions and serves as a data-generative model. In this structure, each of the features X_1, X_2, X_3, X_4 is a parent to every label, which allows for a relationship between $\mathbf{X}$ and $\mathbf{Y}$, each label factor $\mathbf{Y}_{F_k}$ is made fully connected by placing an edge $Y_i \rightarrow Y_j$ for every $Y_i, Y_j \in \mathbf{Y}_{F_k}$, $i < j$. The remaining features X_5 and X_6 are totally disconnected from the labels, and serve as irrelevant features. We consider 4 distinct structures with the following ILF decompositions:

[9] https://github.com/gasse/fgfm-toy

- DAG 1: $\mathcal{F}_I = \{\{Y_1, Y_2\}, \{Y_3, Y_4\}, \{Y_5, Y_6\}, \{Y_7, Y_8\}\}$;
- DAG 2: $\mathcal{F}_I = \{\{Y_1, Y_2, Y_3, Y_4\}, \{Y_5, Y_6, Y_7, Y_8\}\}$;
- DAG 3: $\mathcal{F}_I = \{\{Y_1, Y_2, Y_3, Y_4, Y_5, Y_6\}, \{Y_7, Y_8\}\}$;
- DAG 4: $\mathcal{F}_I = \{\{Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7, Y_8\}\}$.

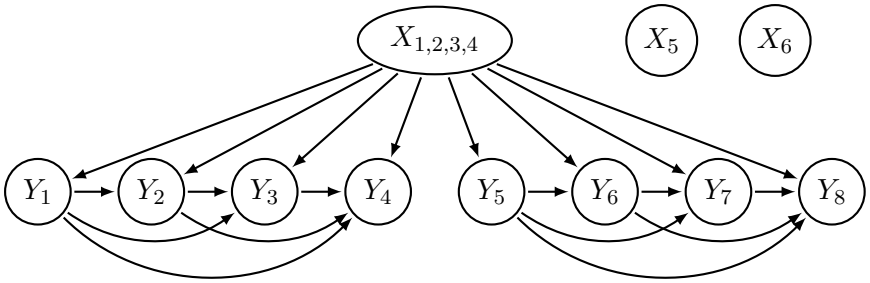


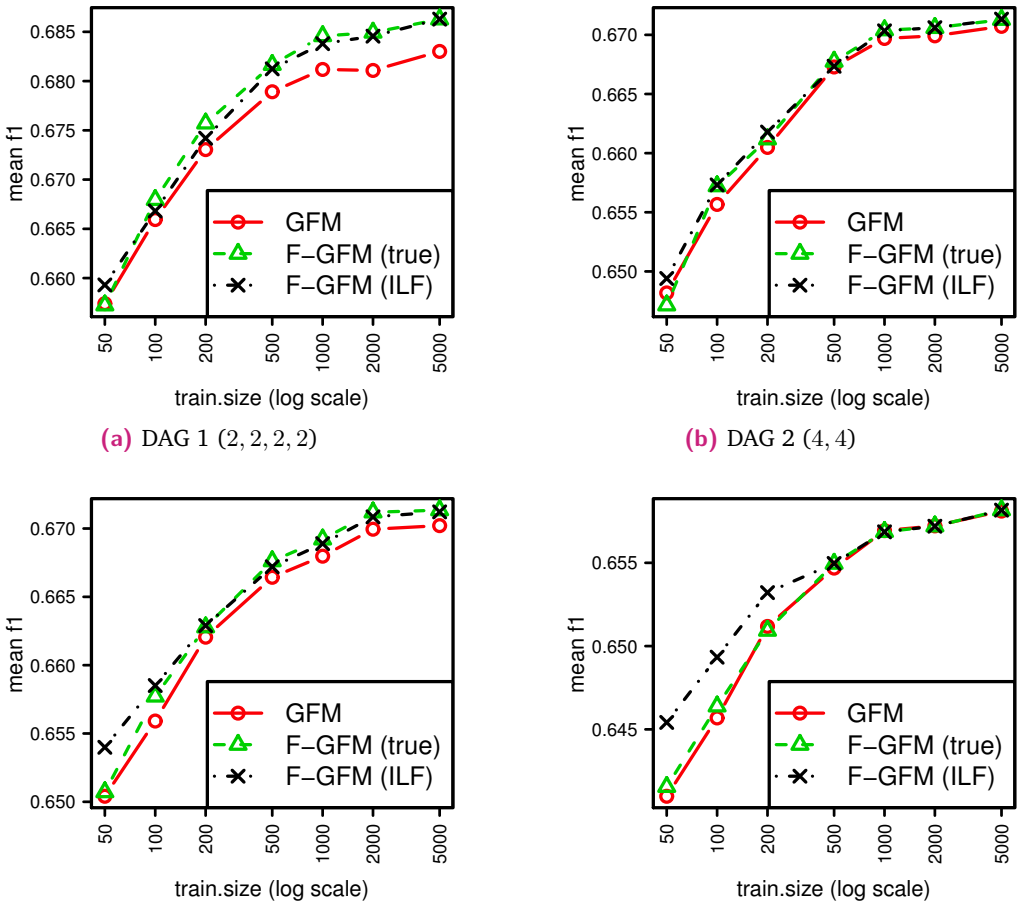
Fig. 5.11. BN structure of our toy problem with DAG 2, i.e. two label factors $\{Y_1, Y_2, Y_3, Y_4\}$ and $\{Y_5, Y_6, Y_7, Y_8\}$. Note that nodes X_1, X_2, X_3 and X_4 are grouped up for readability.

Once these BN structures are fixed, the next step is to generate random distributions $p(\mathbf{x}, \mathbf{y})$ to sample from. For each BN structure we generate a probability distribution by sampling uniformly the conditional probability table of each node from a unit simplex, as discussed in Smith and Tromble [ST04]. We consider 100 such random distributions, and each time we generate 7 training data sets with 50, 100, 200, 500, 1000, 2000 and 5000 samples, and one test set with 5000 samples.

In order to assess separately the influence of the F-GFM procedure and ILF-Compo, we run F-GFM first by using true ILF decomposition of the scenario, and then by using the decomposition obtained with ILF-Compo from the training data. Within ILF-Compo we employ a significance level $\alpha = 0.01$, and to estimate the F-GFM parameters we use the standard multinomial logistic regression model from the *nnet*[VR02] R package, with weight decay regularization and λ chosen over a 3-fold cross validation.

Results

In Figure 5.12 we report the test set F-measure obtained by GFM and F-GFM with the true decomposition (true) and the learned decomposition (learn), with respect to the training size, for each scenario, averaged over the 100 repetitions. As expected, the more data available for training, the more accurate the parameter estimates, and thus the better the F-measure on the test set. F-GFM based on ILF-compo outperforms the original GFM method, sometimes by a significant margin (see Fig.5.12c and 5.12d with small sample sizes). Interestingly, F-GFM based on the learned ILF decomposition performs not only better than GFM, but also better than F-GFM based on the true ILFs, especially in the last scenario with a single ILF of size



(a) DAG 1 $(2, 2, 2, 2)$ (b) DAG 2 $(4, 4)$

(c) DAG 3 $(6, 2)$ (d) DAG 4 (8)

Fig. 5.12. Mean F-measure of GFM and F-GFM on each DAG, with 7 different training sizes (50, 100, 200, 500, 1000, 2000, 5000) displayed on a logarithmic scale, averaged over 100 repetitions with random distributions. F-GFM (true) uses the true decomposition, while F-GFM (ILF) uses the decomposition learned with ILF-Compo from the training data.

Tab. 5.8. F-measure (mean $\pm$ std in percent) achieved by F-GFM and GFM on the original benchmark, over 5x2-CV. Best results are bold-faced (higher is better).

method	emotions	image	scene	yeast	slashdot	genbase	medical	enron
F-GFM	**64.9** $\pm$ **1.1**	**57.7** $\pm$ **1.0**	74.8 $\pm$ 0.9	**64.8** $\pm$ **0.4**	56.6 $\pm$ 1.2	98.5 $\pm$ 0.6	75.7 $\pm$ 1.9	**59.1** $\pm$ **0.8**
GFM	**64.9** $\pm$ **1.1**	**57.7** $\pm$ **1.0**	**75.0** $\pm$ **0.5**	**64.8** $\pm$ **0.4**	**58.3** $\pm$ **0.6**	**98.6** $\pm$ **0.7**	**81.7** $\pm$ **1.5**	56.6 $\pm$ 0.6

8 with small sample sizes. The reason is that the independence relations learned by ILF-Compo are actually observed in the small training sets while being false in the true distribution. As this false decomposition is found almost valid in small sample sizes — at least from a numerical point of view — the restricted parameter space acts as a regularizer which turns out to be beneficial to F-GFM. This is not surprising as Binary Relevance is sometimes shown to outperform other sophisticated MLC techniques exploiting the label correlations when training data are insufficient (see [Lua+12], or Figure 5.10b in our previous experiment), while being based on wrong independence assumptions. The same remark holds for the Naive Bayes model in standard multi-class learning tasks, which wrongly assumes the features to be independent given the output. Overall, in this experiment F-GFM with the learned ILF decomposition behaves usually as good or better than F-GFM based on the ground truth ILF decomposition, which assesses the effectiveness of ILF-Compo and corroborates our theoretical results obtained in Section 5.1.

5.3.3 Real-world benchmark

We now report on an experiment performed on 8 real-world multi-label data sets, that we previously introduced in Section 5.2.3 (see Table 5.4 for details). Here again we employ IFL-Compo with significance level $\alpha = 0.0001$ to learn the ILF decomposition. We report in Table 5.8 the performance of GFM and F-GFM in terms of test set F-measure, averaged over a 5x2-fold cross-validation for each data set. On the four data sets *emotions*, *image*, *scene* and *yeast*, ILF-Compo does not exhibit an ILF decomposition, therefore F-GFM resumes to GFM and shows similar performance. On the remaining data sets, the empirical results are not very convincing, as F-GFM is shown to outperform GFM only on the *enron* data set. On *genbase* it performs comparably with GFM, while on *slashdot* and *medical* it does significantly worse. A potential explanation for this weak performance is that these three data sets exhibit many singleton ILFs, that is, many label factors with a single label. We observed that in such a situation F-GFM (Algorithm 14) ends up multiplying many probabilities of different magnitudes, which can lead to inconsistent estimates of the $\mathbf{P}$ matrix due to a snowball effect. We believe that this issue could be alleviated in practice, by implementing F-GFM more efficiently under careful numerical considerations.

5.4 Discussion

In this chapter, we introduced the concept of irreducible label factors (ILFs), that is, the decomposition of a conditional probability distribution $p(\mathbf{y}|\mathbf{x})$ into minimal disjoint marginal distributions, in order to simplify the multi-label classification problem. We showed that a correct constraint-based procedure exists to identify the ILFs in the general case with only $O(m^2)$ pairwise tests of conditional independence, and derived an efficient procedure under the Composition assumption, ILF-Compo. In a series of synthetic and benchmark experiments, we applied our ILF decomposition approach to the MLC problem under both the subset $0/1$ loss and the F-loss, two popular loss functions in the MLC literature.

For subset $0/1$ loss minimization, the ILF decomposition allows for a drastic reduction of the number of parameters to be extracted from the joint distribution $p(\mathbf{y}|\mathbf{x})$, from $O(2^m)$ to $O(n2^{\frac{m}{n}})$ (assuming n ILFs of equal size). Due to this parameter reduction, one can obtain stronger probability estimates and in the end more accurate Bayes-optimal predictions, as was shown in our experiments. Regarding inference complexity, due to the ILF decomposition it reduces from $O(2^m)$ to $O(n2^{\frac{m}{n}})$, which is also much desirable.

For F-measure maximization, the ILF decomposition also allows for a significant parameter reduction, from $O(m^2)$ to $O(\frac{m^2}{n})$ (again, assuming n ILFs of equal size), and thereby stronger probability estimates. Regarding inference complexity, in the context of the F-measure we were only able to show that it remains within $O(m^3)$, that is, no higher than inference complexity without ILF decomposition. Interestingly, Ye et al. [Ye+12] show that in the context of a fully factorized distribution (i.e., all ILFs are singletons) inference complexity under the F-loss can be reduced to $O(m^2)$. Therefore, we believe that for intermediate decompositions the inference complexity may lie in-between $O(m^2)$ and $O(m^3)$, though we were not able to derive such a procedure.

While we focused here on the subset $0/1$ loss and F-loss, we would like to point out that the ILF decomposition of $p(\mathbf{y}|\mathbf{x})$ may also help minimize other loss functions for which the Bayes-optimal solution is unknown or too expensive to compute, such as the Jaccard loss. As discussed in Section 4.2.5, Monte Carlo methods can be used to derive approximate Bayes-optimal predictions, by minimizing the empirical loss expectation over s independent samples $\{\mathbf{y}^{(i)}\}_{i=1}^{s}$ drawn from the estimated conditional distribution $p(\mathbf{y}|\mathbf{x})$, i.e.,

$$\mathbf{h}^{\star}(\mathbf{x}) = \arg\min_{\hat{\mathbf{y}}} \sum_{i=1}^{s} L(\hat{\mathbf{y}}, \mathbf{y}^{(i)}).$$

Assuming an ILF decomposition, this can be achieved by drawing a sub-sample in each label factor $\mathbf{Y}_F$ from the conditional joint distribution $p(\mathbf{y}_F|\mathbf{x})$ and then concatenating them all to form a complete observation. From this point of view, the ILF decomposition can help to solve the MLC problem more efficiently under any loss function.

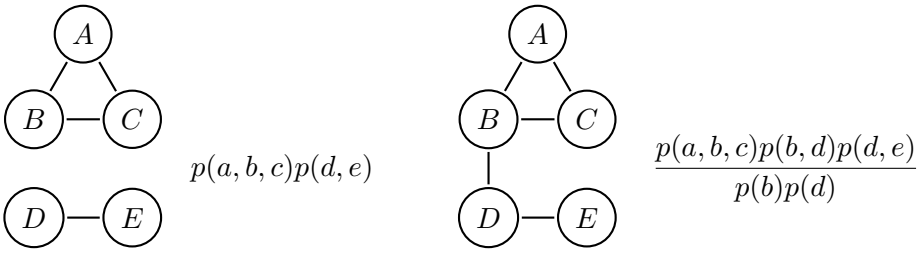
Admittedly, the ILF approach has one major drawback: it relies on the existence of an ILF decomposition of $p(\mathbf{y}|\mathbf{x})$. From Figures A.1 to A.8 we can see that this does not happen very often in real-world problems. Still, there may be some particular situations (in MLC and more broadly multi-variate supervised learning) where such a decomposition is inherent to the problem, in which case our ILF learning procedures may prove particularly useful. Also, we have shown that the F-GFM procedure proposed in Section 5.3.1 could perform exact F-measure maximization when given an ILF decomposition, but we did not investigate on its empirical performance when given an arbitrary decomposition. It may be that such an approximate F-measure maximization procedure could provide a useful alternative when the number of parameters to estimate within GFM becomes too prohibitive.

Conclusion and perspectives

In this thesis, we addressed the specific problem of probabilistic graphical model structure learning, that is, finding the most efficient structure to represent a probability distribution, given only a sample set $\mathcal{D} \sim p(\mathbf{v})$. Our main contributions are 1) a new hybrid algorithm for Bayesian network structure learning, H2PC, presented in Chapter 3, which achieves state-of-the-art performance; and 2) a new generic approach to the multi-label classification problem based on the identification of the irreducible label factors (ILFs) of the conditional distribution of the labels, with a series of quadratic constraint-based characterizations presented in Chapter 5.

The ILF approach is theoretically very appealing, as it breaks down the label set $\mathbf{Y}$ into a unique partition of irreducible and conditionally independent components. This decomposition then allows for a robust learning of $p(\mathbf{y}|\mathbf{x})$ due to the drastic reduction in the number of parameters to be estimated, and an efficient MAP inference due to its natural decomposition into a series of independent and smaller sub-problems. Still, the ILF approach has one inherent major drawback: an ILF decomposition must exist. We discuss below some research perspectives regarding this limitation.

First, we would like to point out that the ILF concept can also be found in *Sum-Product Network* (SPN) models, discussed in Section 4.3.5. Indeed, ILF learning is an essential task in current *Sum-Product Network* (SPN) structure learning algorithms [GD13], as the splitting criterion of a product node is by definition a factorization of the current distribution into disjoint marginal distributions. When located at the top-level of the SPN, a product node corresponds to a decomposition of $p(\mathbf{v})$ into (irreducible) disjoint factors, while when located at a lower level, it corresponds to a decomposition of $p(\mathbf{v}|\mathbf{h})$ with $\mathbf{H}$ some hidden variable, that is, a contextual decomposition. In this sense SPNs push the concept of ILFs one step further, by allowing for both global and contextual decompositions. Interestingly, the procedure proposed in [GD13] to "*partition* $\mathbf{V}$ *into approximately independent subsets* $\mathbf{V}_j$" is actually a direct instantiation of Theorem 5.7 with $\mathbf{Y} = \mathbf{V}$ and $\mathbf{X} = \emptyset$, and therefore is provably correct and optimal (i.e., it yields independent and irreducible subsets)



(a) Disjoint maximal cliques. **(b)** Non-disjoint maximal cliques.

Fig. 5.1. Two undirected chordal graphs, i.e., decomposable models.

when p supports the Composition property. Therefore, we believe that the global problem of SPN structure learning may be an interesting follow-up to our work. A direct question could be: which mixture model $p(\mathbf{v}) = \sum_{\mathbf{h}} p(\mathbf{h})p(\mathbf{v}|\mathbf{h})$ allows for a nice ILF decomposition of each $p(\mathbf{v}|\mathbf{h})$?

Second, in this thesis we did not consider biased ILF decompositions, that is, imposing an arbitrary decomposition when none exists. Under this relaxation one can obtain biased but simple models of $p(\mathbf{y}|\mathbf{x})$, which can be preferable to correct but complex ones, due to their computational tractability and their relative immunity to over-fitting. Such an approach is quite popular when learning decomposable models from data, e.g., tree models [CL68] or low-treewidth models [BJ01]. In the context of ILF decompositions, the problem of learning the least biased model may then formulate as: which decomposition violates the least the dependencies in the data, with ILFs of maximum size 6?

Finally, we notice that the disjoint factorization of ILFs bears a close resemblance to the non-disjoint factorization of decomposable models[1]. For simplicity, consider $\mathbf{Y} = \mathbf{V}$ and $\mathbf{X} = \emptyset$. An ILF factorization is expressed as a product of disjoint marginal distributions,

$$p(\mathbf{v}) = \prod_{\mathbf{V}_F \in \mathcal{F}} p(\mathbf{v}_F), \tag{5.1}$$

where $\mathcal{F}$ is a partition of $\mathbf{V}$. On the other hand, the factorization of a decomposable model is expressed as a ratio of products of non-disjoint marginal distributions,

$$p(\mathbf{v}) = \frac{\prod_{\mathbf{V}_F \in \mathcal{F}} p(\mathbf{v}_F)}{\prod_{\mathbf{V}_I \in \mathcal{I}} p(\mathbf{v}_I)}, \tag{5.2}$$

where $\mathcal{F}$ is a cover of $\mathbf{V}$, and $\mathcal{I}$ contains different intersects between elements of $\mathcal{F}$. Equations (5.1) and (5.2) are best represented by undirected chordal graphs, as in Figure 5.1. In such a structure, $p(\mathbf{v})$ is expressed as the product of the marginal distribution of each maximal clique in the graph, divided by the product of marginal distributions of clique intersections. Clearly, decomposable models are

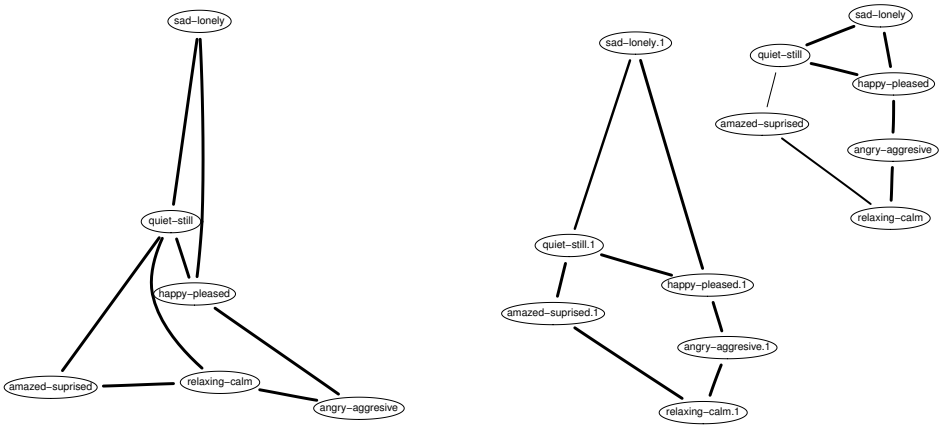
[1] Decomposable models are briefly discussed in Section 2.1.2.

more expressive than ILF models, while retaining many interesting properties such as a local estimation of each marginal distribution, and MAP inference complexity bounded by the largest clique size. An interesting follow-up to our work would be to extend our theoretical results on ILFs to the broader class of decomposable models. To our knowledge, the characterization of decomposable models with CI tests remains an open problem [Stu05][§9.2, Direction 2].

Supplementary material

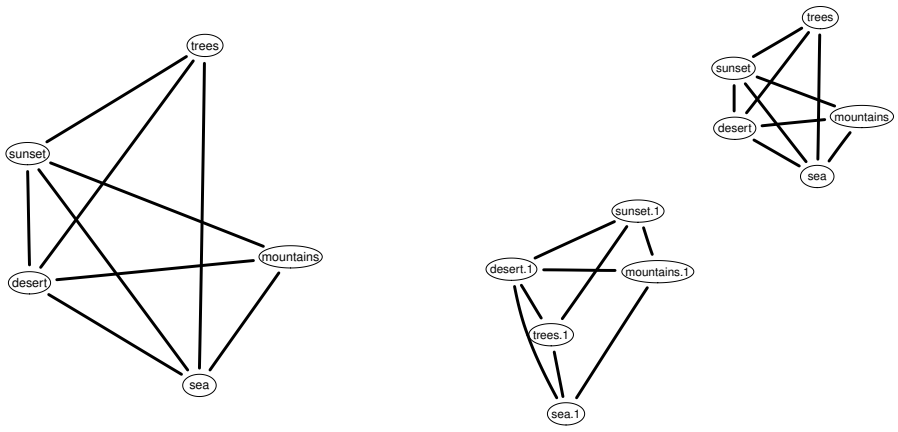
A

A.1 Decomposition graphs



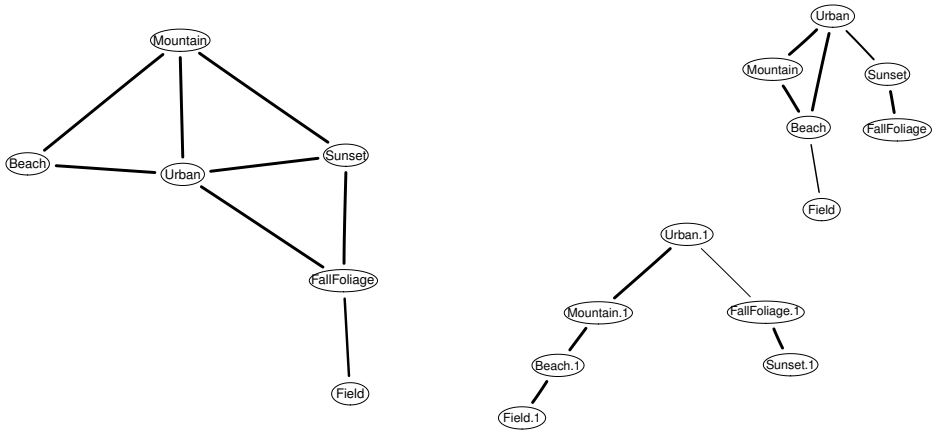
(a) emotions **(b)** emotions2

Fig. A.1. Typical ILF-Compo decomposition graphs for *emotions* and *emotions2*.



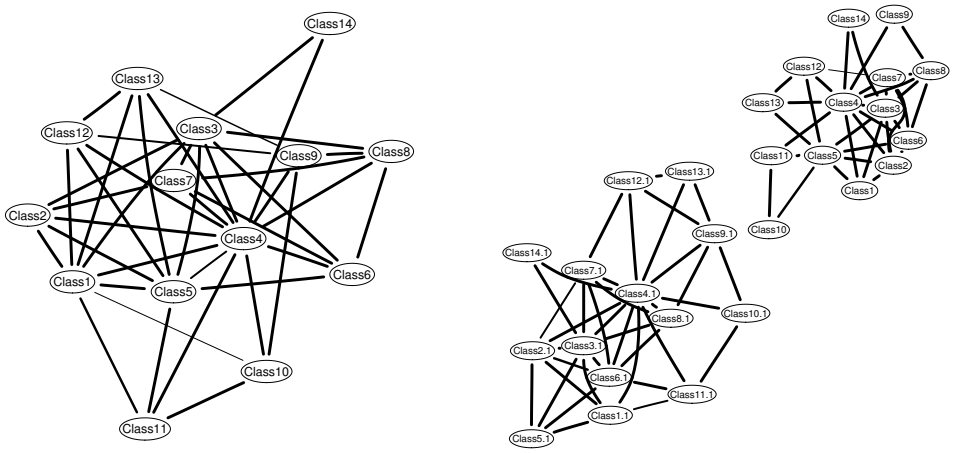
(a) image **(b)** image2

Fig. A.2. Typical ILF-Compo decomposition graphs for *image* and *image2*.



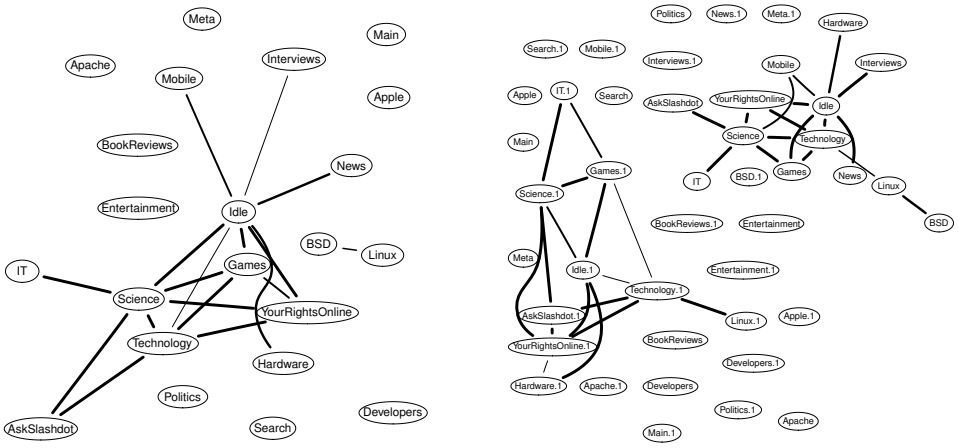
(a) scene **(b)** scene2

Fig. A.3. Typical ILF-Compo decomposition graphs for *scene* and *scene2*.



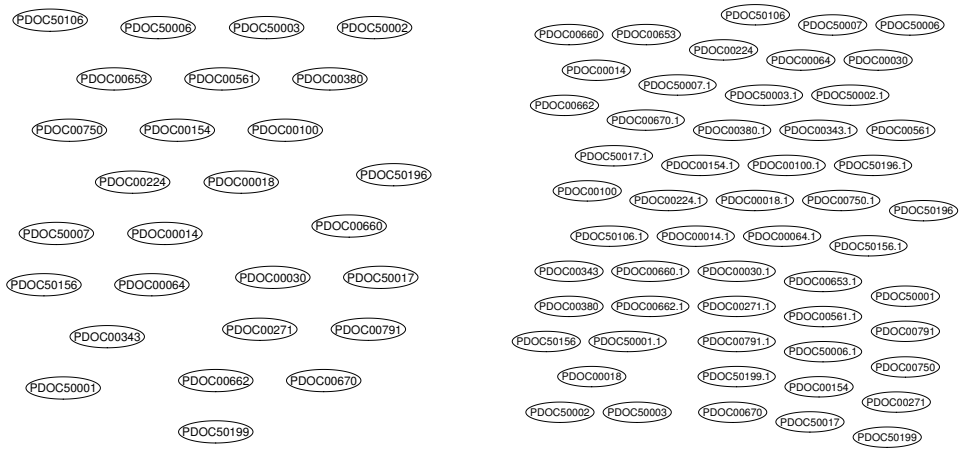
(a) yeast **(b)** yeast2

Fig. A.4. Typical ILF-Compo decomposition graphs for *yeast* and *yeast2*.



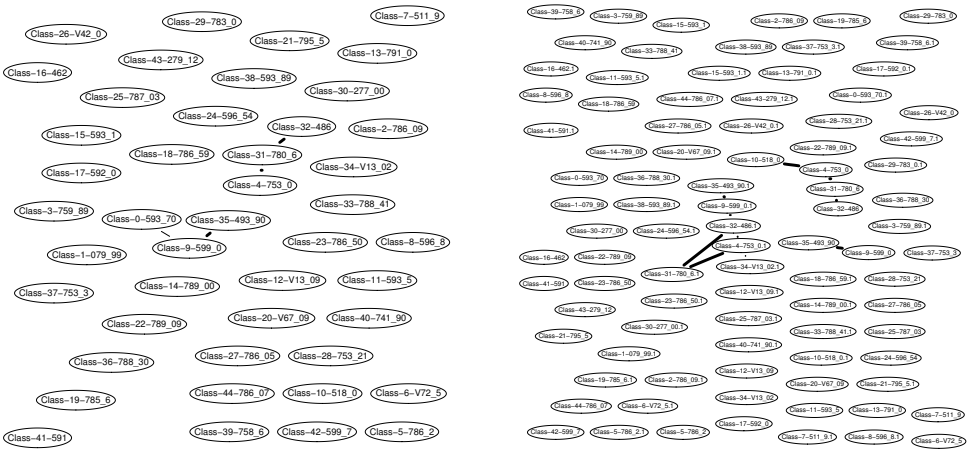
(a) slashdot **(b)** slashdot2

Fig. A.5. Typical ILF-Compo decomposition graphs for *slashdot* and *slashdot2*.



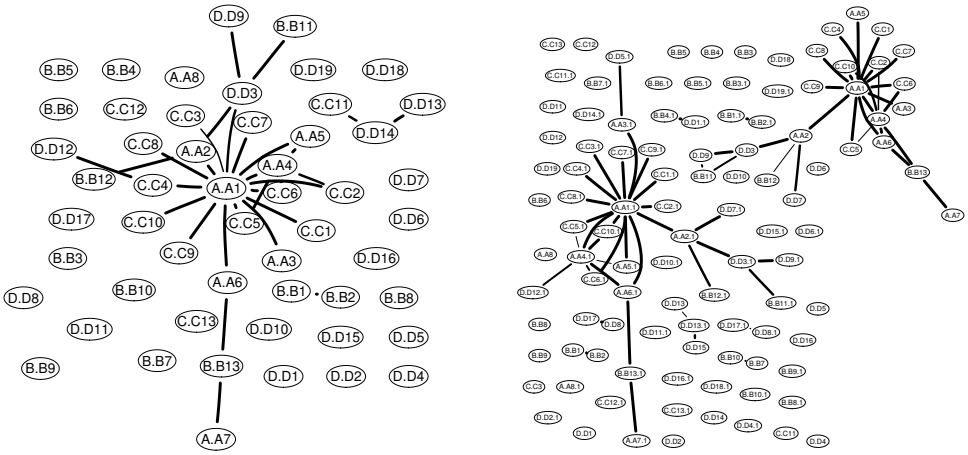
(a) genbase **(b)** genbase2

Fig. A.6. Typical ILF-Compo decomposition graphs for *genbase* and *genbase2*.



(a) medical **(b)** medical2

Fig. A.7. Typical ILF-Compo decomposition graphs for *medical* and *medical2*.



(a) enron **(b)** enron2

Fig. A.8. Typical ILF-Compo decomposition graphs for *enron* and *enron2*.

A.2 Additional benchmark measures

Tab. A.1. Hamming loss (mean ± std in percent) achieved by comparative methods on the original and the duplicated benchmark, over 5x2-CV. Best results are bold-faced (lower is better).

method	emotions	image	scene	yeast	slashdot	genbase	medical	enron
F-LP	20.5 ± 1.4	19.7 ± 0.5	9.4 ± 0.3	21.0 ± 0.2	4.7 ± 0.1	**0.1 ± 0.0**	**1.0 ± 0.0**	5.6 ± 0.1
LP	20.5 ± 1.4	19.7 ± 0.5	**9.3 ± 0.3**	21.0 ± 0.2	4.9 ± 0.1	0.2 ± 0.1	1.2 ± 0.1	5.7 ± 0.1
PCC	21.5 ± 0.8	22.2 ± 1.0	11.6 ± 0.3	21.2 ± 0.4	na	na	na	na
MCC	20.6 ± 1.4	20.8 ± 0.6	11.1 ± 0.5	21.5 ± 0.6	5.1 ± 0.1	**0.1 ± 0.0**	1.1 ± 0.0	6.0 ± 0.2
BR	20.2 ± 0.5	19.9 ± 0.3	11.0 ± 0.2	**20.1 ± 0.3**	4.9 ± 0.1	**0.1 ± 0.0**	1.1 ± 0.0	6.0 ± 0.1
RAkEL	**19.4 ± 0.7**	**19.1 ± 0.4**	9.5 ± 0.3	**20.1 ± 0.2**	4.8 ± 0.1	**0.1 ± 0.0**	1.1 ± 0.0	5.8 ± 0.1
HOMER	20.9 ± 0.8	21.7 ± 0.5	12.3 ± 0.9	24.0 ± 0.7	5.6 ± 0.1	**0.1 ± 0.1**	1.5 ± 0.2	6.4 ± 0.1
CC	21.8 ± 1.2	20.8 ± 0.6	11.1 ± 0.5	21.4 ± 0.5	5.2 ± 0.1	**0.1 ± 0.1**	**1.0 ± 0.1**	6.0 ± 0.1
ECC	20.5 ± 0.9	19.9 ± 0.5	9.9 ± 0.4	20.4 ± 0.3	4.5 ± 0.1	**0.1 ± 0.1**	**1.0 ± 0.1**	5.4 ± 0.1
LEAD	21.2 ± 1.0	19.4 ± 0.4	10.7 ± 0.4	**20.1 ± 0.3**	**4.1 ± 0.0**	0.2 ± 0.1	1.1 ± 0.0	**5.1 ± 0.1**

method	emotions2	image2	scene2	yeast2	slashdot2	genbase2	medical2	enron2
F-LP	23.5 ± 0.7	21.3 ± 0.5	**10.6 ± 0.2**	23.1 ± 0.3	4.8 ± 0.1	**0.1 ± 0.0**	**1.2 ± 0.0**	5.9 ± 0.1
LP	27.4 ± 0.8	25.2 ± 0.5	11.7 ± 0.3	24.6 ± 0.4	6.5 ± 0.1	1.2 ± 0.1	2.6 ± 0.0	7.0 ± 0.1
PCC	22.9 ± 0.6	22.9 ± 0.6	13.2 ± 0.1	na	na	na	na	na
MCC	23.3 ± 0.8	22.9 ± 0.4	12.6 ± 0.2	22.4 ± 0.4	5.0 ± 0.0	**0.1 ± 0.0**	**1.2 ± 0.0**	5.7 ± 0.0
BR	22.3 ± 0.5	20.3 ± 0.3	12.7 ± 0.1	21.0 ± 0.2	4.8 ± 0.0	**0.1 ± 0.0**	1.3 ± 0.1	5.7 ± 0.1
RAkEL	**21.8 ± 0.5**	**20.1 ± 0.4**	11.1 ± 0.2	21.2 ± 0.2	4.7 ± 0.1	**0.1 ± 0.0**	1.3 ± 0.1	5.6 ± 0.1
HOMER	24.4 ± 0.5	24.3 ± 0.6	13.6 ± 0.3	25.2 ± 0.3	6.6 ± 0.2	0.8 ± 0.1	2.0 ± 0.1	7.0 ± 0.4
CC	24.5 ± 0.3	21.9 ± 0.4	12.8 ± 0.3	23.0 ± 0.4	5.0 ± 0.0	**0.1 ± 0.0**	**1.2 ± 0.1**	5.7 ± 0.0
ECC	22.5 ± 0.7	20.6 ± 0.4	11.1 ± 0.3	21.9 ± 0.3	4.4 ± 0.0	0.2 ± 0.0	**1.2 ± 0.1**	5.4 ± 0.1
LEAD	22.2 ± 0.6	20.5 ± 0.3	12.0 ± 0.2	**20.5 ± 0.2**	**4.2 ± 0.0**	0.2 ± 0.0	**1.2 ± 0.1**	**5.2 ± 0.1**

Tab. A.2. Micro-F_1 score (mean $\pm$ std in percent) achieved by comparative methods on the original and the duplicated benchmark, over 5x2-CV. Best results are bold-faced (higher is better).

method	emotions	image	scene	yeast	slashdot	genbase	medical	enron
F-LP	**67.8 ± 2.1**	**57.1 ± 1.0**	**73.6 ± 0.8**	63.8 ± 0.4	51.8 ± 1.1	98.5 ± 0.5	80.8 ± 1.0	51.4 ± 1.1
LP	**67.8 ± 2.1**	**57.1 ± 1.0**	**73.6 ± 0.7**	63.8 ± 0.4	51.5 ± 0.8	98.1 ± 0.6	77.0 ± 1.1	50.4 ± 0.5
PCC	65.6 ± 1.4	51.0 ± 2.2	67.4 ± 1.0	64.0 ± 0.6	na	na	na	na
MCC	67.6 ± 2.2	54.4 ± 1.4	68.5 ± 1.4	63.9 ± 0.7	50.9 ± 0.9	98.5 ± 0.5	79.9 ± 1.0	50.9 ± 0.8
BR	64.4 ± 1.2	41.8 ± 2.2	67.4 ± 0.8	63.1 ± 0.6	51.3 ± 1.1	98.5 ± 0.5	79.5 ± 1.0	50.9 ± 0.8
RAkEL	68.1 ± 1.2	57.4 ± 0.9	72.4 ± 0.9	**65.1 ± 0.4**	51.9 ± 0.9	98.5 ± 0.5	79.6 ± 1.0	52.2 ± 0.8
HOMER	67.5 ± 1.2	52.7 ± 1.1	64.0 ± 2.7	64.8 ± 0.6	47.4 ± 1.0	98.5 ± 0.6	73.4 ± 3.0	48.8 ± 0.7
CC	64.4 ± 2.3	54.2 ± 1.3	68.4 ± 1.5	61.8 ± 0.8	50.3 ± 0.9	98.5 ± 0.6	80.3 ± 1.1	50.5 ± 0.8
ECC	67.1 ± 1.3	54.6 ± 1.2	71.9 ± 1.2	64.9 ± 0.5	**54.1 ± 1.0**	**98.6 ± 0.5**	**81.1 ± 1.5**	**54.3 ± 0.6**
LEAD	60.5 ± 2.1	48.4 ± 1.5	65.1 ± 1.3	62.6 ± 0.5	45.8 ± 0.8	98.3 ± 0.7	78.2 ± 0.9	47.3 ± 0.8

method	emotions2	image2	scene2	yeast2	slashdot2	genbase2	medical2	enron2
F-LP	62.3 ± 1.2	**54.6 ± 1.0**	**70.0 ± 0.6**	60.1 ± 0.6	49.7 ± 0.4	**98.5 ± 0.4**	**76.1 ± 0.7**	47.1 ± 0.7
LP	55.6 ± 1.2	47.6 ± 1.0	66.3 ± 0.9	57.4 ± 0.5	34.5 ± 0.7	86.7 ± 1.0	49.2 ± 1.2	34.1 ± 2.0
PCC	63.4 ± 1.1	50.6 ± 1.2	63.4 ± 0.5	na	na	na	na	na
MCC	62.5 ± 1.6	50.6 ± 0.7	64.9 ± 0.7	61.9 ± 0.7	48.5 ± 0.5	98.4 ± 0.4	75.5 ± 1.0	50.2 ± 0.5
BR	61.6 ± 0.9	45.2 ± 0.9	63.9 ± 0.4	62.1 ± 0.5	48.4 ± 0.5	**98.5 ± 0.4**	75.0 ± 1.2	50.1 ± 0.5
RAkEL	**63.8 ± 0.9**	50.4 ± 1.0	67.9 ± 0.6	**62.7 ± 0.4**	49.1 ± 0.5	**98.5 ± 0.4**	74.8 ± 1.3	**51.0 ± 0.5**
HOMER	62.7 ± 0.8	52.5 ± 0.5	62.0 ± 0.7	62.3 ± 0.4	34.6 ± 0.7	91.4 ± 1.3	64.8 ± 1.4	44.2 ± 6.3
CC	59.9 ± 0.7	52.0 ± 0.9	64.5 ± 0.9	59.9 ± 0.5	48.6 ± 0.5	98.4 ± 0.4	75.4 ± 1.3	50.0 ± 0.4
ECC	**63.8 ± 0.9**	54.0 ± 0.8	68.7 ± 0.9	62.5 ± 0.5	**51.0 ± 0.6**	98.4 ± 0.4	75.4 ± 1.3	52.6 ± 0.3
LEAD	57.9 ± 1.5	44.5 ± 1.2	59.7 ± 1.4	61.7 ± 0.4	43.9 ± 0.6	97.9 ± 0.3	75.8 ± 1.1	46.9 ± 0.5

Tab. A.3. Macro-F_1 score (mean $\pm$ std in percent) achieved by comparative methods on the original and the duplicated benchmark, over 5x2-CV. Best results are bold-faced (higher is better).

method	emotions	image	scene	yeast	slashdot	genbase	medical	enron
F-LP	**67.0 ± 2.2**	57.8 ± 1.0	74.4 ± 0.8	41.2 ± 0.5	31.9 ± 1.1	7.3 ± 2.9	**25.1 ± 2.8**	18.7 ± 1.5
LP	**67.0 ± 2.2**	57.8 ± 1.0	**74.5 ± 0.7**	41.2 ± 0.5	31.4 ± 0.7	**10.1 ± 4.4**	24.3 ± 2.7	17.3 ± 1.3
PCC	64.2 ± 1.7	49.3 ± 3.2	68.2 ± 1.0	33.6 ± 0.7	na	na	na	na
MCC	66.3 ± 2.0	54.4 ± 1.4	69.7 ± 1.4	37.3 ± 2.0	32.0 ± 1.1	7.3 ± 2.9	25.0 ± 2.9	18.9 ± 0.9
BR	60.3 ± 2.3	40.7 ± 2.6	68.1 ± 0.9	32.5 ± 0.5	31.8 ± 1.0	7.6 ± 2.5	24.9 ± 2.8	18.9 ± 1.0
RAkEL	65.8 ± 1.3	**58.0 ± 1.0**	73.4 ± 0.9	36.3 ± 0.4	32.2 ± 1.0	7.7 ± 3.4	25.0 ± 2.9	**19.3 ± 1.0**
HOMER	65.9 ± 1.2	53.0 ± 1.2	64.6 ± 2.8	**42.2 ± 1.1**	29.9 ± 1.1	7.4 ± 3.2	23.8 ± 3.0	18.7 ± 1.2
CC	60.3 ± 3.3	54.3 ± 1.5	69.6 ± 1.5	36.3 ± 1.3	31.6 ± 1.1	7.3 ± 3.7	25.0 ± 2.9	18.7 ± 1.4
ECC	65.5 ± 1.3	54.8 ± 1.4	72.8 ± 1.2	36.5 ± 0.9	**32.7 ± 0.9**	8.3 ± 4.0	25.0 ± 2.1	18.9 ± 1.3
LEAD	57.1 ± 2.6	48.4 ± 1.7	65.1 ± 1.4	32.4 ± 0.8	23.1 ± 1.2	4.9 ± 3.3	23.0 ± 1.5	9.8 ± 0.8

method	emotions2	image2	scene2	yeast2	slashdot2	genbase2	medical2	enron2
F-LP	61.6 ± 1.0	**55.2 ± 1.0**	**71.1 ± 0.5**	39.8 ± 0.9	27.9 ± 0.7	6.2 ± 2.5	18.6 ± 0.4	14.0 ± 0.9
LP	54.7 ± 1.2	48.1 ± 0.9	67.4 ± 0.8	34.7 ± 0.5	13.0 ± 0.4	**30.8 ± 3.1**	11.6 ± 0.4	11.4 ± 0.8
PCC	**62.6 ± 1.1**	50.3 ± 1.3	64.3 ± 0.4	na	na	na	na	na
MCC	61.4 ± 2.0	50.7 ± 0.8	66.1 ± 0.6	36.7 ± 0.8	28.1 ± 0.8	5.9 ± 2.1	18.1 ± 0.6	**16.3 ± 0.6**
BR	59.6 ± 1.0	45.0 ± 0.9	64.8 ± 0.4	33.8 ± 0.2	27.7 ± 0.8	6.1 ± 2.3	18.3 ± 0.6	**16.3 ± 0.6**
RAkEL	62.5 ± 1.0	50.5 ± 1.0	68.8 ± 0.6	36.5 ± 0.3	27.8 ± 0.7	6.4 ± 2.8	18.2 ± 0.7	16.2 ± 0.6
HOMER	61.7 ± 0.8	52.9 ± 0.5	63.1 ± 0.8	**42.6 ± 1.0**	22.7 ± 1.0	6.4 ± 2.7	**21.1 ± 1.9**	13.8 ± 3.8
CC	58.1 ± 0.8	52.3 ± 0.9	65.7 ± 0.7	38.8 ± 0.7	**28.3 ± 0.7**	6.5 ± 2.8	18.3 ± 0.8	16.2 ± 0.6
ECC	**62.6 ± 1.1**	54.4 ± 0.8	69.9 ± 0.9	38.0 ± 0.6	27.7 ± 0.7	6.3 ± 2.2	18.3 ± 0.8	15.9 ± 0.7
LEAD	53.8 ± 2.3	44.3 ± 1.2	59.4 ± 1.5	31.5 ± 0.5	22.1 ± 0.7	4.6 ± 2.2	20.2 ± 1.8	9.2 ± 0.6

B Proofs

For the sake of conciseness, the obvious Symmetry property (i.e., $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$ equivalent to $\mathbf{Y} \perp\!\!\!\perp \mathbf{X} \mid \mathbf{Z}$) will be used implicitly in the proofs.

Proof of Theorem 5.1. First, we prove that $\mathbf{Y}_{F_i} \cup \mathbf{Y}_{F_j} \in \mathcal{F}$. From the LF assumption for $\mathbf{Y}_{F_i}$ and $\mathbf{Y}_{F_j}$ we have $\mathbf{Y}_{F_i} \perp\!\!\!\perp \mathbf{Y} \setminus \mathbf{Y}_{F_i} \mid \mathbf{X}$ and $\mathbf{Y}_{F_j} \perp\!\!\!\perp \mathbf{Y} \setminus \mathbf{Y}_{F_j} \mid \mathbf{X}$. Using the Weak Union property we obtain that $\mathbf{Y}_{F_i} \perp\!\!\!\perp \mathbf{Y} \setminus (\mathbf{Y}_{F_i} \cup \mathbf{Y}_{F_j}) \mid \mathbf{X} \cup \mathbf{Y}_{F_j} \setminus \mathbf{Y}_{F_i}$, and similarly with the Decomposition property we get $\mathbf{Y}_{F_j} \setminus \mathbf{Y}_{F_i} \perp\!\!\!\perp \mathbf{Y} \setminus (\mathbf{Y}_{F_i} \cup \mathbf{Y}_{F_j}) \mid \mathbf{X}$. We may now apply the Contraction property to show that $\mathbf{Y}_{F_i} \cup \mathbf{Y}_{F_j} \perp\!\!\!\perp \mathbf{Y} \setminus (\mathbf{Y}_{F_i} \cup \mathbf{Y}_{F_j}) \mid \mathbf{X}$. Therefore, $\mathbf{Y}_{F_i} \cup \mathbf{Y}_{F_j}$ is a LF by definition. Second, we prove that $\mathbf{Y}_{F_i} \cap \mathbf{Y}_{F_j} \in \mathcal{F}$. From the LF assumption for $\mathbf{Y}_{F_i}$ and $\mathbf{Y}_{F_j}$ we have $\mathbf{Y}_{F_i} \perp\!\!\!\perp \mathbf{Y} \setminus \mathbf{Y}_{F_i} \mid \mathbf{X}$ and $\mathbf{Y}_{F_j} \perp\!\!\!\perp \mathbf{Y} \setminus \mathbf{Y}_{F_j} \mid \mathbf{X}$. Using the Weak Union property we obtain $\mathbf{Y}_{F_i} \cap \mathbf{Y}_{F_j} \perp\!\!\!\perp (\mathbf{Y} \setminus (\mathbf{Y}_{F_i} \cup \mathbf{Y}_{F_j})) \cup (\mathbf{Y}_{F_j} \setminus \mathbf{Y}_{F_i}) \mid \mathbf{X} \cup \mathbf{Y}_{F_i} \setminus \mathbf{Y}_{F_j}$, and similarly with the Decomposition property we get $\mathbf{Y}_{F_i} \cap \mathbf{Y}_{F_j} \perp\!\!\!\perp \mathbf{Y}_{F_i} \setminus \mathbf{Y}_{F_j} \mid \mathbf{X}$. We may now apply the Contraction property to show that $\mathbf{Y}_{F_i} \cap \mathbf{Y}_{F_j} \perp\!\!\!\perp \mathbf{Y} \setminus (\mathbf{Y}_{F_i} \cap \mathbf{Y}_{F_j}) \mid \mathbf{X}$. Therefore, $\mathbf{Y}_{F_i} \cap \mathbf{Y}_{F_j}$ is a LF by definition. Third, we prove that $\mathbf{Y}_{F_i} \setminus \mathbf{Y}_{F_j} \in \mathcal{F}$. From the LF assumption for $\mathbf{Y}_{F_i}$ and $\mathbf{Y}_{F_j}$ we have $\mathbf{Y}_{F_i} \perp\!\!\!\perp \mathbf{Y} \setminus \mathbf{Y}_{F_i} \mid \mathbf{X}$ and $\mathbf{Y}_{F_j} \perp\!\!\!\perp \mathbf{Y} \setminus \mathbf{Y}_{F_j} \mid \mathbf{X}$. Using the Weak Union property we obtain $\mathbf{Y}_{F_i} \setminus \mathbf{Y}_{F_j} \perp\!\!\!\perp \mathbf{Y} \setminus \mathbf{Y}_{F_i} \mid \mathbf{X} \cup \mathbf{Y}_{F_j}$, and similarly with the Decomposition property we get $\mathbf{Y}_{F_j} \perp\!\!\!\perp \mathbf{Y}_{F_i} \setminus \mathbf{Y}_{F_j} \mid \mathbf{X}$. We may now apply the Contraction property to show that $\mathbf{Y}_{F_i} \setminus \mathbf{Y}_{F_j} \perp\!\!\!\perp \mathbf{Y} \setminus (\mathbf{Y}_{F_i} \setminus \mathbf{Y}_{F_j}) \mid \mathbf{X}$. Therefore, $\mathbf{Y}_{F_i} \setminus \mathbf{Y}_{F_j}$ is a LF by definition. Finally, we prove that $\mathcal{F}_I$ forms a partition of $\mathbf{Y}$. Consider a non-empty LF $\mathbf{Y}_{F_i} \in \mathcal{F}$. Then either $\mathbf{Y}_{F_i}$ is an ILF, or one of its proper non-empty subsets $\mathbf{Y}_{F_j} \subset \mathbf{Y}_{F_i}$ is an ILF and the remaining set $\mathbf{Y}_{F_i} \setminus \mathbf{Y}_{F_j}$ is a non-empty LF. By applying the same reasoning recursively, the non-empty LF $\mathbf{Y} \in \mathcal{F}$ breaks down into an irreducible partition of ILFs. Now, consider two distinct ILFs $\mathbf{Y}_{F_i}, \mathbf{Y}_{F_j} \in \mathcal{F}_I$, then $\mathbf{Y}_{F_i} \cap \mathbf{Y}_{F_j}$ is a LF, which is necessarily empty due to the ILF assumption for $\mathbf{Y}_{F_i}$ or $\mathbf{Y}_{F_j}$. As a result all ILFs are mutually disjoint, and $\mathcal{F}_I$ forms a unique partition of $\mathbf{Y}$. □

We now introduce Lemmas B.1 and B.2 that will be appear recurrently our subsequent demonstrations.

Lem. B.1 *Two distinct labels Y_i and Y_j belong to the same irreducible label factor if there exists $\mathbf{Z} \subseteq \mathbf{Y} \setminus \{Y_i, Y_j\}$ such that $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid (\mathbf{X} \cup \mathbf{Z})$.*

Proof of Lemma B.1. By contradiction, suppose Y_i and Y_j do not belong to the same irreducible label factor, and let $\mathbf{Y}_{F_i}$ denote the irreducible label factor to which Y_i belongs. From the label factor definition we have $\mathbf{Y}_{F_i} \perp\!\!\!\perp \mathbf{Y} \setminus \mathbf{Y}_{F_i} \mid \mathbf{X}$. Let $\mathbf{Z}$ denote any arbitrary subset of $\mathbf{Y} \setminus \{Y_i, Y_j\}$, we can apply the Weak Union property to obtain $\mathbf{Y}_{F_i} \setminus \mathbf{Z} \perp\!\!\!\perp \mathbf{Y} \setminus (\mathbf{Y}_{F_i} \cup \mathbf{Z}) \mid \mathbf{X} \cup \mathbf{Z}$. Then, from the Decomposition property we have $\{Y_i\} \perp\!\!\!\perp \{Y_j\} \mid \mathbf{X} \cup \mathbf{Z}$. This is true for every such a $\mathbf{Z}$ subset, which concludes the proof. □

Lem. B.2 *Let $\mathbf{Y}_F$ be an irreducible label factor. Then, for every nonempty proper subset $\mathbf{Z}$ of $\mathbf{Y}_F$, we have $\mathbf{Z} \not\perp\!\!\!\perp \mathbf{Y}_F \setminus \mathbf{Z} \mid \mathbf{X} \cup \mathbf{Y} \setminus \mathbf{Y}_F$.*

Proof of Lemma B.2. By contradiction, suppose such a $\mathbf{Z}$ exists with $\mathbf{Z} \perp\!\!\!\perp \mathbf{Y}_F \setminus \mathbf{Z} \mid \mathbf{X} \cup \mathbf{Y} \setminus \mathbf{Y}_F$. From the label factor assumption of $\mathbf{Y}_F$, we also have that $\mathbf{Y}_F \perp\!\!\!\perp \mathbf{Y} \setminus \mathbf{Y}_F \mid \mathbf{X}$, and therefore $\mathbf{Z} \perp\!\!\!\perp \mathbf{Y} \setminus \mathbf{Y}_F \mid \mathbf{X}$ due to the Decomposition property. We may now apply the Contraction property on these two statements to obtain $\mathbf{Z} \perp\!\!\!\perp \mathbf{Y} \setminus \mathbf{Z} \mid \mathbf{X}$ which contradicts the irreducible label factor assumption for $\mathbf{Y}_F$. This concludes the proof. □

Proof of Theorem 5.3. If a path exists between Y_i and Y_j in $\mathcal{G}$ then owing to Lemma B.1 all pairs of successive labels in the path are in the same ILF, and by transitivity Y_i and Y_j necessarily belong to the same ILF. We may now prove the converse. Suppose that Y_i and Y_j belong to the same irreducible label factor, denoted $\mathbf{Y}_F$. Define $\{\mathbf{V}, \mathbf{W}\}$ a partition of $\mathbf{Y}$ such that $Y_i \in \mathbf{V}$ and $Y_j \in \mathbf{W}$. Then, owing to Lemma B.2, we have that $\mathbf{V} \cap \mathbf{Y}_F \not\perp\!\!\!\perp \mathbf{W} \cap \mathbf{Y}_F \mid \mathbf{X} \cup \mathbf{Y} \setminus \mathbf{Y}_F$. Using the Weak Union property, we obtain $\mathbf{V} \not\perp\!\!\!\perp \mathbf{W} \mid \mathbf{X}$. Consider V_1 an arbitrary label from $\mathbf{V}$. Using the Contraction property, we have that either $\{V_1\} \not\perp\!\!\!\perp \mathbf{W} \mid \mathbf{X}$ or $\mathbf{V} \setminus \{V_1\} \not\perp\!\!\!\perp \mathbf{W} \mid \mathbf{X} \cup \{V_1\}$. Consider V_2 another arbitrary label from $\mathbf{V} \setminus \{V_1\}$, we can apply the Contraction property again on the second expression to obtain that either $\{V_2\} \not\perp\!\!\!\perp \mathbf{W} \mid \mathbf{X} \cup \{V_1\}$or $\mathbf{V} \setminus \{V_1, V_2\} \not\perp\!\!\!\perp \mathbf{W} \mid \mathbf{X} \cup \{V_1, V_2\}$. If we proceed recursively, we will necessarily find a variable $V_k \in \mathbf{V}$ such that $\{V_k\} \not\perp\!\!\!\perp \mathbf{W} \mid \mathbf{X} \cup \{V_1, \ldots, V_{k-1}\}$. Likewise, we can proceed along the same line to exhibit a variable $W_l \in \mathbf{W}$ such that $\{V_k\} \not\perp\!\!\!\perp \{W_l\} \mid \mathbf{X} \cup \{V_1, \ldots, V_{k-1}\} \cup \{W_1, \ldots, W_{l-1}\}$. In other words, for every partition $\{\mathbf{V}, \mathbf{W}\}$ of the labels such that $Y_i \in \mathbf{V}$ and $Y_j \in \mathbf{W}$, there exists at least one label $\{V_k\}$ in $\mathbf{V}$, one label $\{W_l\}$ in $\mathbf{W}$ and one subset $\mathbf{Z} \subseteq \mathbf{Y} \setminus \{V_k, W_l\}$, such that $\{V_k\} \not\perp\!\!\!\perp \{W_l\} \mid \mathbf{X} \cup \mathbf{Z}$. So there necessarily exists a path between Y_i and Y_j in $\mathcal{G}$. This concludes the proof. □

Proof of Theorem 5.4. If p is faithfull to the DAG $\mathcal{G}$, then the adjacency condition in Theorem 5.3 expresses as, there exists a path in $\mathcal{G}$ between Y_i and Y_j that can be made open by conditioning on the features $\mathbf{X}$ and a subset of the remaining labels

$\mathbf{Z} \subseteq \mathbf{Y} \setminus \{Y_i, Y_j\}$. From the d-separation criterion, given a path between Y_i and Y_j, if there exists an intermediate node that is a collider, then either it belongs to $\mathbf{X}$ and it does not block the path, either it belongs to $\mathbf{Y}$ and can be added to $\mathbf{Z}$ so that it does not block the path. If there exists an intermediate node that is a non-collider, then either it belongs to $\mathbf{Y}$ and can be kept out of $\mathbf{Z}$ so that it does not block the path, either it belongs to $\mathbf{X}$ and the path is always closed. So the only paths that satisfy the condition in Theorem 5.3 are those which exhibit no intermediate non-collider node that belongs to $\mathbf{X}$. Therefore, two labels are adjacent in Theorem 5.3 *iff* they are connected by a path with all intermediate non-collider nodes in $\mathbf{Y}$. Moreover, if two nodes Y_i and Y_j fulfill this condition, and the same is true for Y_j and Y_k, then Y_i and Y_k also fulfill the condition by concatenating the two open paths $Y_i - Y_j$ and $Y_j - Y_k$. This is related to the Weak Transitivity property of DAGs. Therefore two labels are connected in Theorem 5.3 and belong to the same ILF *iff* they are connected by a path with all intermediate non-collider nodes in $\mathbf{Y}$. This concludes the proof. □

Proof of Theorem 5.5. If a path exists between Y_i and Y_j in $\mathcal{G}$ then owing to Lemma B.1 all pairs of successive labels in the path are in the same ILF, and by transitivity Y_i and Y_j necessarily belong to the same ILF. We may now prove the converse. Suppose that Y_i and Y_j belong to the same ILF, denoted $\mathbf{Y}_F$. Define $\{\mathbf{Z}, \mathbf{W}\}$ a partition of $\mathbf{Y}$ such that $Y_i \in \mathbf{Z}$ and $Y_j \in \mathbf{W}$. Then, owing to Lemma B.2, we have that $\mathbf{Z} \cap \mathbf{Y}_F \not\perp\!\!\!\perp \mathbf{W} \cap \mathbf{Y}_F \mid \mathbf{X} \cup \mathbf{Y} \setminus \mathbf{Y}_F$. Using the Weak Union property, we obtain $\mathbf{Z} \not\perp\!\!\!\perp \mathbf{W} \mid \mathbf{X}$. To keep the subsequent developments uncluttered, we adopt the notation $\{Z|Z > Y_i\}$ and $\{W|W > Y_i\}$ to denote respectively the sets $\{Y|Y > Y_i, Y \in \mathbf{Z}\}$ and $\{Y|Y > Y_i, Y \in \mathbf{W}\}$, so that $Z \in \mathbf{Z}$ and $W \in \mathbf{W}$ by convention. Now, let Y_1 denote the first label in the ordering, and suppose it belongs to $\mathbf{Z}$. Due to the Contraction property, we have that either $\{Y_1\} \not\perp\!\!\!\perp \{W|W > Y_1\} \mid \mathbf{X} \cup \{Z|Z > Y_1\}$ or $\{Z|Z > Y_1\} \not\perp\!\!\!\perp \{W|W > Y_1\} \mid \mathbf{X}$. Now, let W_1 denote the first label in $\{W|W > Y_1\}$. Due to the Intersection property the first statement extends further into $\{Y_1\} \not\perp\!\!\!\perp \{W_1\} \mid \mathbf{X} \cup \{Z|Z > Y_1\} \cup \{W|W > Y_1\} \setminus \{W_1\}$ or $\{Y_1\} \not\perp\!\!\!\perp \{W|W > W_1\} \mid \mathbf{X} \cup \{Z|Z > Y_1\} \cup \{W_1\}$. Similarly, with W_2 the second label in $\{W|W > Y_1\}$ that last statement extends further into either $\{Y_1\} \not\perp\!\!\!\perp \{W_2\} \mid \mathbf{X} \cup \{Z|Z > Y_1\} \cup \{W|W > Y_1\} \setminus \{W_2\}$ or $\{Y_1\} \not\perp\!\!\!\perp \{W|W > W_2\} \mid \mathbf{X} \cup \{Z|Z > Y_1\} \cup \{W_2\}$. If we proceed recursively, we will necessarily find that either $\{Z|Z > Y_1\} \not\perp\!\!\!\perp \{W|W > Y_1\} \mid \mathbf{X}$ or there exists a label $Y_l \in \{W|W > Y_1\}$ such that $\{Y_1\} \not\perp\!\!\!\perp \{Y_l\} \mid \mathbf{X} \cup \{Y|Y > Y_1\} \setminus \{Y_l\}$. On the other hand, if Y_1 belongs to $\mathbf{W}$, then similarly we have that either $\{Z|Z > Y_1\} \not\perp\!\!\!\perp \{W|W > Y_1\} \mid \mathbf{X}$ or there exists a label $Y_l \in \{Z|Z > Y_1\}$ such that $\{Y_1\} \not\perp\!\!\!\perp \{Y_l\} \mid \mathbf{X} \cup \{Y|Y > Y_1\} \setminus \{Y_l\}$. In both cases we end up with the same result. We may apply the same deduction recursively with Y_2 and $\{Z|Z > Y_1\} \not\perp\!\!\!\perp \{W|W > Y_1\} \mid \mathbf{X}$, then Y_3 and $\{Z|Z > Y_2\} \not\perp\!\!\!\perp \{W|W > Y_2\} \mid \mathbf{X}$ and so on, until we exhibit two labels Y_k and Y_l ($Y_k < Y_l$) such that $\{Y_k\} \not\perp\!\!\!\perp \{Y_l\} \mid \mathbf{X} \cup \{Y|Y > Y_k\} \setminus \{Y_l\}$, with either $Y_k \in \mathbf{Z}$ and

$Y_l \in \mathbf{W}$ or $Y_k \in \mathbf{W}$ and $Y_l \in \mathbf{Z}$. In other words, for every partition $\{\mathbf{Z}, \mathbf{W}\}$ of the label set such that $Y_i \in \mathbf{Z}$ and $Y_j \in \mathbf{W}$, there exists at least one label $\{Y_k\}$ in $\mathbf{Z}$ and one label $\{Y_l\}$ in $\mathbf{W}$ such that $\{Y_k\} \not\perp\!\!\!\perp \{Y_l\} \mid \mathbf{X} \cup \{Y | Y > Y_k\} \setminus \{Y_l\}$ if $Y_k < Y_l$, or $\{Y_l\} \not\perp\!\!\!\perp \{Y_k\} \mid \mathbf{X} \cup \{Y | Y > Y_l\} \setminus \{Y_k\}$ if $Y_l < Y_k$. So there necessarily exists a path between Y_i and Y_j in $\mathcal{G}$. This concludes the proof. □

Proof of Theorem 5.6. We first prove by contradiction that, for any p, if Y_i and Y_j belong to the same irreducible label factor then there exists a path between Y_i and Y_j in $\mathcal{G}$. Suppose there is no such path, i.e., there exists an ordering $Y_1, \ldots, Y_n$ and a partition $\{\mathbf{Z}, \mathbf{W}\}$ of the labels such that $Y_i \in \mathbf{Z}$, $Y_j \in \mathbf{W}$ and every label in $\mathbf{Z}$ is non-adjacent in $\mathcal{G}$ to every label in $\mathbf{W}$. Equivalently, for every Z in $\mathbf{Z}$ there exists a Markov boundary in $\mathbf{U} \setminus \{Y | Y < Z\}$ which does not contain any label from $\mathbf{W}$, and for every W in $\mathbf{W}$ there exists a Markov boundary in $\mathbf{U} \setminus \{Y | Y < W\}$ which does not contain any label from $\mathbf{Z}$. To keep the subsequent developments uncluttered, we adopt the notation $\{Z | Z > Y_i\}$ and $\{W | W > Y_i\}$ to denote respectively the sets $\{Y | Y > Y_i, Y \in \mathbf{Z}\}$ and $\{Y | Y > Y_i, Y \in \mathbf{W}\}$, so that $Z \in \mathbf{Z}$ and $W \in \mathbf{W}$ by convention. Now, let Y_k denote the last label in the ordering that belongs to $\mathbf{Z}$, and $\mathbf{M}_k$ its Markov boundary in $\mathbf{U} \setminus \{Y | Y < Y_k\}$. Then, due to the Markov blanket assumption for $\mathbf{M}_k$ we have that $\{Z | Z \geq Y_k\} \perp\!\!\!\perp \{W | W \geq Y_k\} \cup \mathbf{X} \setminus \mathbf{M}_k \mid \mathbf{M}_k$, on which we apply the Weak Union property to obtain $\{Z | Z \geq Y_k\} \perp\!\!\!\perp \{W | W \geq Y_k\} \mid \mathbf{X}$. We can now turn to the preceding label in the ordering, Y_{k-1}. If that label belongs to $\mathbf{Z}$, then similarly we have $\{Z | Z = Y_{k-1}\} \perp\!\!\!\perp \{W | W \geq Y_{k-1}\} \cup \mathbf{X} \setminus \mathbf{M}_{k-1} \mid \mathbf{M}_{k-1}$ which yields $\{Z | Z = Y_{k-1}\} \perp\!\!\!\perp \{W | W \geq Y_{k-1}\} \mid \mathbf{X} \cup \{Z | Z \geq Y_k\}$ due to the Weak Union property. Note that because $Y_{k-1} \in \mathbf{Z}$ we also have $\{W | W \geq Y_{k-1}\} = \{W | W \geq Y_k\}$, which allows us to apply the Contraction property with our previous result to obtain $\{Z | Z \geq Y_{k-1}\} \perp\!\!\!\perp \{W | W \geq Y_{k-1}\} \mid \mathbf{X}$. On the other hand, if Y_{k-1} belongs to $\mathbf{W}$, then similarly we have $\{Z | Z \geq Y_{k-1}\} \cup \mathbf{X} \setminus \mathbf{M}_{k-1} \perp\!\!\!\perp \{W | W = Y_{k-1}\} \mid \mathbf{M}_{k-1}$ which yields $\{Z | Z \geq Y_{k-1}\} \perp\!\!\!\perp \{W | W = Y_{k-1}\} \mid \mathbf{X} \cup \{W | W \geq Y_k\}$ due to the Weak Union property, and we can apply the Contraction property to obtain $\{Z | Z \geq Y_{k-1}\} \perp\!\!\!\perp \{W | W \geq Y_{k-1}\} \mid \mathbf{X}$. In both cases we end up with the same result. In the same way, we can proceed recursively with every label in reverse order to obtain eventually $\mathbf{Z} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{X}$. Suppose now that Y_i and Y_j do belong to the same irreducible label factor $\mathbf{Y}_F$, then from Lemma B.2 we have that $\mathbf{Z} \cap \mathbf{Y}_F \not\perp\!\!\!\perp \mathbf{W} \cap \mathbf{Y}_F \mid \mathbf{X} \cup \mathbf{Y} \setminus \mathbf{Y}_F$ on which we apply the Weak Union property to obtain $\mathbf{Z} \not\perp\!\!\!\perp \mathbf{W} \mid \mathbf{X}$. This contradicts our initial result, and concludes the first part of this proof.

We shall now turn to the second part and prove that the converse holds when p supports the Intersection property. Suppose that Y_i and Y_j do not belong to the same irreducible label factor, and let $\{\mathbf{Z}, \mathbf{W}\}$ be a partition of the labels such that $\mathbf{Z}$ is the irreducible label factor which contains Y_i. From the label factor assumption for $\mathbf{Z}$ we have that $\mathbf{Z} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{X}$. Now, consider Y_1 an arbitrary label from $\mathbf{Y}$. If Y_1 belongs

to $\mathbf{Z}$ then from the Weak Union property we obtain $\{Y_1\} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{U} \setminus (\mathbf{W} \cup \{Y_1\})$, which defines a Markov blanket of Y_1 in $\mathbf{U}$. Moreover, either this Markov blanket is a Markov boundary, or one of its proper subsets is. Because the Intersection property holds, we have that this Markov boundary is unique and thus $\mathbf{M}_1 \in \mathbf{U} \setminus (\mathbf{W} \cup \{Y_1\})$, which does not contain any label from $\mathbf{W}$. On the other hand, if Y_1 belongs to $\mathbf{W}$, then similarly we have $\mathbf{Z} \perp\!\!\!\perp \{Y_1\} \mid \mathbf{U} \setminus (\mathbf{Z} \cup \{Y_1\})$ and the Markov boundary of Y_1 in $\mathbf{U}$ does not contain any label from $\mathbf{Z}$. We can now go on with a second arbitrary label $Y_2 \in \mathbf{Y} \setminus \{Y_1\}$. From the Decomposition property we have that $\mathbf{Z} \setminus \{Y_1\} \perp\!\!\!\perp \mathbf{W} \setminus \{Y_1\} \mid \mathbf{X}$. If Y_2 belongs to $\mathbf{Z}$ then from the Weak Union property we have $\{Y_2\} \perp\!\!\!\perp \mathbf{W} \setminus \{Y_1\} \mid \mathbf{U} \setminus (\mathbf{W} \cup \{Y_1, Y_2\})$ and from the Intersection property the Markov boundary of Y_2 in $\mathbf{U} \setminus \{Y_1\}$ does not contain any label from $\mathbf{W}$. On the other hand, if Y_2 belongs to $\mathbf{W}$ we have $\mathbf{Z} \setminus \{Y_1\} \perp\!\!\!\perp \{Y_2\} \mid \mathbf{U} \setminus (\mathbf{Z} \cup \{Y_1, Y_2\})$ and the Markov boundary of Y_2 in $\mathbf{U} \setminus \{Y_1\}$ does not contain any label from $\mathbf{Z}$. We can proceed iteratively with every label in the same way, in any order, to obtain that the Markov boundary of each label Y_k in $\mathbf{U} \setminus \{Y | Y < Y_k\}$ does not contain any label from $\mathbf{W}$ when $Y_k \in \mathbf{Z}$, nor any label from $\mathbf{Z}$ when $Y_k \in \mathbf{W}$. So, for every label ordering, there exists a partition $\{\mathbf{Z}, \mathbf{W}\}$ such that $Y_i \in \mathbf{Z}$, $Y_j \in \mathbf{W}$ and no label from $\mathbf{Z}$ is adjacent to a label from $\mathbf{W}$ in $\mathcal{G}$. In other words, for every ordering there may be no path between Y_i and Y_j in $\mathcal{G}$. This concludes the proof. □

Proof of Theorem 5.7. If a path exists between Y_i and Y_j in $\mathcal{G}$ then owing to Lemma B.1 all pairs of successive labels in the path are in the same ILF, and by transitivity Y_i and Y_j necessarily belong to the same ILF. We may now prove the converse. Suppose that Y_i and Y_j belong to the same irreducible label factor, denoted $\mathbf{Y}_F$. Define $\{\mathbf{W}_i, \mathbf{W}_j\}$ a partition of $\mathbf{Y}$ such that $Y_i \in \mathbf{W}_i$ and $Y_j \in \mathbf{W}_j$. Then, owing to Lemma B.2, we have that $\mathbf{W}_i \cap \mathbf{Y}_F \not\perp\!\!\!\perp \mathbf{W}_j \cap \mathbf{Y}_F \mid \mathbf{X} \cup \mathbf{Y} \setminus \mathbf{Y}_F$. Using the Weak Union property, we obtain $\mathbf{W}_i \not\perp\!\!\!\perp \mathbf{W}_j \mid \mathbf{X}$. Consider W_1^i an arbitrary label from $\mathbf{W}_i$. Using the Composition property, we have that either $\{W_1^i\} \not\perp\!\!\!\perp \mathbf{W}_j \mid \mathbf{X}$ or $\mathbf{W}_i \setminus \{W_1^i\} \not\perp\!\!\!\perp \mathbf{W}_j \mid \mathbf{X}$. Consider W_2^i another arbitrary label from $\mathbf{W}_i \setminus \{W_1^i\}$, we can apply the Composition property again on the second expression to obtain that either $\{W_2^i\} \not\perp\!\!\!\perp \mathbf{W}_j \mid \mathbf{X}$ or $\mathbf{W}_i \setminus \{W_1^i, W_2^i\} \not\perp\!\!\!\perp \mathbf{W}_j \mid \mathbf{X}$. If we proceed recursively, we will necessarily find a variable $W_k^i \in \mathbf{W}_i$ such that $\{W_k^i\} \not\perp\!\!\!\perp \mathbf{W}_j \mid \mathbf{X}$. Likewise, we can proceed along the same line to exhibit a variable $W_l^j \in \mathbf{W}_j$ such that $\{W_k^i\} \not\perp\!\!\!\perp \{W_l^j\} \mid \mathbf{X}$. In other words, for every partition $\{\mathbf{W}_i, \mathbf{W}_j\}$ of the labels such that $Y_i \in \mathbf{W}_i$ and $Y_j \in \mathbf{W}_j$, there exists at least one label $\{W_k^i\}$ in $\mathbf{W}_i$ and one label $\{W_l^j\}$ in $\mathbf{W}_j$, such that $\{W_k^i\} \not\perp\!\!\!\perp \{W_l^j\} \mid \mathbf{X}$. So there necessarily exists a path between Y_i and Y_j in $\mathcal{G}$. This concludes the proof. □

We now introduce Lemma B.3 in order to derive Theorem 5.8.

Lem. B.3 *Suppose p supports the Composition property. Then, for two distinct labels Y_i and Y_j, if $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid \mathbf{X}$ then $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid \mathbf{M}_i$ for every Markov blanket $\mathbf{M}_i$ of Y_i in $\mathbf{X}$. Moreover, if there exists a Markov blanket $\mathbf{M}_i$ of Y_i in $\mathbf{X}$ such that $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid \mathbf{M}_i$ then $\{Y_i\} \not\perp\!\!\!\perp \{Y_j\} \mid \mathbf{X}$.*

Proof of Lemma B.3. First, we prove that $\{Y_i\} \perp\!\!\!\perp \{Y_j\} \mid \mathbf{X}$ implies $\{Y_i\} \perp\!\!\!\perp \{Y_j\} \mid \mathbf{M}_i$, for every Markov blanket $\mathbf{M}_i$ of Y_i in $\mathbf{X}$. We may rewrite $\{Y_i\} \perp\!\!\!\perp \{Y_j\} \mid \mathbf{X}$ as $\{Y_i\} \perp\!\!\!\perp \{Y_j\} \mid \mathbf{M}_i \cup \mathbf{X} \setminus \mathbf{M}_i$. From the Markov blanket assumption for $\mathbf{M}_i$, we also have $\{Y_i\} \perp\!\!\!\perp \mathbf{X} \setminus \mathbf{M}_i \mid \mathbf{M}_i$. We can now apply the Contraction property, to obtain $\{Y_i\} \perp\!\!\!\perp \{Y_j\} \cup \mathbf{X} \setminus \mathbf{M}_i \mid \mathbf{M}_i$, and then the Decomposition property which yields $\{Y_i\} \perp\!\!\!\perp \{Y_j\} \mid \mathbf{M}_i$. Second, we prove that if there exists a Markov blanket $\mathbf{M}_i$ of Y_i in $\mathbf{X}$ such that $\{Y_i\} \perp\!\!\!\perp \{Y_j\} \mid \mathbf{M}_i$ then necessarily $\{Y_i\} \perp\!\!\!\perp \{Y_j\} \mid \mathbf{X}$ holds. From the Markov blanket assumption, we have $\{Y_i\} \perp\!\!\!\perp \mathbf{X} \setminus \mathbf{M}_i \mid \mathbf{M}_i$. Using the Composition property we obtain $\{Y_i\} \perp\!\!\!\perp \{Y_j\} \cup \mathbf{X} \setminus \mathbf{M}_i \mid \mathbf{M}_i$, and then from the Weak Union property we have $\{Y_i\} \perp\!\!\!\perp \{Y_j\} \mid \mathbf{X}$. This concludes the proof. □

Proof of Theorem 5.8. If p supports the Composition property, then from Lemma B.3 the statement $\{Y_i\} \perp\!\!\!\perp \{Y_j\} \mid \mathbf{M}_i$, with $\mathbf{M}_i$ a Markov blanket of Y_i in $\mathbf{X}$, is equivalent to the statement $\{Y_i\} \perp\!\!\!\perp \{Y_j\} \mid \mathbf{X}$, and we can use Theorem 5.7 to conclude. □

Proof of Theorem 5.9. To keep the subsequent developments uncluttered, we consider without loss of generality that the label set $\mathbf{Y} = \{Y_1, \ldots, Y_m\}$ is ordered according to $<$, so that $Y_i < Y_j \iff i < j$. Second, we denote $\mathbf{Y}_{ind}^{i,j}$ the set $\mathbf{Y}_{ind}^{i}$ in its intermediary state at line 5 when Y_j is being processed, while $\mathbf{Y}_{ind}^{i}$ denotes its state at the end of the procedure. Last, we adopt the notation $\{Z|Z > Y_k\}$ and $\{W|W > Y_k\}$ to denote respectively the sets $\{Y|Y > Y_k, Y \in \mathbf{Z}\}$ and $\{Y|Y > Y_k, Y \in \mathbf{W}\}$ (with $\mathbf{Z}$, $\mathbf{W}$ subsets of $\mathbf{Y}$), so that $Z \in \mathbf{Z}$ and $W \in \mathbf{W}$ by convention.

We start by proving that Y_i and Y_j are in the same ILF if Y_i and Y_j are connected in $\mathcal{G}$. If two labels Y_p and Y_q (with $Y_p < Y_q$) are adjacent in $\mathcal{G}$, then there exists a set $\mathbf{Y}_{ind}^{p,q}$ such that $\{Y_p\} \not\perp\!\!\!\perp \{Y_q\} \mid \mathbf{X} \cup \{Y|Y < Y_p\} \cup \mathbf{Y}_{ind}^{p,q}$, and from Lemma B.1 Y_p and Y_q belong to the same ILF. Now, if a path exists between Y_i and Y_j in $\mathcal{G}$, then all pairs of successive labels in the path are in the same ILF, and by transitivity Y_i and Y_j necessarily belong to the same ILF.

To show the converse, we shall prove by contradiction that if Y_i and Y_j belong to the same ILF, then there exists a path between Y_i and Y_j in $\mathcal{G}$. Suppose there is no such path, then there exists a partition $\{\mathbf{Z}, \mathbf{W}\}$ of $\mathbf{Y}$ such that $Y_i \in \mathbf{Z}$, $Y_j \in \mathbf{W}$, and every label in $\mathbf{Z}$ is non-adjacent to every label in $\mathbf{W}$. Equivalently, for every label

$Y_k \in \mathbf{Y}$ we have $\{W|W > Y_k\} \subseteq \mathbf{Y}^k_{ind}$ if $Y_k \in \mathbf{Z}$, and $\{Z|Z > Y_k\} \subseteq \mathbf{Y}^k_{ind}$ if $Y_k \in \mathbf{W}$. To proceed, we shall first prove by induction that

$$\forall k > i, \;\; \{Y_i\} \perp\!\!\!\perp \mathbf{Y}^{i,k}_{ind} \mid \mathbf{X} \cup \{Y|Y < Y_i\}.$$

For $k = i + 1$, we have that $\mathbf{Y}^{i,k}_{ind} = \emptyset$ so the result holds trivially. Suppose that $\{Y_i\} \perp\!\!\!\perp \mathbf{Y}^{i,k}_{ind} \mid \mathbf{X} \cup \{Y|Y < Y_i\}$ holds for some k. If $\{Y_i\} \perp\!\!\!\perp \{Y_k\} \mid \mathbf{X} \cup \{Y|Y < Y_i\} \cup \mathbf{Y}^{i,k}_{ind}$, then $\mathbf{Y}^{i,k+1}_{ind} = \mathbf{Y}^{i,k}_{ind} \cup \{Y_k\}$ and $\{Y_i\} \perp\!\!\!\perp \mathbf{Y}^{i,k+1}_{ind} \mid \mathbf{X} \cup \{Y|Y < Y_i\}$ due to the Contraction property. Otherwise, $\mathbf{Y}^{i,k+1}_{ind} = \mathbf{Y}^{i,k}_{ind}$ and we end up with the same result. Therefore, the result holds for every $k > i$ by induction, and setting $k = m$ yields $\{Y_i\} \perp\!\!\!\perp \mathbf{Y}^{i}_{ind} \mid \mathbf{X} \cup \{Y|Y < Y_i\}$. Now, we prove a second result by induction:

$$\forall k, \;\; \{Z|Z \geq Y_k\} \perp\!\!\!\perp \{W|W \geq Y_k\} \mid \mathbf{X} \cup \{Y|Y < Y_k\}.$$

For $k = m$, we have $\{Z|Z \geq Y_k\} = \{Z|Z \geq Y_k\} = \emptyset$ so the result holds trivially. Consider the previous label, Y_{k-1}, and suppose it belongs to $\mathbf{Z}$, then due to our previous result we have $\{Y_{k-1}\} \perp\!\!\!\perp \mathbf{Y}^{k-1}_{ind} \mid \mathbf{X} \cup \{Y|Y < Y_{k-1}\}$. Since $\{W|W > Y_{k-1}\} \subseteq \mathbf{Y}^{k-1}_{ind}$, we may apply the Decomposition property to obtain $\{Y_{k-1}\} \perp\!\!\!\perp \{W|W \geq Y_{k-1}\} \mid \mathbf{X} \cup \{Y|Y < Y_{k-1}\}$. Combining the last expression with $\{Z|Z \geq Y_k\} \perp\!\!\!\perp \{W|W \geq Y_k\} \mid \mathbf{X} \cup \{Y|Y < Y_k\}$ yields $\{Z|Z \geq Y_{k-1}\} \perp\!\!\!\perp \{W|W \geq Y_{k-1}\} \mid \mathbf{X} \cup \{Y|Y < Y_{k-1}\}$ due to the Contraction property. The same demonstration holds if $Y_{k-1} \in \mathbf{W}$. Therefore, the results holds for every k by induction. Setting $k = 1$ in the expression above yields $\mathbf{Z} \perp\!\!\!\perp \mathbf{W} \mid \mathbf{X}$, therefore Y_i and Y_j belong to distinct ILFs. This concludes the proof. □

We now introduce Lemma B.4 in order to derive Theorem 5.10.

Lem. B.4 *Suppose p supports the Intersection property. Let $\mathbf{Y}_1$, $\mathbf{Y}_2$ denote two disjoint subsets of $\mathbf{Y}$, and define $\mathbf{M}_i$ the Markov boundary of $\mathbf{Y}_i$ in $\mathbf{U}$. Then, $\mathbf{M} = (\mathbf{M}_1 \cup \mathbf{M}_2) \setminus (\mathbf{Y}_1 \cup \mathbf{Y}_2)$ is the Markov boundary of $\mathbf{Y}_1 \cup \mathbf{Y}_2$ in $\mathbf{U}$.*

Proof of Lemma B.4. We show first that the statement holds for $n = 2$ and then conclude that it holds for all n by induction. First, we will prove that $\mathbf{M}$ is a Markov blanket. Let $\mathbf{W}$ denote $\mathbf{U} \setminus (\mathbf{Y}_1 \cup \mathbf{Y}_2 \cup \mathbf{M})$. From the Markov blanket assumption of $\mathbf{M}_1$ we have $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \cup (\mathbf{Y}_2 \cup \mathbf{M}_2) \setminus (\mathbf{Y}_1 \cup \mathbf{M}_1) \mid \mathbf{M}_1$, on which we apply the Weak Union property to obtain $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \mid \mathbf{M} \cup \mathbf{Y}_2$. Similarly we can derive $\mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \mid \mathbf{M} \cup \mathbf{Y}_1$. Combining these two statements yields $\mathbf{Y}_1 \cup \mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \mid \mathbf{M}$ due to the Intersection property, which is the definition of a Markov blanket of $\mathbf{Y}_1 \cup \mathbf{Y}_2$ in $\mathbf{U}$. We will now prove that $\mathbf{M}$ is minimal. Suppose there exists $\mathbf{Z} \subseteq \mathbf{M}$ such that $\mathbf{Y}_1 \cup \mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{Z} \mid \mathbf{M} \setminus \mathbf{Z}$, and let $\{\mathbf{Z}_1, \mathbf{Z}_2\}$ be a partition of $\mathbf{Z}$ such that $\mathbf{Z}_1 = \mathbf{Z} \cap \mathbf{M}_1$ (and thus $\mathbf{Z}_2 \subseteq \mathbf{Z} \cap \mathbf{M}_2$). From the Weak Union property we have $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{Z}_1 \mid \mathbf{M} \setminus \mathbf{Z}_1 \cup \mathbf{Y}_2$. From the Markov blanket assumption of $\mathbf{M}_1$ we also

have $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \cup (\mathbf{M}_2 \cup \mathbf{Y}_2) \setminus \mathbf{M}_1 \mid \mathbf{M}_1$. Combining these two statements yields $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \cup (\mathbf{M}_2 \cup \mathbf{Y}_2) \setminus \mathbf{M}_1 \cup \mathbf{Z}_1 \mid \mathbf{M}_1 \setminus \mathbf{Z}_1$ due to the Intersection property. Similarly, we can derive $\mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \cup (\mathbf{M}_1 \cup \mathbf{Y}_1) \setminus \mathbf{M}_2 \cup \mathbf{Z}_2 \mid \mathbf{M}_2 \setminus \mathbf{Z}_2$. Therefore, we have $\mathbf{Z}_1 = \emptyset$ and $\mathbf{Z}_2 = \emptyset$ due to the Markov boundary assumption of respectively $\mathbf{M}_1$ and $\mathbf{M}_2$, which implies $\mathbf{Z} = \emptyset$ and $\mathbf{M}$ is a Markov boundary of $\mathbf{Y}_1 \cup \mathbf{Y}_2$ in $\mathbf{U}$. To conclude for any $n > 2$, it suffices to set $\mathbf{Y}_1$ equal to $\mathbf{Y}_1 \cup \ldots \cup \mathbf{Y}_{n-1}$ and $\mathbf{Y}_2 = \mathbf{Y}_n$ to conclude by induction. This concludes the proof. □

Proof of Theorem 5.10. If p is faithful to a DAG, then the Markov boundary in $\mathbf{U}$ of each label Y_i is given by $\mathbf{M}_i = \mathbf{PC}_{Y_i} \cup \mathbf{SP}_{Y_i}$. Also, p supports the Intersection property so we can use Lemma B.4 to conclude. □

Proof of Theorem 5.11. We show first that the statement holds for $n = 2$ and then conclude that it holds for all n by induction. Let $\mathbf{W}$ denote $\mathbf{U} \setminus (\mathbf{Y}_1 \cup \mathbf{Y}_2 \cup \mathbf{M})$. From the Markov blanket assumption of $\mathbf{M}_1$ we have $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \cup (\mathbf{M}_2 \cup \mathbf{Y}_2) \setminus \mathbf{M}_1 \mid \mathbf{M}_1$, on which we apply the Weak Union property to obtain $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \mid \mathbf{M} \cup \mathbf{Y}_2$. Similarly we can derive $\mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \mid \mathbf{M}$. Combining these two statements yields $\mathbf{Y}_1 \cup \mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \mid \mathbf{M}$ due to the Contraction property, which is the definition of a Markov blanket of $\mathbf{Y}_1 \cup \mathbf{Y}_2$ in $\mathbf{U}$. We will now prove that $\mathbf{M}$ is minimal when p supports the Intersection assumption. Suppose there exists $\mathbf{Z} \subseteq \mathbf{M}$ such that $\mathbf{Y}_1 \cup \mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{Z} \mid \mathbf{M} \setminus \mathbf{Z}$, and let $\{\mathbf{Z}_1, \mathbf{Z}_2\}$ be a partition of $\mathbf{Z}$ such that $\mathbf{Z}_1 = \mathbf{Z} \cap \mathbf{M}_1$ (and thus $\mathbf{Z}_2 \subseteq \mathbf{Z} \cap \mathbf{M}_2$). First, we will prove that $\mathbf{Z}_1 = \emptyset$. From the Weak Union property we have $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{Z}_1 \mid (\mathbf{M} \cup \mathbf{Y}_2) \setminus \mathbf{Z}_1$. From the Markov blanket assumption of $\mathbf{M}_1$ we also have $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \cup (\mathbf{M}_2 \cup \mathbf{Y}_2) \setminus \mathbf{M}_1 \mid \mathbf{M}_1$. Combining these two statements yields $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \cup (\mathbf{M}_2 \cup \mathbf{Y}_2) \setminus \mathbf{M}_1 \cup \mathbf{Z}_1 \mid \mathbf{M}_1 \setminus \mathbf{Z}_1$ due to the Intersection property, and therefore $\mathbf{Z}_1 = \emptyset$ due to the Markov boundary assumption of $\mathbf{M}_1$. Second, we prove that $\mathbf{Z}_2 = \emptyset$. From the Decomposition property we have $\mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{Z} \mid \mathbf{M} \setminus \mathbf{Z}$, on which we apply the Weak Union property to obtain $\mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{Z}_2 \mid \mathbf{M} \setminus \mathbf{Z}_2$. From the Markov blanket assumption of $\mathbf{M}_2$ we also have $\mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{M}_1 \setminus (\mathbf{Y}_2 \cup \mathbf{M}_2) \mid \mathbf{M}_2$. Again, combining these two statements yields $\mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{M}_1 \setminus (\mathbf{Y}_2 \cup \mathbf{M}_2) \cup \mathbf{Z}_2 \mid \mathbf{M}_2 \setminus \mathbf{Z}_2$ due to the Intersection property, and therefore $\mathbf{Z}_2 = \emptyset$ due to the Markov boundary assumption of $\mathbf{M}_2$. Finally, we obtain $\mathbf{Z} = \mathbf{Z}_1 \cup \mathbf{Z}_2 = \emptyset$, so $\mathbf{M}$ is a Markov boundary of $\mathbf{Y}_1 \cup \mathbf{Y}_2$ in $\mathbf{U}$. To conclude for any $n > 2$, it suffices to set $\mathbf{Y}_1$ equal to $\mathbf{Y}_1 \cup \ldots \cup \mathbf{Y}_{n-1}$ and $\mathbf{Y}_2 = \mathbf{Y}_n$ to conclude by induction. This concludes the proof. □

Proof of Theorem 5.12. We show first that the statement holds for $n = 2$ and then conclude that it holds for all n by induction. Let $\mathbf{W}$ denote $\mathbf{U} \setminus (\mathbf{Y}_1 \cup \mathbf{Y}_2 \cup \mathbf{M})$. From the Markov blanket assumption of $\mathbf{M}_1$ we have $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{M}_2 \setminus \mathbf{M}_1 \mid \mathbf{M}_1$, on which we apply the Weak Union property to obtain $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \mid \mathbf{M}$. Similarly we can

derive $\mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \mid \mathbf{M}$. Combining these two statements yields $\mathbf{Y}_1 \cup \mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \mid \mathbf{M}$ due to the Composition property, which is the definition of a Markov blanket of $\mathbf{Y}_1 \cup \mathbf{Y}_2$ in $\mathbf{U}$.

We will now prove that $\mathbf{M}$ is minimal when p supports the Intersection property. Let $\mathbf{W}$ denote $\mathbf{U} \setminus (\mathbf{Y}_1 \cup \mathbf{Y}_2 \cup \mathbf{M})$. Suppose there exists $\mathbf{Z} \subseteq \mathbf{M}$ such that $\mathbf{Y}_1 \cup \mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{Z} \mid \mathbf{M} \setminus \mathbf{Z}$, and let $\{\mathbf{Z}_1, \mathbf{Z}_2\}$ be a partition of $\mathbf{Z}$ such that $\mathbf{Z}_1 = \mathbf{Z} \cap \mathbf{M}_1$ (and thus $\mathbf{Z}_2 \subseteq \mathbf{Z} \cap \mathbf{M}_2$). From the Decomposition property we have $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{Z} \mid \mathbf{M} \setminus \mathbf{Z}$, on which we apply the Weak Union property to obtain $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{Z}_1 \mid \mathbf{M} \setminus \mathbf{Z}_1$. From the Markov blanket assumption of $\mathbf{M}_1$ we also have $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{M}_2 \setminus \mathbf{M}_1 \mid \mathbf{M}_1$. Combining these two statements yields $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{M}_2 \setminus \mathbf{M}_1 \cup \mathbf{Z}_1 \mid \mathbf{M}_1 \setminus \mathbf{Z}_1$ due to the Intersection property. Similarly, we can derive $\mathbf{Y}_2 \perp\!\!\!\perp \mathbf{W} \cup \mathbf{M}_1 \setminus \mathbf{M}_2 \cup \mathbf{Z}_2 \mid \mathbf{M}_2 \setminus \mathbf{Z}_2$. Therefore, we have $\mathbf{Z}_1 = \emptyset$ and $\mathbf{Z}_2 = \emptyset$ due to the Markov boundary assumption of respectively $\mathbf{M}_1$ and $\mathbf{M}_2$, which implies $\mathbf{Z} = \emptyset$ and $\mathbf{M}$ is a Markov boundary of $\mathbf{Y}_1 \cup \mathbf{Y}_2$ in $\mathbf{U}$. To conclude for any $n > 2$, it suffices to set $\mathbf{Y}_1$ equal to $\mathbf{Y}_1 \cup \ldots \cup \mathbf{Y}_{n-1}$ and $\mathbf{Y}_2 = \mathbf{Y}_n$ to conclude by induction. This concludes the proof. □

Bibliography

[AAN15] Al-Salemi, Bassam, Aziz, Mohd Juzaiddin Ab, and Noah, Shahrul Azman. „Boosting algorithms with topic modeling for multi-label text categorization: A comparative empirical study.“ In: *Journal of Information Science* 41.5 (2015), pp. 732–746 (cit. on p. 103).

[Ad00] Acid, Silvia and de Campos, Luis M. „Learning Right Sized Belief Networks by Means of a Hybrid Methodology.“ In: *PKDD*. Ed. by Zighed, Djamel A., Komorowski, Henryk Jan, and Zytkow, Jan M. Vol. 1910. Lecture Notes in Computer Science. Springer, 2000, pp. 309–315 (cit. on p. 92).

[Ad01] Acid, Silvia and de Campos, Luis M. „A hybrid methodology for learning belief networks: BENEDICT.“ In: *International Journal of Approximate Reasoning* 27.3 (2001), pp. 235–262 (cit. on p. 92).

[Aka74] Akaike, Hirotugu. „A new look at the statistical model identification“. In: *IEEE Transactions on Automatic Control* 19 (1974), pp. 716–723 (cit. on p. 79).

[AMP96] Andersson, Steen A., Madigan, David, and Perlman, Michael D. „An Alternative Markov Property for Chain Graphs.“ In: *UAI*. Ed. by Horvitz, Eric and Jensen, Finn Verner. Morgan Kaufmann, 1996, pp. 40–48 (cit. on p. 46).

[Ant+13] Antonucci, Alessandro, Corani, Giorgio, Mauá, Denis Deratani, and Gabaglio, Sandra. „An Ensemble of Bayesian Networks for Multilabel Classification.“ In: *IJCAI*. Ed. by Rossi, Francesca. IJCAI/AAAI, 2013 (cit. on p. 129).

[Aye94] Ayers, Derek D. „A Bayesian Method Reexamined.“ In: *UAI*. Ed. by Mántaras, Ramon López de and Poole, David. Morgan Kaufmann, 1994, pp. 23–27 (cit. on p. 77).

[BA02] Burnham, Kenneth P. and Anderson, David Robert. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Verlag, 2002 (cit. on p. 80).

[Bes74] Besag, Julian. „Spatial Interaction and the Statistical Analysis of Lattice Systems“. English. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 36.2 (1974), pp. 192–236 (cit. on p. 23).

[BF97] Breiman, L. and Friedman, J. H. „Predicting Multivariate Responses in Multiple Linear Regression“. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 59 (1 1997), pp. 3–54 (cit. on p. 114).

[BG10] Bradley, Joseph K. and Guestrin, Carlos. „Learning Tree Conditional Random Fields.“ In: *ICML*. Ed. by Fürnkranz, Johannes and Joachims, Thorsten. Omnipress, 2010, pp. 127–134 (cit. on p. 132).

[Bis06] Bishop, Christopher M. *Pattern recognition and machine learning*. New York: Springer, 2006 (cit. on p. 95).

[BJ01] Bach, Francis R. and Jordan, Michael I. „Thin Junction Trees.“ In: *NIPS*. Ed. by Dietterich, Thomas G., Becker, Suzanna, and Ghahramani, Zoubin. MIT Press, 2001, pp. 569–576 (cit. on p. 186).

[BLL11] Bielza, Concha, Li, G., and Larrañaga, Pedro. „Multi-dimensional classification with Bayesian networks.“ In: *International Journal of Approximate Reasoning* 52.6 (2011), pp. 705–727 (cit. on pp. 103, 128, 143).

[Bor+15] Borchani, Hanen, Varando, Gherardo, Bielza, Concha, and Larrañaga, Pedro. „A survey on multi-output regression.“ In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5.5 (2015), pp. 216–233 (cit. on p. 108).

[Bre96] Breiman, Leo. „Bagging Predictors.“ In: *Machine Learning* 24.2 (1996), pp. 123–140 (cit. on p. 125).

[BRR98] Blockeel, Hendrik, Raedt, Luc De, and Ramon, Jan. „Top-Down Induction of Clustering Trees.“ In: *ICML*. Ed. by Shavlik, Jude W. Morgan Kaufmann, 1998, pp. 55–63 (cit. on p. 103).

[Bun91] Buntine, Wray L. „Theory Refinement on Bayesian Networks.“ In: *UAI*. Ed. by D'Ambrosio, Bruce and Smets, Philippe. Morgan Kaufmann, 1991, pp. 52–60 (cit. on p. 76).

[CB13] Cussens, James and Bartlett, Mark. „Advances in Bayesian Network Learning using Integer Programming.“ In: *UAI*. Ed. by Nicholson, Ann and Smyth, Padhraic. AUAI Press, 2013 (cit. on p. 81).

[CH16] Chen, Lisha and Huang, Jianhua Z. „Sparse reduced-rank regression with covariance estimation.“ In: *Statistics and Computing* 26.1-2 (2016), pp. 461–470 (cit. on p. 108).

[CH91] Cooper, Gregory F. and Herskovits, Edward. „A Bayesian Method for Constructing Bayesian Belief Networks from Databases.“ In: *UAI*. Ed. by D'Ambrosio, Bruce and Smets, Philippe. Morgan Kaufmann, 1991, pp. 86–94 (cit. on pp. 76, 77, 92).

[Che+02] Cheng, Jie, Greiner, Russell, Kelly, Jonathan, Bell, David A., and Liu, Weiru. „Learning Bayesian networks from data: An information-theory based approach.“ In: *Artificial Intelligence* 137.1-2 (2002), pp. 43–90 (cit. on p. 93).

[Chi+10] Chierichetti, Flavio, Kumar, Ravi, Pandey, Sandeep, and Vassilvitskii, Sergei. „Finding the Jaccard Median.“ In: *SODA*. Ed. by Charikar, Moses. SIAM, 2010, pp. 293–311 (cit. on p. 111).

[Chi02] Chickering, David Maxwell. „Optimal Structure Identification With Greedy Search.“ In: *Journal of Machine Learning Research* 3 (2002), pp. 507–554 (cit. on pp. 81, 82, 93).

[Chi95] Chickering, David Maxwell. „Learning Bayesian Networks is NP-Complete.“ In: *AISTATS*. Ed. by Fisher, Doug and Lenz, Hans-Joachim. Springer, 1995, pp. 121–130 (cit. on p. 81).

[CHM04] Chickering, David Maxwell, Heckerman, David, and Meek, Christopher. „Large-Sample Learning of Bayesian Networks is NP-Hard.“ In: *Journal of Machine Learning Research* 5 (2004), pp. 1287–1330 (cit. on pp. 71, 84, 146).

[CL68] Chow, C. K. and Liu, C. N. „Approximating discrete probability distributions with dependence trees.“ In: *IEEE Transactions on Information Theory* 14.3 (1968), pp. 462–467 (cit. on pp. 120, 186).

[CM02] Chickering, David Maxwell and Meek, Christopher. „Finding Optimal Bayesian Networks.“ In: *UAI*. Ed. by Darwiche, Adnan and Friedman, Nir. Morgan Kaufmann, 2002, pp. 94–102 (cit. on p. 82).

[CMH03] Chickering, David Maxwell, Meek, Christopher, and Heckerman, David. „Large-Sample Learning of Bayesian Networks is NP-Hard.“ In: *UAI*. Ed. by Meek, Christopher and Kjærulff, Uffe. Morgan Kaufmann, 2003, pp. 124–133 (cit. on p. 81).

[CMM12] Cherman, Everton Alvares, Metz, Jean, and Monard, Maria Carolina. „Incorporating label dependency into the binary relevance framework for multi-label classification.“ In: *Expert Systems With Applications* 39.2 (2012), pp. 1647–1655 (cit. on pp. 103, 119, 121).

[Cor+14] Corani, Giorgio, Antonucci, Alessandro, Mauá, Denis Deratani, and Gabaglio, Sandra. „Trading off Speed and Accuracy in Multilabel Classification.“ In: *Probabilistic Graphical Models*. Ed. by Gaag, Linda C. van der and Feelders, A. J. Vol. 8754. Lecture Notes in Computer Science. Springer, 2014, pp. 145–159 (cit. on p. 103).

[Cow+99] Cowell, R. G., Dawid, A. P., Lauritzen, Steffen L., and Spiegelhalter, D. J. *Probabilistic networks and expert systems*. Springer, 1999 (cit. on p. 40).

[Coz13] Cozman, Fábio Gagliardi. „Independence for full conditional probabilities: Structure, factorization, non-uniqueness, and Bayesian networks.“ In: *International Journal of Approximate Reasoning* 54.9 (2013), pp. 1261–1278 (cit. on p. 5).

[Cru+06] Cruz-Ramírez, Nicandro, Acosta-Mesa, Héctor-Gabriel, Barrientos-Martínez, Rocío-Erandi, and Nava-Fernández, Luis-Alonso. „How Good Are the Bayesian Information Criterion and the Minimum Description Length Principle for Model Selection? A Bayesian Network Analysis.“ In: *MICAI*. Ed. by Gelbukh, Alexander F. and García, Carlos A. Reyes. Vol. 4293. Lecture Notes in Computer Science. Springer, 2006, pp. 494–504 (cit. on p. 80).

[Cus11] Cussens, James. „Bayesian network learning with cutting planes.“ In: *UAI*. Ed. by Cozman, Fábio Gagliardi and Pfeffer, Avi. AUAI Press, 2011, pp. 153–160 (cit. on p. 81).

[CW93] Cox, David R. and Wermuth, Nanny. „Linear Dependencies Represented by Chain Graphs“. In: *Statistical Science* 8.3 (1993), pp. 204–218 (cit. on pp. 42, 47).

[CW96] Cox, David R. and Wermuth, Nanny. *Multivariate Dependencies: Models, Analysis and Interpretation*. Chapman and Hall, 1996 (cit. on p. 47).

[Daw10] Dawid, A. Philip. „Beware of the DAG!“ In: *NIPS Causality: Objectives and Assessment*. Ed. by Guyon, Isabelle, Janzing, Dominik, and Schölkopf, Bernhard. Vol. 6. JMLR Proceedings. JMLR.org, 2010, pp. 59–86 (cit. on p. 37).

[Daw79] Dawid, A. Philip. „Conditional Independence in Statistical Theory“. In: *Journal of the Royal Statistical Society, Series B* 41 (1979), pp. 1–31 (cit. on p. 14).

[Daw80] Dawid, A. Philip. „Conditional independence for statistical operations“. In: *The Annals of Statistics* 8.3 (1980), pp. 598–617 (cit. on p. 14).

[DCH10] Dembczynski, Krzysztof, Cheng, Weiwei, and Hüllermeier, Eyke. „Bayes Optimal Multilabel Classification via Probabilistic Classifier Chains.“ In: *ICML*. Ed. by Fürnkranz, Johannes and Joachims, Thorsten. Omnipress, 2010, pp. 279–286 (cit. on pp. 126, 166, 167).

[DD99] Dash, Denver and Druzdzel, Marek J. „A Hybrid Anytime Algorithm for the Construction of Causal Models From Sparse Data.“ In: *UAI*. Ed. by Laskey, Kathryn B. and Prade, Henri. Morgan Kaufmann, 1999, pp. 142–149 (cit. on pp. 91, 92).

[de +02] de Campos, Luis M., Fernández-Luna, Juan M., Gámez, José A., and Puerta, Jose Miguel. „Ant colony optimization for learning Bayesian networks.“ In: *International Journal of Approximate Reasoning* 31.3 (2002), pp. 291–311 (cit. on p. 81).

[de 06] de Campos, Luis M. „A Scoring Function for Learning Bayesian Networks based on Mutual Information and Conditional Independence Tests.“ In: *Journal of Machine Learning Research* 7 (2006), pp. 2149–2187 (cit. on p. 92).

[de 96] de Campos, Luis M. „Characterizations of Decomposable Dependency Models (Research Note).“ In: *Journal of Artificial Intelligence Research* 5 (1996), pp. 289–300 (cit. on p. 40).

[Dem+10] Dembczynski, Krzysztof, Waegeman, Willem, Cheng, Weiwei, and Hüllermeier, Eyke. „Regret Analysis for Performance Metrics in Multi-Label Classification: The Case of Hamming and Subset Zero-One Loss.“ In: *ECML/PKDD (1)*. Ed. by Balcázar, José L., Bonchi, Francesco, Gionis, Aristides, and Sebag, Michèle. Vol. 6321. Lecture Notes in Computer Science. Springer, 2010, pp. 280–295 (cit. on p. 107).

[Dem+11] Dembczynski, Krzysztof, Waegeman, Willem, Cheng, Weiwei, and Hüllermeier, Eyke. „An Exact Algorithm for F-Measure Maximization.“ In: *NIPS*. Ed. by Shawe-Taylor, John, Zemel, Richard S., Bartlett, Peter L., Pereira, Fernando C. N., and Weinberger, Kilian Q. 2011, pp. 1404–1412 (cit. on pp. 107, 111, 171–173).

[Dem+12] Dembczynski, Krzysztof, Waegeman, Willem, Cheng, Weiwei, and Hüllermeier, Eyke. „On label dependence and loss minimization in multi-label classification.“ In: *Machine Learning* 88.1-2 (2012), pp. 5–45 (cit. on pp. 103, 118, 119).

[Dem+13] Dembczynski, Krzysztof, Jachnik, Arkadiusz, Kotlowski, Wojciech, Waegeman, Willem, and Hüllermeier, Eyke. „Optimizing the F-Measure in Multi-Label Classification: Plug-in Rule Approach versus Structured Loss Minimization.“ In: *ICML (3)*. Vol. 28. JMLR Proceedings. JMLR.org, 2013, pp. 1130–1138 (cit. on p. 176).

[Den+09] Deng, Yue, Li, Dong, Xie, Xudong, Lam, Kin-Man, and Dai, Qionghai. „Partially occluded face completion and recognition.“ In: *ICIP*. IEEE, 2009, pp. 4145–4148 (cit. on p. 113).

[dFP03] de Campos, Luis M., Fernández-Luna, Juan M., and Puerta, Jose Miguel. „An iterated local search algorithm for learning Bayesian networks with restarts based on conditional independence tests.“ In: *International Journal of Intelligent Systems* 18.2 (2003), pp. 221–235 (cit. on p. 92).

[Dic45] Dice, Lee Raymond. „Measures of the Amount of Ecologic Association Between Species“. In: *Ecology* 26.3 (July 1945), pp. 297–302 (cit. on p. 111).

[dJ11] de Campos, Cassio Polpo and Ji, Qiang. „Efficient Structure Learning of Bayesian Networks using Constraints.“ In: *Journal of Machine Learning Research* 12 (2011), pp. 663–689 (cit. on p. 81).

[DR08] Drton, Mathias and Richardson, Thomas S. „Binary Models for Marginal Independence“. In: *Journal of the Royal Statistical Society* 70.2 (2008), pp. 287–309 (cit. on pp. 42, 43).

[Drt09] Drton, Mathias. „Discrete Chain Graph Models“. In: *Bernoulli* 15 (Sept. 2009), pp. 736–753 (cit. on pp. 43, 48, 51–53).

[DWH12] Dembczynski, Krzysztof, Waegeman, Willem, and Hüllermeier, Eyke. „An Analysis of Chaining in Multi-Label Classification.“ In: *ECAI*. Ed. by Raedt, Luc De, Bessière, Christian, Dubois, Didier, et al. Vol. 242. Frontiers in Artificial Intelligence and Applications. IOS Press, 2012, pp. 294–299 (cit. on pp. 120, 124, 126, 167).

[FMY14] Fan, Xiannian, Malone, Brandon M., and Yuan, Changhe. „Finding Optimal Bayesian Network Structures with Constraints Learned from Data.“ In: *UAI*. Ed. by Zhang, Nevin L. and Tian, Jin. AUAI Press, 2014, pp. 200–209 (cit. on p. 81).

[FNP99] Friedman, Nir, Nachman, Iftach, and Pe'er, Dana. „Learning Bayesian Network Structure from Massive Datasets: The "Sparse Candidate" Algorithm.“ In: *UAI*. Ed. by Laskey, Kathryn B. and Prade, Henri. Morgan Kaufmann, 1999, pp. 206–215 (cit. on pp. 92, 93).

[Fry90] Frydenberg, Morten. „The chain graph Markov property“. In: *Scandinavian Journal of Statistics* 17.4 (1990), pp. 333–353 (cit. on pp. 44, 45).

[FY96] Friedman, Nir and Yakhini, Zohar. „On the Sample Complexity of Learning Bayesian Networks.“ In: *UAI*. Ed. by Horvitz, Eric and Jensen, Finn Verner. Morgan Kaufmann, 1996, pp. 274–282 (cit. on p. 80).

[GD13] Gens, Robert and Domingos, Pedro M. „Learning the Structure of Sum-Product Networks.“ In: *ICML (3)*. Vol. 28. JMLR Proceedings. JMLR.org, 2013, pp. 873–880 (cit. on p. 185).

[GE03] Guyon, Isabelle and Elisseeff, André. „An Introduction to Variable and Feature Selection.“ In: *Journal of Machine Learning Research* 3 (2003), pp. 1157–1182 (cit. on p. 157).

[GEA14] Gharroudi, Ouadie, Elghazel, Haytham, and Aussem, Alex. „A Comparison of Multi-Label Feature Selection Methods Using the Random Forest Paradigm“. In: *Advances in Artificial Intelligence*. Ed. by Sokolova, Marina and Beek, Peter van. Vol. 8436. Lecture Notes in Computer Science. Springer, 2014, pp. 95–106 (cit. on p. 151).

[Gei87] Geiger, Dan. „The Non-axiomatizability of Dependencies in Directed Acyclic Graphs“. In: Report (1987) (cit. on p. 33).

[GG11] Guo, Yuhong and Gu, Suicheng. „Multi-Label Classification Using Conditional Dependency Networks.“ In: *IJCAI*. Ed. by Walsh, Toby. IJCAI/AAAI, 2011, pp. 1300–1305 (cit. on pp. 103, 124, 128).

[GL99] Glover, Fred and Laguna, Manuel. *TABU search*. Kluwer, 1999, pp. I–XIX, 1–382 (cit. on p. 81).

[GM91] Ghosh, Jayanta Kumar and Mukerjee, Rahul. „Characterization of priors under which Bayesian and frequentist Bartlett corrections are equivalent in the multi-parameter case“. In: *Journal of Multivariate Analysis* 38.2 (1991), pp. 385–393 (cit. on p. 6).

[GMS02] Geiger, Dan, Meek, Christopher, and Sturmfels, Bernd. „Factorization of Discrete Probability Distributions.“ In: *UAI*. Ed. by Darwiche, Adnan and Friedman, Nir. Morgan Kaufmann, 2002, pp. 162–169 (cit. on p. 23).

[Goo+14] Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, et al. „Generative Adversarial Nets.“ In: *NIPS*. Ed. by Ghahramani, Zoubin, Welling, Max, Cortes, Corinna, Lawrence, Neil D., and Weinberger, Kilian Q. 2014, pp. 2672–2680 (cit. on p. 118).

[GP88] Geiger, Dan and Pearl, Judea. „On the logic of causal models.“ In: *UAI*. Ed. by Shachter, Ross D., Levitt, Tod S., Kanal, Laveen N., and Lemmer, John F. North-Holland, 1988, pp. 3–14 (cit. on pp. 27, 34).

[GP90] Geiger, Dan and Pearl, Judea. „Logical and algorithmic properties of independence and their application to Bayesian networks.“ In: *Annals of Mathematics and Artificial Intelligence* 2 (1990), pp. 165–178 (cit. on p. 25).

[Grü07] Grünwald, Peter. *Minimum Description Length Principle*. MIT press, Cambridge, MA, 2007 (cit. on pp. 78, 80).

[GS04] Godbole, Shantanu and Sarawagi, Sunita. „Discriminative Methods for Multi-Labeled Classification“. In: *In Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2004, pp. 22–30 (cit. on p. 121).

[GS98] Gómez-Villegas, Miguel A. and Sanz, Luis. „Reconciling Bayesian and frequentist evidence in the point null testing problem“. In: *Test* 7.1 (1998), pp. 207–216 (cit. on p. 6).

[GVP89] Geiger, Dan, Verma, Thomas, and Pearl, Judea. „d-Separation: From Theorems to Algorithms.“ In: *UAI*. Ed. by Henrion, Max, Shachter, Ross D., Kanal, Laveen N., and Lemmer, John F. North-Holland, 1989, pp. 139–148 (cit. on p. 27).

[GVP90] Geiger, Dan, Verma, Thomas, and Pearl, Judea. „Identifying independence in bayesian networks.“ In: *Networks* 20.5 (1990), pp. 507–534 (cit. on pp. 27, 28).

[Hal+09] Hall, Mark A., Frank, Eibe, Holmes, Geoffrey, et al. „The WEKA data mining software: an update.“ In: *SIGKDD Explorations* 11.1 (2009), pp. 10–18 (cit. on p. 167).

[Han+15] Han, Chao, Chen, Jian, Wu, Qingyao, Mu, Shuai, and Min, Huaqing. „Sparse Markov chain-based semi-supervised multi-instance multi-label method for protein function prediction.“ In: *Journal of Bioinformatics and Computational Biology* 13.5 (2015) (cit. on p. 103).

[HC71] Hammersley, John M. and Clifford, Peter E. „Markov random fields on finite graphs and lattices“. In: Unpublished manuscript (1971) (cit. on p. 23).

[Hec+00] Heckerman, David, Chickering, David Maxwell, Meek, Christopher, Rounthwaite, Robert, and Kadie, Carl Myers. „Dependency Networks for Inference, Collaborative Filtering, and Data Visualization.“ In: *Journal of Machine Learning Research* 1 (2000), pp. 49–75 (cit. on p. 127).

[HGC95] Heckerman, David, Geiger, Dan, and Chickering, David Maxwell. „Learning Bayesian Networks: The Combination of Knowledge and Statistical Data.“ In: *Machine Learning* 20.3 (1995), pp. 197–243 (cit. on pp. 74, 81).

[HM81] Howard, Ronald A. and Matheson, James E. „Influence diagrams“. In: *Readings on the Principles and Applications of Decision Analysis* 2 (1981). Ed. by Howard, Ronald A. and Matheson, James E. (cit. on p. 27).

[Ho95] Ho, Tin Kam. „Random decision forests.“ In: *ICDAR*. IEEE Computer Society, 1995, pp. 278–282 (cit. on p. 125).

[Isi25] Ising, Ernst. „Beitrag zur Theorie des Ferromagnetismus“. German. In: *Zeitschrift für Physik* 31.1 (1925), pp. 253–258 (cit. on pp. 20, 64).

[Jaa+10] Jaakkola, Tommi S., Sontag, David, Globerson, Amir, and Meila, Marina. „Learning Bayesian Network Structure using LP Relaxations.“ In: *AISTATS*. Ed. by Teh, Yee Whye and Titterington, D. Mike. Vol. 9. JMLR Proceedings. JMLR.org, 2010, pp. 358–365 (cit. on p. 81).

[Jan07] Jansche, Martin. „A Maximum Expected Utility Framework for Binary Sequence Labeling.“ In: *ACL*. Ed. by Carroll, John A., Bosch, Antal van den, and Zaenen, Annie. The Association for Computational Linguistics, 2007 (cit. on pp. 171, 172).

[JFY09] Joachims, Thorsten, Finley, Thomas, and Yu, Chun-Nam John. „Cutting-plane training of structural SVMs.“ In: *Machine Learning* 77.1 (2009), pp. 27–59 (cit. on p. 132).

[Kau96] Kauermann, Göran. „On a Dualization of Graphical Gaussian Models“. In: *Scandinavian Journal of Statistics* 23.1 (1996), pp. 105–116 (cit. on p. 42).

[KF09] Koller, Daphne and Friedman, Nir. *Probabilistic Graphical Models - Principles and Techniques.* MIT Press, 2009, pp. I–XXXV, 1–1231 (cit. on pp. 3, 91, 96).

[Koc+07] Kocev, Dragi, Vens, Celine, Struyf, Jan, and Dzeroski, Saso. „Ensembles of Multi-Objective Decision Trees.“ In: *ECML*. Ed. by Kok, Joost N., Koronacki, Jacek, Mántaras, Ramon López de, et al. Vol. 4701. Lecture Notes in Computer Science. Springer, 2007, pp. 624–631 (cit. on p. 103).

[Koj+10] Kojima, Kaname, Perrier, Eric, Imoto, Seiya, and Miyano, Satoru. „Optimal Search on Clustered Structural Constraint for Learning Bayesian Network Structure." In: *Journal of Machine Learning Research* 11 (2010), pp. 285–310 (cit. on pp. 81, 100).

[Kol63] Kolmogorov, Andreï Nikolaïevitch. „On Tables of Random Numbers". In: *Sankhya: The Indian Journal of Statistics, Series A (1961-2002)* 25.4 (1963), pp. 369–376 (cit. on p. 78).

[Kos02] Koster, Jan T.A. „Marginalizing and conditioning in graphical models". In: *Bernoulli* 8.6 (Dec. 2002), pp. 817–840 (cit. on pp. 57, 59, 64).

[KS04] Koivisto, Mikko and Sood, Kismat. „Exact Bayesian Structure Discovery in Bayesian Networks." In: *Journal of Machine Learning Research* 5 (2004), pp. 549–573 (cit. on p. 81).

[KS96] Koller, Daphne and Sahami, Mehran. „Toward Optimal Feature Selection." In: *ICML*. Ed. by Saitta, Lorenza. Morgan Kaufmann, 1996, pp. 284–292 (cit. on pp. 143, 151).

[KT05] Kang, Changsung and Tian, Jin. „Local Markov Property for Models Satisfying Composition Axiom." In: *UAI*. AUAI Press, 2005, pp. 284–291 (cit. on p. 155).

[Kul68] Kullback, Solomon. *Information theory and statistics*. Dover Publications, 1968 (cit. on p. 83).

[Kum+12] Kumar, Abhishek, Vembu, Shankar, Menon, Aditya Krishna, and Elkan, Charles. „Learning and Inference in Probabilistic Classifier Chains with Beam Search." In: *ECML/PKDD (1)*. Ed. by Flach, Peter A., Bie, Tijl De, and Cristianini, Nello. Vol. 7523. Lecture Notes in Computer Science. Springer, 2012, pp. 665–680 (cit. on p. 126).

[Lar+96] Larrañaga, Pedro, Poza, Mikel, Yurramendi, Yosu, Murga, Roberto H., and Kuijpers, Cindy M. H. „Structure Learning of Bayesian Networks by Genetic Algorithms: A Performance Analysis of Control Parameters." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18.9 (1996), pp. 912–926 (cit. on p. 81).

[Lau+90] Lauritzen, Steffen L., Dawid, A. Philip, Larsen, B. N., and Leimer, Hanns-Georg. „Independence properties of directed markov fields." In: *Networks* 20.5 (1990), pp. 491–505 (cit. on p. 30).

[Lau96] Lauritzen, Steffen L. *Graphical Models*. Oxford University Press, 1996 (cit. on p. 20).

[LBH15] LeCun, Yann, Bengio, Yoshua, and Hinton, Geoffrey. „Deep learning." In: *Nature* 521.7553 (2015), pp. 436–444 (cit. on p. 125).

[LC15] Liu, Shuhua Monica and Chen, Jiun-Hung. „A multi-label classification based approach for sentiment classification." In: *Expert Systems with Applications* 42.3 (2015), pp. 1083–1093 (cit. on p. 103).

[Li+15] Li, Ke, Liu, Yi, Wang, Quanxin, et al. „A Spacecraft Electrical Characteristics Multi-Label Classification Method Based on Off-Line FCM Clustering and On-Line WPSVM". In: *PLoS ONE* 10.11 (Nov. 2015), pp. 1–16 (cit. on p. 103).

[Li+16] Li, Cheng, Wang, Bingyu, Pavlu, Virgil, and Aslam, Javed A. „Conditional Bernoulli Mixtures for Multi-label Classification.“ In: *ICML*. Ed. by Balcan, Maria-Florina and Weinberger, Kilian Q. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 2482–2491 (cit. on p. 138).

[Li08] Li, Sanjiang. „Causal models have no complete axiomatic characterization“. In: *CoRR* abs/0804.2401 (2008) (cit. on p. 33).

[LK13] Lee, Jae-Sung and Kim, Dae-Won. „Feature selection for multi-label classification using multivariate mutual information.“ In: *Pattern Recognition Letters* 34.3 (2013), pp. 349–357 (cit. on p. 151).

[LK15] Lee, Jae-Sung and Kim, Dae-Won. „Fast multi-label feature selection based on information-theoretic feature ranking.“ In: *Pattern Recognition* 48.9 (2015), pp. 2761–2771 (cit. on p. 151).

[LKC15] Lotter, William, Kreiman, Gabriel, and Cox, David R. „Unsupervised Learning of Visual Structure using Predictive Generative Networks.“ In: *CoRR* abs/1511.06380 (2015) (cit. on p. 118).

[LMP01] Lafferty, John D., McCallum, Andrew, and Pereira, Fernando C. N. „Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.“ In: *ICML*. Ed. by Brodley, Carla E. and Danyluk, Andrea Pohoreckyj. Morgan Kaufmann, 2001, pp. 282–289 (cit. on pp. 129, 132).

[LMY12] Liu, Zhifa, Malone, Brandon M., and Yuan, Changhe. „Empirical evaluation of scoring functions for Bayesian network model selection.“ In: *BMC Bioinformatics* 13.S-15 (2012), S14 (cit. on p. 81).

[LPM01] Levitz, Michael, Perlman, Michael D., and Madigan, David. „Separation and Completeness Properties for Amp Chain Graph Markov Models“. In: *The Annals of Statistics* 29.6 (Dec. 2001), pp. 1751–1784 (cit. on p. 46).

[LT16] Lin, Henry W. and Tegmark, Max. „Why does deep and cheap learning work so well?“ In: *CoRR* abs/1608.08225 (2016) (cit. on p. 125).

[Lua+12] Luaces, Oscar, Díez, Jorge, Barranquero, José, Coz, Juan José del, and Bahamonde, Antonio. „Binary relevance efficacy for multilabel classification.“ In: *Progress in Artificial Intelligence* 1.4 (2012), pp. 303–313 (cit. on pp. 103, 119, 161, 181).

[LW02] Liaw, Andy and Wiener, Matthew. „Classification and Regression by randomForest“. In: *R News* 2.3 (2002), pp. 18–22 (cit. on p. 162).

[LW89] Lauritzen, Steffen L. and Wermuth, N. „Graphical Models for Associations between Variables, some of which are Qualitative and some Quantitative“. In: *The Annals of Statistics* 17.1 (Mar. 1989), pp. 31–57 (cit. on p. 44).

[MA10a] Morais, Sergio Rodrigues de and Aussem, Alex. „A novel Markov boundary based feature subset selection algorithm.“ In: *Neurocomputing* 73.4-6 (2010), pp. 578–584 (cit. on p. 157).

[MA10b] Morais, Sergio Rodrigues de and Aussem, Alex. „An Efficient and Scalable Algorithm for Local Bayesian Network Structure Discovery.“ In: *ECML/PKDD (3)*. Ed. by Balcázar, José L., Bonchi, Francesco, Gionis, Aristides, and Sebag, Michèle. Vol. 6323. Lecture Notes in Computer Science. Springer, 2010, pp. 164–179 (cit. on pp. 89, 90, 94).

[Mee97] Meek, Christopher. „Graphical models: selecting causal and statistical models“. PhD thesis. Carnegie Mellon University, 1997 (cit. on p. 81).

[MJM15] Malone, Brandon, Järvisalo, Matti, and Myllymäki, Petri. „Impact of Learning Strategies on the Quality of Bayesian Networks: An Empirical Evaluation.“ In: *UAI*. Ed. by Meila, Marina and Heskes, Tom. AUAI Press, 2015, pp. 562–571 (cit. on p. 81).

[Mou74] Moussouris, John. „Gibbs and Markov random systems with constraints“. English. In: *Journal of Statistical Physics* 10.1 (1974), pp. 11–33 (cit. on p. 23).

[MR98] Maron, Oded and Ratan, Aparna Lakshmi. „Multiple-Instance Learning for Natural Scene Classification.“ In: *ICML*. Ed. by Shavlik, Jude W. Morgan Kaufmann, 1998, pp. 341–349 (cit. on p. 164).

[MT99] Margaritis, Dimitris and Thrun, Sebastian. „Bayesian Network Induction via Local Neighborhoods.“ In: *NIPS*. Ed. by Solla, Sara A., Leen, Todd K., and Müller, Klaus-Robert. The MIT Press, 1999, pp. 505–511 (cit. on pp. 156, 157).

[MW03] Moore, Andrew W. and Wong, Weng-Keen. „Optimal Reinsertion: A New Search Operator for Accelerated and More Accurate Bayesian Network Structure Learning.“ In: *ICML*. Ed. by Fawcett, Tom and Mishra, Nina. AAAI Press, 2003, pp. 552–559 (cit. on p. 93).

[NKP03] Nielsen, Jens Dalgaard, Kocka, Tomás, and Peña, José M. „On Local Optima in Learning Bayesian Networks.“ In: *UAI*. Ed. by Meek, Christopher and Kjærulff, Uffe. Morgan Kaufmann, 2003, pp. 435–442 (cit. on p. 82).

[OIM04] Ott, Sascha, Imoto, Seiya, and Miyano, Satoru. „Finding Optimal Models for Small Gene Networks.“ In: *Pacific Symposium on Biocomputing*. Ed. by Altman, Russ B., Dunker, A. Keith, Hunter, Lawrence, Jung, Tiffany A., and Klein, Teri E. World Scientific, 2004, pp. 557–567 (cit. on p. 81).

[PBT05] Peña, José M., Björkegren, Johan, and Tegnér, Jesper. „Scalable, Efficient and Correct Learning of Markov Boundaries Under the Faithfulness Assumption.“ In: *ECSQARU*. Ed. by Godo, Lluis. Vol. 3571. Lecture Notes in Computer Science. Springer, 2005, pp. 136–147 (cit. on p. 87).

[PC10] Petterson, James and Caetano, Tibério S. „Reverse Multi-Label Learning.“ In: *NIPS*. Ed. by Lafferty, John D., Williams, Christopher K. I., Shawe-Taylor, John, Zemel, Richard S., and Culotta, Aron. Curran Associates, Inc., 2010, pp. 1912–1920 (cit. on p. 107).

[PD11] Poon, Hoifung and Domingos, Pedro M. „Sum-Product Networks: A New Deep Architecture.“ In: *UAI*. Ed. by Cozman, Fábio Gagliardi and Pfeffer, Avi. AUAI Press, 2011, pp. 337–346 (cit. on pp. 67, 132, 134).

[Pea09] Pearl, Judea. *Causality: Models, Reasoning and Inference*. 2nd ed. Cambridge University Press, Sept. 2009 (cit. on p. 37).

[Pea12] Pearl, Judea. „The Do-Calculus Revisited.“ In: *UAI*. Ed. by Freitas, Nando de and Murphy, Kevin P. AUAI Press, 2012, pp. 3–11 (cit. on p. 36).

[Pea85] Pearl, Judea. „Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning“. In: *Proceedings of the Cognitive Science Society (CSS-7)*. 1985 (cit. on p. 27).

[Pea89] Pearl, Judea. *Probabilistic reasoning in intelligent systems - networks of plausible inference.* Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, 1989, pp. I–XIX, 1–552 (cit. on pp. 14, 15, 27, 33, 144, 155).

[Pea95] Pearl, Judea. „Causal diagrams for empirical research“. In: *Biometrika* 82.4 (1995), pp. 669–710 (cit. on p. 36).

[Peh+15] Peharz, Robert, Tschiatschek, Sebastian, Pernkopf, Franz, and Domingos, Pedro M. „On Theoretical Properties of Sum-Product Networks.“ In: *AISTATS*. Ed. by Lebanon, Guy and Vishwanathan, S. V. N. Vol. 38. JMLR Workshop and Conference Proceedings. JMLR.org, 2015 (cit. on p. 132).

[Peh15] Peharz, Robert. „Foundations of Sum-Product Networks for Probabilistic Modeling“. PhD thesis. Graz University of Technology, 2015 (cit. on p. 134).

[Peñ+06] Peña, José M., Nilsson, Roland, Björkegren, Johan, and Tegnér, Jesper. „Identifying the Relevant Nodes Without Learning the Model.“ In: *UAI*. AUAI Press, 2006 (cit. on p. 155).

[Peñ+07] Peña, José M., Nilsson, Roland, Björkegren, Johan, and Tegnér, Jesper. „Towards scalable and data efficient learning of Markov boundaries.“ In: *International Journal of Approximate Reasoning* 45.2 (2007), pp. 211–232 (cit. on pp. 89, 96, 101, 144, 156, 157).

[Peñ08] Peña, Jose M. „Learning Gaussian Graphical Models of Gene Networks with False Discovery Rate Control.“ In: *EvoBIO*. Ed. by Marchiori, Elena and Moore, Jason H. Vol. 4973. Lecture Notes in Computer Science. Springer, 2008, pp. 165–176 (cit. on p. 95).

[Peñ14] Peña, Jose M. „Marginal AMP chain graphs.“ In: *International Journal of Approximate Reasoning* 55.5 (2014), pp. 1185–1206 (cit. on pp. 16, 61).

[Peñ15] Peña, Jose M. „Factorization, Inference and Parameter Learning in Discrete AMP Chain Graphs.“ In: *ECSQARU*. Ed. by Destercke, Sébastien and Denoeux, Thierry. Vol. 9161. Lecture Notes in Computer Science. Springer, 2015, pp. 335–345 (cit. on pp. 46, 47).

[PIM08] Perrier, Eric, Imoto, Seiya, and Miyano, Satoru. „Finding Optimal Bayesian Network Given a Super-Structure.“ In: *Journal of Machine Learning Research* 9 (2008), pp. 2251–2286 (cit. on pp. 94, 100).

[PP86] Pearl, Judea and Paz, Azaria. „Graphoids: Graph-Based Logic for Reasoning about Relevance Relations or When would x tell you more about y if you already know z?“ In: *ECAI*. 1986, pp. 357–363 (cit. on pp. 14, 24).

[PTT15] Papagiannopoulou, Christina, Tsoumakas, Grigorios, and Tsamardinos, Ioannis. „Discovering and Exploiting Deterministic Label Relationships in Multi-Label Learning.“ In: *KDD*. Ed. by Cao, Longbing, Zhang, Chengqi, Joachims, Thorsten, et al. ACM, 2015, pp. 915–924 (cit. on p. 156).

[PV87] Pearl, Judea and Verma, Thomas. „The Logic of Representing Dependencies by Directed Graphs.“ In: *AAAI*. Ed. by Forbus, Kenneth D. and Shrobe, Howard E. Morgan Kaufmann, 1987, pp. 374–379 (cit. on p. 14).

[R C16] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2016 (cit. on p. 95).

[Rea+09] Read, Jesse, Pfahringer, Bernhard, Holmes, Geoffrey, and Frank, Eibe. „Classifier Chains for Multi-label Classification.“ In: *ECML/PKDD*. Ed. by Buntine, Wray L., Grobelnik, Marko, Mladenic, Dunja, and Shawe-Taylor, John. Vol. 5782. Lecture Notes in Computer Science. Springer, 2009, pp. 254–269 (cit. on pp. 103, 120, 123, 167).

[Rea+11] Read, Jesse, Pfahringer, Bernhard, Holmes, Geoff, and Frank, Eibe. „Classifier chains for multi-label classification.“ In: *Machine Learning* 85.3 (2011), pp. 333–359 (cit. on p. 120).

[Rea+16] Read, Jesse, Reutemann, Peter, Pfahringer, Bernhard, and Holmes, Geoff. „MEKA: A Multi-label/Multi-target Extension to WEKA“. In: *Journal of Machine Learning Research* 17.21 (2016), pp. 1–5 (cit. on p. 167).

[RG16] Rahman, Tahrima and Gogate, Vibhav. „Merging Strategies for Sum-Product Networks: From Trees to Graphs“. In: *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, UAI 2016, June 25-29, Jersey City, New Jersey, USA*. 2016, pp. 617–626 (cit. on p. 132).

[Ric03] Richardson, Thomas. „Markov Properties for Acyclic Directed Mixed Graphs“. In: *Scandinavian Journal of Statistics* 30.1 (2003), pp. 145–157 (cit. on p. 63).

[Ris78] Rissanen, Jorma. „Modeling by shortest data description“. In: *Automatica* 14.5 (1978), pp. 465–471 (cit. on pp. 77, 80).

[Ris89] Rissanen, Jorma. „Stochastic Complexity in Statistical Inquiry“. In: *World Scientific, Series in Computer Science* 15 (1989) (cit. on p. 80).

[RMC15] Radford, Alec, Metz, Luke, and Chintala, Soumith. „Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks.“ In: *CoRR* abs/1511.06434 (2015) (cit. on p. 118).

[RML14] Read, Jesse, Martino, Luca, and Luengo, David. „Efficient monte carlo methods for multi-dimensional learning with classifier chains.“ In: *Pattern Recognition* 47.3 (2014), pp. 1535–1546 (cit. on p. 166).

[Rob73] Robinson, Robert W. „Counting Labeled Acyclic Digraphs.“ In: *New Directions in Graph Theory*. Ed. by Harary, F. New York: Academic Press, 1973 (cit. on p. 81).

[RS02] Richardson, Thomas and Spirtes, Peter. „Ancestral graph Markov models“. In: *Annals of Statistics* 30.4 (2002), pp. 962–1030 (cit. on pp. 43, 47, 48, 57–59).

[SA12] Scutari, Marco and Adriana, Brogini. „Bayesian Network Structure Learning with Permutation Tests“. In: *Communications in Statistics - Theory and Methods* 41.16-17 (2012), pp. 3233–3243 (cit. on p. 96).

[Sad11] Sadeghi, Kayvan. „Markov Equivalences for Subclasses of Loopless Mixed Graphs“. In: *ArXiv e-prints* (Oct. 2011). arXiv: `1110.4539 [stat.OT]` (cit. on p. 63).

[Sad12] Sadeghi, Kayvan. „Graphical representation of independence structures“. PhD thesis. University of Oxford, 2012 (cit. on p. 54).

[Sad13] Sadeghi, Kayvan. „Stable mixed graphs“. In: *Bernoulli* 19.5B (Nov. 2013), pp. 2330–2358 (cit. on p. 63).

[Sad16] Sadeghi, Kayvan. „Marginalization and conditioning for LWF chain graphs“. In: *The Annals of Statistics* 44.4 (Aug. 2016), pp. 1792–1816 (cit. on pp. 56, 59–61).

[SB15] Streich, Andreas P. and Buhmann, Joachim M. „Asymptotic analysis of estimators on multi-label data.“ In: *Machine Learning* 99.3 (2015), pp. 373–409 (cit. on p. 107).

[SB98] Studený, Milan and Bouckaert, Remco R. „On chain graph models for description of conditional independence structures“. In: *The Annals of Statistics* 26.4 (Aug. 1998), pp. 1434–1495 (cit. on p. 44).

[Sch78] Schwarz, Gideon. „Estimating the dimension of a model“. In: *The annals of statistics* 6 (1978), pp. 461–464 (cit. on p. 80).

[Scu10] Scutari, Marco. „Learning Bayesian Networks with the bnlearn R Package“. In: *Journal of Statistical Software* 35.3 (2010), pp. 1–22 (cit. on pp. 95, 156).

[SG91] Spirtes, Peter and Glymour, Clark. „An Algorithm for Fast Recovery of Sparse Causal Graphs.“ In: *Social Science Computer Review* 9.1 (1991), pp. 62–72 (cit. on pp. 86, 93).

[SGS90] Spirtes, Peter, Glymour, Clark, and Schienes, Richard. „Causality from Probability.“ In: *Evolving Knowledge in Natural and Artificial Intelligence*. Ed. by McKee, G. Pitman, 1990 (cit. on p. 85).

[SGS93] Spirtes, Peter, Glymour, Clark, and Schienes, Richard. *Causation, prediction, and search*. New York: Springer-Verlag, 1993 (cit. on pp. 86, 88, 91).

[Sim51] Simpson, Edward H. „The Interpretation of Interaction in Contingency Tables“. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 13.2 (1951), pp. 238–241 (cit. on p. 34).

[SK86] Speed, Terence P and Kiiveri, Harri T. „Gaussian Markov Distributions over Finite Graphs“. In: *The Annals of Statistics* 14.1 (Mar. 1986), pp. 138–150 (cit. on p. 42).

[SL14] Sadeghi, Kayvan and Lauritzen, Steffen. „Markov properties for mixed graphs“. In: *Bernoulli* 20.2 (May 2014), pp. 676–696 (cit. on pp. 14, 16, 43, 63).

[SL15] Sadeghi, Kayvan and Lauritzen, Steffen L. „Unifying Markov properties in graphical models“. In: Unpublished manuscript (2015) (cit. on pp. 41, 54, 62, 64).

[SLA13] Statnikov, Alexander R., Lemeire, Jan, and Aliferis, Constantin F. „Algorithms for discovery of multiple Markov boundaries.“ In: *Journal of Machine Learning Research* 14.1 (2013), pp. 499–566 (cit. on pp. 144, 147).

[SM05] Singh, Ajit P. and Moore, Andrew W. *Finding optimal Bayesian networks by dynamic programming*. Tech. rep. 2nd revision. 2005 (cit. on p. 81).

[SM06] Silander, Tomi and Myllymäki, Petri. „A Simple Approach for Finding the Globally Optimal Bayesian Network Structure.“ In: *UAI*. AUAI Press, 2006 (cit. on p. 81).

[SM12] Sutton, Charles A. and McCallum, Andrew. „An Introduction to Conditional Random Fields.“ In: *Foundations and Trends in Machine Learning* 4.4 (2012), pp. 267–373 (cit. on p. 132).

[SP15] Sonntag, Dag and Peña, Jose M. „Chain graph interpretations and their relations revisited.“ In: *International Journal of Approximate Reasoning* 58 (2015), pp. 39–56 (cit. on pp. 54, 63).

[Spo+13] Spolaôr, Newton, Cherman, Everton Alvares, Monard, Maria Carolina, and Lee, Huei Diana. „A Comparison of Multi-label Feature Selection Methods using the Problem Transformation Approach.“ In: *Electronic Notes in Theoretical Computer Science* 292 (2013), pp. 135–151 (cit. on p. 151).

[Spo80] Spohn, Wolfgang. „Stochastic independence, causal independence, and shieldability.“ In: *Journal of Philosophical Logic* 9.1 (1980), pp. 73–99 (cit. on p. 14).

[ST04] Smith, Noah and Tromble, Roy. *Sampling Uniformly from the Unit Simplex*. Tech. rep. Johns Hopkins University, 2004, pp. 1–6 (cit. on pp. 162, 179).

[Stu05] Studeny, Milan. *Probabilistic Conditional Independence Structures*. 1st ed. Springer, 2005 (cit. on pp. 3, 28, 65, 149, 187).

[Stu89] Studeny, Milan. „Multiinformation and the problem of characterization of conditional independence relations“. In: *Problems of Control and Information Theory* 1.18 (1989), pp. 3–16 (cit. on p. 15).

[Stu92] Studeny, Milan. „Conditional independence relations have no finite complete characterization.“ In: *Transactions of the 11th Prague Conference*. Ed. by Kubik, S. and Visek, J.A. Vol. B. Information Theory, Statistical Decision Functions and Random Processes. Kluwer, Dordrecht - Boston - London, 1992, pp. 377–396 (cit. on p. 15).

[Stu97] Studený, Milan. „A recovery algorithm for chain graphs.“ In: *International Journal of Approximate Reasoning* 17.2-3 (1997), pp. 265–293 (cit. on p. 45).

[Stu98] Studený, Milan. „Bayesian Networks from the Point of View of Chain Graphs.“ In: *UAI*. Ed. by Cooper, Gregory F. and Moral, Serafín. Morgan Kaufmann, 1998, pp. 496–503 (cit. on p. 45).

[Suc+14] Sucar, Luis Enrique, Bielza, Concha, Morales, Eduardo F., et al. „Multi-label classification with Bayesian network-based chain classifiers.“ In: *Pattern Recognition Letters* 41 (2014), pp. 14–22 (cit. on p. 120).

[Sul09] Sullivant, Seth. „Gaussian conditional independence relations have no finite complete characterization“. In: *Journal of Pure and Applied Algebra* 213.8 (2009). Theoretical Effectivity and Practical Effectivity of Gröbner Bases, pp. 1502–1506 (cit. on p. 15).

[SV93] Singh, Moninder and Valtorta, Marco. „An Algorithm for the Construction of Bayesian Network Structures from Data.“ In: *UAI*. Ed. by Heckerman, David and Mamdani, E. H. Morgan Kaufmann, 1993, pp. 259–265 (cit. on p. 92).

[TAS03a] Tsamardinos, Ioannis, Aliferis, Constantin F., and Statnikov, Alexander R. „Algorithms for Large Scale Markov Blanket Discovery.“ In: *FLAIRS Conference*. Ed. by Russell, Ingrid and Haller, Susan M. AAAI Press, 2003, pp. 376–381 (cit. on pp. 89, 157).

[TAS03b] Tsamardinos, Ioannis, Aliferis, Constantin F., and Statnikov, Alexander R. „Time and sample efficient discovery of Markov blankets and direct causal relations.“ In: *KDD*. Ed. by Getoor, Lise, Senator, Ted E., Domingos, Pedro M., and Faloutsos, Christos. ACM, 2003, pp. 673–678 (cit. on pp. 86, 87, 157).

[TB10] Tsamardinos, Ioannis and Borboudakis, Giorgos. „Permutation Testing Improves Bayesian Network Learning.“ In: *ECML/PKDD*. Ed. by Balcázar, José L., Bonchi, Francesco, Gionis, Aristides, and Sebag, Michèle. Vol. 6323. Lecture Notes in Computer Science. Springer, 2010, pp. 322–337 (cit. on p. 156).

[TBA06] Tsamardinos, Ioannis, Brown, Laura E., and Aliferis, Constantin F. „The max-min hill-climbing Bayesian network structure learning algorithm.“ In: *Machine Learning* 65.1 (2006), pp. 31–78 (cit. on pp. 71, 87, 92, 93, 100).

[TK07] Tsoumakas, Grigorios and Katakis, Ioannis. „Multi-Label Classification: An Overview.“ In: *IJDWM* 3.3 (2007), pp. 1–13 (cit. on p. 118).

[TKV08] Tsoumakas, Grigorios, Katakis, Ioannis, and Vlahavas, Ioannis. „Effective and Efficient Multilabel Classification in Domains with Large Number of Labels“. In: *Proceedings of the ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*. Antwerp, Belgium, 2008 (cit. on p. 166).

[TKV10] Tsoumakas, Grigorios, Katakis, Ioannis, and Vlahavas, Ioannis P. „Mining Multi-label Data.“ In: *Data Mining and Knowledge Discovery Handbook*. Ed. by Maimon, Oded and Rokach, Lior. Springer, 2010, pp. 667–685 (cit. on pp. 107, 168).

[TKV11] Tsoumakas, Grigorios, Katakis, Ioannis, and Vlahavas, Ioannis P. „Random k-Labelsets for Multilabel Classification.“ In: *IEEE Transactions on Knowledge and Data Engineering* 23.7 (2011), pp. 1079–1089 (cit. on pp. 123, 161, 165).

[Tso+05] Tsochantaridis, Ioannis, Joachims, Thorsten, Hofmann, Thomas, and Altun, Yasemin. „Large Margin Methods for Structured and Interdependent Output Variables.“ In: *Journal of Machine Learning Research* 6 (2005), pp. 1453–1484 (cit. on p. 132).

[Tso+11] Tsoumakas, Grigorios, Spyromitros-Xioufis, Eleftherios, Vilcek, Jozef, and Vlahavas, Ioannis. „Mulan: A Java Library for Multi-Label Learning“. In: *Journal of Machine Learning Research* 12 (2011), pp. 2411–2414 (cit. on p. 167).

[TV07] Tsoumakas, Grigorios and Vlahavas, Ioannis P. „Random k -Labelsets: An Ensemble Method for Multilabel Classification.“ In: *ECML*. Ed. by Kok, Joost N., Koronacki, Jacek, Mántaras, Ramon López de, et al. Vol. 4701. Lecture Notes in Computer Science. Springer, 2007, pp. 406–417 (cit. on pp. 103, 123, 161, 165).

[Van79] Van Rijsbergen, Cornelis Joost Keith. *Information Retrieval (2nd edition)*. Butterworths, London, 1979 (cit. on p. 110).

[VM12] Villanueva, Edwin and Maciel, Carlos Dias. „Optimized Algorithm for Learning Bayesian Network Super-structures.“ In: *ICPRAM (1)*. Ed. by Carmona, Pedro Latorre, Sánchez, J. Salvador, and Fred, Ana L. N. SciTePress, 2012, pp. 217–222 (cit. on p. 94).

[VP88] Verma, Thomas and Pearl, Judea. „Causal networks: semantics and expressiveness.“ In: *UAI*. Ed. by Shachter, Ross D., Levitt, Tod S., Kanal, Laveen N., and Lemmer, John F. North-Holland, 1988, pp. 69–78 (cit. on pp. 27, 40, 43, 54).

[VP90] Verma, Thomas and Pearl, Judea. „Equivalence and synthesis of causal models.“ In: *UAI*. Ed. by Bonissone, Piero P., Henrion, Max, Kanal, Laveen N., and Lemmer, John F. Elsevier, 1990, pp. 255–270 (cit. on pp. 40, 57, 84, 85).

[VR02] Venables, William N. and Ripley, Brian D. *Modern Applied Statistics with S*. Fourth. ISBN 0-387-95457-0. New York: Springer, 2002 (cit. on p. 179).

[Waa09] Waal, Peter R. de. „Marginals of DAG-Isomorphic Independence Models." In: *ECSQARU*. Ed. by Sossai, Claudio and Chemello, Gaetano. Vol. 5590. Lecture Notes in Computer Science. Springer, 2009, pp. 192–203 (cit. on p. 7).

[Wae+14] Waegeman, Willem, Dembczynski, Krzysztof, Jachnik, Arkadiusz, Cheng, Weiwei, and Hüllermeier, Eyke. „On the bayes-optimality of F-measure maximizers." In: *Journal of Machine Learning Research* 15.1 (2014), pp. 3333–3388 (cit. on pp. 111, 171–173).

[Wan+14] Wang, Shangfei, Wang, Jun, Wang, Zhaoyu, and Ji, Qiang. „Enhancing multi-label classification by modeling dependencies among labels." In: *Pattern Recognition* 47.10 (2014), pp. 3405–3413 (cit. on pp. 103, 122).

[Wan+15] Wang, Ling, Zhou, Tie Hua, Lee, Yang Koo, Cheoi, Kyung-Joo, and Ryu, Keun Ho. „An efficient refinement algorithm for multi-label image annotation with correlation model." In: *Telecommunication Systems* 60.2 (2015), pp. 285–301 (cit. on p. 103).

[WB09] Wang, Zhou and Bovik, Alan C. „Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures". In: *Signal Processing Magazine, IEEE* 26.1 (2009), pp. 98–117 (cit. on p. 118).

[WCP94] Wermuth, Nanny, Cox, David R., and Pearl, Judea. „Explanations for multivariate structures derived from univariate recursive regressions". In: *Ber. Stoch. Verw. Geb.* (1994) (cit. on p. 57).

[Wer11] Wermuth, Nanny. „Probability distributions with summary graph structure". In: *Bernoulli* 17.3 (Aug. 2011), pp. 845–879 (cit. on p. 57).

[WHZ14] Wu, Jian-Sheng, Huang, Sheng-Jun, and Zhou, Zhi-Hua. „Genome-Wide Protein Function Prediction through Multi-Instance Multi-Label Learning." In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 11.5 (2014), pp. 891–902 (cit. on p. 103).

[WMC09] Wermuth, Nanny, Marchetti, Giovanni M., and Cox, David R. „Triangular systems for symmetric binary variables". In: *Electronic Journal of Statistics* 3 (2009), pp. 932–955 (cit. on p. 15).

[Wri20] Wright, Sewall. „The Relative Importance of Heredity and Environment in Determining the Piebald Pattern of Guinea-Pigs". In: *Proceedings of the National Academy of Sciences of the United States of America* 6.6 (1920), pp. 320–332 (cit. on p. 27).

[Wri34] Wright, Sewall. „The Method of Path Coefficients". In: *The Annals of Mathematical Statistics* 5.3 (Sept. 1934), pp. 161–215 (cit. on p. 27).

[WWL02] Wong, S. K. Michael, Wu, Dan, and Lin, Tao. „A Structural Characterization of DAG-Isomorphic Dependency Models." In: *Canadian Conference on AI*. Ed. by Cohen, Robin and Spencer, Bruce. Vol. 2338. Lecture Notes in Computer Science. Springer, 2002, pp. 195–209 (cit. on p. 33).

[Ye+12] Ye, Nan, Chai, Kian Ming Adam, Lee, Wee Sun, and Chieu, Hai Leong. „Optimizing F-measure: A Tale of Two Approaches." In: *ICML*. icml.cc / Omnipress, 2012 (cit. on pp. 171, 182).

[YM13] Yuan, Changhe and Malone, Brandon M. „Learning Optimal Bayesian Networks: A Shortest Path Perspective.“ In: *Journal of Artificial Intelligence Research* 48 (2013), pp. 23–65 (cit. on p. 81).

[YMW11] Yuan, Changhe, Malone, Brandon M., and Wu, XiaoJian. „Learning Optimal Bayesian Networks Using A* Search.“ In: *IJCAI*. Ed. by Walsh, Toby. IJCAI/AAAI, 2011, pp. 2186–2191 (cit. on p. 81).

[Zar+11] Zaragoza, Julio H., Sucar, Luis Enrique, Morales, Eduardo F., Bielza, Concha, and Larrañaga, Pedro. „Bayesian Chain Classifiers for Multidimensional Classification.“ In: *IJCAI*. Ed. by Walsh, Toby. IJCAI/AAAI, 2011, pp. 2192–2197 (cit. on p. 120).

[Zha+15a] Zhang, Wen, Liu, Feng, Luo, Longqiang, and Zhang, Jingxia. „Predicting drug side effects by multi-label learning and ensemble learning.“ In: *BMC Bioinformatics* 16 (2015), p. 365 (cit. on p. 103).

[Zha+15b] Zhao, Kaili, Zhang, Honggang, Ma, Zhanyu, Song, Yi-Zhe, and Guo, Jun. „Multi-label learning with prior knowledge for facial expression analysis.“ In: *Neurocomputing* 157 (2015), pp. 280–289 (cit. on p. 103).

[Zha+16] Zhao, Han, Adel, Tameem, Gordon, Geoff, and Amos, Brandon. „Collapsed Variational Inference for Sum-Product Networks.“ In: *ICML*. Ed. by Balcan, Maria-Florina and Weinberger, Kilian Q. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 1310–1318 (cit. on p. 133).

[ZMP15] Zhao, Han, Melibari, Mazen, and Poupart, Pascal. „On the Relationship between Sum-Product Networks and Bayesian Networks.“ In: *ICML*. Ed. by Bach, Francis R. and Blei, David M. Vol. 37. JMLR Proceedings. JMLR.org, 2015, pp. 116–124 (cit. on pp. 67, 132, 135).

[Zuf+15] Zufferey, Damien, Hofer, Thomas, Hennebert, Jean, et al. „Performance comparison of multi-label learning algorithms on clinical data for chronic diseases.“ In: *Computers in Biology and Medicine* 65 (2015), pp. 34–43 (cit. on p. 103).

[ZZ07] Zhang, Min-Ling and Zhou, Zhi-Hua. „ML-KNN: A lazy learning approach to multi-label learning.“ In: *Pattern Recognition* 40.7 (2007), pp. 2038–2048 (cit. on p. 123).

[ZZ10] Zhang, Min-Ling and Zhang, Kun. „Multi-label learning by exploiting label dependency.“ In: *KDD*. Ed. by Rao, Bharat, Krishnapuram, Balaji, Tomkins, Andrew, and 0001, Qiang Yang. ACM, 2010, pp. 999–1008 (cit. on pp. 103, 121, 166).

[ZZ14] Zhang, Min-Ling and Zhou, Zhi-Hua. „A Review on Multi-Label Learning Algorithms.“ In: *IEEE Transactions on Knowledge and Data Engineering* 26.8 (2014), pp. 1819–1837 (cit. on p. 110).

Author's publications

Refereed Journal Papers

[GAE14] Gasse, Maxime, Aussem, Alex, and Elghazel, Haytham. „A hybrid algorithm for Bayesian network structure learning with application to multi-label learning." In: *Expert Systems with Applications* 41.15 (2014), pp. 6755–6772 (cit. on pp. 1, 71, 93, 94, 143, 153).

Refereed Conference Papers

[GA16a] Gasse, Maxime and Aussem, Alex. „F-Measure Maximization in Multi-Label Classification with Conditionally Independent Label Subsets." In: *ECML/PKDD (1)*. Ed. by Frasconi, Paolo, Landwehr, Niels, Manco, Giuseppe, and Vreeken, Jilles. Vol. 9851. Lecture Notes in Computer Science. Springer, 2016, pp. 619–631 (cit. on pp. 1, 141, 171).

[GA16b] Gasse, Maxime and Aussem, Alex. „Identifying the irreducible disjoint factors of a multivariate probability distribution." In: *PGM*. Ed. by Antonucci, Alessandro, Corani, Giorgio, and de Campos, Cassio Polpo. Vol. 52. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 183–194 (cit. on p. 1).

[GAE15] Gasse, Maxime, Aussem, Alex, and Elghazel, Haytham. „On the Optimality of Multi-Label Classification under Subset Zero-One Loss for Distributions Satisfying the Composition Property." In: *ICML*. Ed. by Bach, Francis R. and Blei, David M. Vol. 37. JMLR Proceedings. JMLR.org, 2015, pp. 2531–2539 (cit. on pp. 1, 141, 154).

[Aus+14] Aussem, Alex, Caillet, Pascal, Klemm, Zara, et al. „Analysis of risk factors of hip fracture with causal Bayesian networks." In: *IWBBIO*. Ed. by Rojas, Ignacio and Guzman, Francisco M. Ortuño. Copicentro Editorial, 2014, pp. 1074–1085.

[Le +14] Le Goff, Ronan, Garcia, David, Gasse, Maxime, and Aussem, Alex. „Optimal Sensor Locations for Polymer Injection Molding Process". In: *ESAFORM*. Vol. 611. Key Engineering Materials. Trans Tech Publications, July 2014, pp. 1724–1733.

[GAE12] Gasse, Maxime, Aussem, Alex, and Elghazel, Haytham. „An Experimental Comparison of Hybrid Algorithms for Bayesian Network Structure Learning." In: *ECML/PKDD*. Ed. by Flach, Peter A., Bie, Tijl De, and Cristianini, Nello. Vol. 7523. Lecture Notes in Computer Science. Springer, 2012, pp. 58–73 (cit. on pp. 1, 71, 93, 94).