Distance induction in first order logic

Michèle Sebag

Distance Induction in First Order Logic

Michèle Sebag

LMS - CNRS ura 317
Ecole Polytechnique, 91128 Palaiseau

LRI - CNRS ura 410
Université Paris-Sud, 91405 Orsay

Michele.Sebag@polytechnique.fr

Abstract. A distance on the problem domain allows one to tackle some typical goals of machine learning, e.g. classification or conceptual clustering, via robust data analysis algorithms (e.g. k-nearest neighbors or k-means).

A method for building a distance on first-order logic domains is presented in this paper. The distance is constructed from examples expressed as definite or constrained clauses, via a two-step process: a set of d hypotheses is first learnt from the training examples. These hypotheses serve as new descriptors of the problem domain $\mathcal{L}_h$: they induce a mapping π from $\mathcal{L}_h$ onto the space of integers $\mathbb{N}^d$. The distance between any two examples E and F is finally defined as the Euclidean distance between $\pi(E)$ and $\pi(F)$. The granularity of this hypothesis-driven distance (HDD) is controlled via the user-supplied parameter d.

The relevance of a HDD is evaluated from the predictive accuracy of the k-NN classifier based on this distance. Preliminary experiments demonstrate the potentialities of distance induction, in terms of predictive accuracy, computational cost, and tolerance to noise.

1 Introduction

The expert indeed knows to which extent any two examples or hypotheses on a problem domain, are similar: a relevant distance indeed represents a powerful, even if implicit, background knowledge. Distances can support many machine learning tasks:

- A distance or similarity function is needed to cluster the examples, which is the core of unsupervised learning [9, 4]. Clustering also constitutes a main stage of knowledge discovery in databases (KDD) [8]: one must somehow divide the enormous amount of available data, in order for knowledge to be conquered. Inductive logic programming (ILP) [15] can benefit from clustering, too: e.g. *KBG* uses a similarity function specifically designed for first-order languages, and gradually constructs hypotheses by generalizing the most similar examples and/or hypotheses [2].
- A distance allows the retrieval of the examples or hypotheses most similar to the instance at hand. In case-based reasoning (CBR), the retrieval stage commands the success of the whole process; hence much attention has been paid in CBR to developing flexible distances or similarity functions on structured domains [1]. Retrieving the nearest neighbors of the instance at hand also constitutes the core of instance-based learning. The ILP system *RIBL* [7] consists

of a k-NN classifier relying on an extended version of the first-order distance of *KBG*.
A fruitful combination of inductive learning and k-NN classifier in attribute-value domains is described in [5]: *RISE* uses as default rule the majority vote of the k rules whose hypotheses are the closest to the instance at hand [5].
• In the field of analogy, one looks for "optimal" mappings from the source onto the target context; the optimality criterion most often refers to a relational or structural distance [10, 3].

In this paper, we first compare the respective advantages and weaknesses of rules and distances in regard to supervised learning. We then discuss previous work devoted to constructing distances on first-order languages [2, 7]. Section 3 presents an alternative to distances based on syntax and weights, namely *hypothesis-driven distances* (HDD). We show that a set of d hypotheses induces a mapping π from the problem domain $\mathcal{L}_h$ onto the space of vectors of integers $\mathbb{N}^d$. A distance on $\mathcal{L}_h$ then follows, by defining the distance between two any examples or further hypotheses E and F as the Euclidean distance between $\pi(E)$ and $\pi(F)$. The properties and biases of HDDs are studied.
DISTILL (for *Distance Induction with* STILL) uses the ILP system *STILL* [18] to construct rather blindly d hypotheses, where d is supplied by the user. These hypotheses only serve here as system of coordinates: further examples or hypotheses are given a numerical description within this system. *DISTILL* finally computes the distance between any two examples with same polynomial complexity as in *STILL* (section 4).
This approach is validated on the mutagenesis problem: the 1-NN classifier based on the distance constructed by *DISTILL*, demonstrates to be quite competitive with respect to prominent ILP learners such as *FOIL* [16] and *PROGOL* [14] on this problem. *DISTILL* also improves on *STILL* [18]: it involves one less parameter and shows little sensitivity with respect to parameter d for $d \geq 30$.
We last conclude with some perspectives for further research.

2 State of the art

This section first presents our motivation for constructing distances on first-order logic space, and briefly recalls some previous work devoted to this aim.

2.1 Rules *versus* Distances

The main advantages of instance-based (e.g. k-NN) classifiers versus standard rule learning are extensively discussed in [7]: simply put, k-NN classifiers accurately deal with both symbolic and numerical data, on one hand, and with noisy data, on the other hand. Further, the predictive accuracy obtained by a k-NN classifier (in leave-one-out evaluation mode) gives hints into the quality of the data, and derives lower bounds on the optimal predictive accuracy [6].

Practically, a k-NN classifier allows for a flexible modeling of the target concept, more easily than rules or even oblique decision trees [11]. This can be exemplified as follows: in the bidimensional space $\mathbb{R}^2$, a set of n rules characterizes the target concept as the union of n rectangles; an oblique decision tree with n leaves characterizes it as the union of n polygons. And a set of N examples, plus a distance, induces a fine grained partition of the problem domain into N cells (the Voronoï cells); the target concept is characterized as the union of those cells that are centered on a positive example.

Compared to rules, instance-based classifiers suffer from their low intelligibility: the classification of an instance is justified by exhibiting the most similar example(s), rather than a high-level hypothesis.

2.2 Related work

Most distances on attribute-value languages are computed as the weighted sum of the elementary distances d_i defined on the attribute domains:

given $E = \wedge_i[att_i = V_i]$ and $F = \wedge_i[att_i = W_i]$, $d(E,F) = \sum_i w_i d_i(V_i, W_i)$

The distance accuracy (evaluated as the predictive accuracy of the corresponding k-NN classifier) critically depends on weights w_i, usually adjusted by trial and error. These can also be determined by an optimization algorithm [12].

Weight-based distances have been first extended to first-order logic languages in [2] and later refined in [7]. In both cases, the distance between any two conjunctive formulae is basically computed from that of their literals; the distance between two literals (built on the same symbol of predicate) is computed from the distance between their arguments, the weight of the predicate, and the weights of the predicate arguments. A global perspective on the examples, accounting for the semantics of the domain, is offered by computing the distance between two terms from the distance between the literals where they both appear. (Combinatorial explosion is prevented via syntactic restrictions on the literals examined). In *KBG* [2], the distances between terms are computed via a fixed point method, whereas *RIBL* [7] uses an iterative resolution.
The resulting similarity map critically depends on both the syntax and the weights. This limitation is partly addressed by *RIBL*, which iteratively refines the weights proposed by the expert.

To sum up, these distances combine built-in knowledge (the elementary distances on the domains of attributes or predicate arguments), with weights, i.e. non-declarative biases either manually or automatically adjusted.

3 Hypothesis-driven distances

This section investigates how a set of hypotheses can be used to map a problem domain onto a metric space. The properties and limitations of the distance constructed from this mapping, or hypothesis-driven distance (HDD), are studied.

3.1 Principle

Let $\mathcal{L}_h$ denote the language of hypotheses (including the language of instances via the single representation trick). Let $\mathcal{H} = \{h_1, \dots h_d\}$ denote a set of d hypotheses. One notices [19] that $\mathcal{H}$ induces a mapping π from $\mathcal{L}_h$ onto the boolean space of dimension d, by associating to any example or hypothesis E the vector of booleans coding whether E is subsumed by h_i, noted $E \prec h_i$:

$$\begin{array}{l} \pi : \mathcal{L}_h \rightarrow \{0,1\}^d \\ \qquad E \rightarrow \pi(E) = (\pi_1(E), \dots, \pi_d(E)), \qquad \text{where } \pi_i(E) = 1 \text{ iff } E \prec h_i \end{array}$$

Note that this projection onto $\{0,1\}^d$ does not make any assumption on $\mathcal{L}_h$: besides $\mathcal{H}$, it only invokes the covering test (checking whether $E \prec h_i$).

And $\{0,1\}^d$ is a metric space; a distance on $\mathcal{L}_h$ thus naturally follows, by setting:

$$\forall\, E, F \in \mathcal{L}_h,\ dist(E,F) = \sum_{i=1}^{d} |\pi_i(E) - \pi_i(F)|$$

By construction, *dist* is symmetrical and satisfies the triangular inequality:

$$\forall\, E, F, G,\ dist(E,F) \leq dist(E,G) + dist(G,F)$$

Still, it does not satisfy the identity relation[1]: $(dist(E,F) = 0) \not\Rightarrow (E = F)$.

3.2 Local behavior of HDD

Hypotheses-based distances locally depend upon the context. Consider examples E and F, together with the single hypothesis h (Table 1). As E is covered by h ($\pi(E) = 1$), and F is not ($\pi(F) = 0$), one has $dist(E,F) = 1$.

Table 1: Mapping based on hypothesis h = [Atom = carbon] ∧ [Type > 20]

	Initial description				*Mapping*
	Atom	*Size*	*Type*	*El. charge*	π
E	*carbon*	*small*	*22*	*3.45*	*1*
F	*carbon*	*large*	*17*	*5.22*	*0*

Consider examples E' and F' constructed from E and F via replacing a common feature ($Atom = carbon$) by another feature (say $Atom = oxygen$). Any weight-based distance $dist_w$ would give $dist_w(E,F) = dist_w(E',F')$. More generally, weight-based distances are invariant by translation (consistently modifying a feature shared by any two examples does not modify their distance).

This is not necessary the case for hypotheses-based distances, due to the fact that $\pi(E)$ globally depends on E (since $\pi(E') = \pi(F') = 0$, $dist(E',F') = 0$). A modification of any given feature of E may, or not, have an effect on $\pi(E)$ depending on the other features.

A hypothesis-driven distance thereby encodes local discontinuities of the problem domain, corresponding to the frontiers of hypotheses h_i.

[1] Properly speaking, *dist* is hence a semi-distance, rather than a distance. The distinction is omitted in what follows for the sake of simplicity.

The property of *non invariance by translation* is desirable as it enables to emulate the "versatile similarities" of experts. An expert may consider two devices manufactured by a given firm, as very similar; what s/he really means is that same failures are likely observed on these devices. But (rather unexpectedly for the naive knowledge engineer) the same devices manufactured by another firm, happen to be judged quite dissimilar...

3.3 Limitations of HDDs

HDDs do not present any interest whenever they are based on a concise set of hypotheses $\mathcal{H}$: e.g. *dist* gets rather coarse if any example is covered by a single hypothesis, such as happens if $\mathcal{H}$ is a decision tree (either E and F are covered by the same hypothesis, and $dist(E,F) = 0$, or $dist(E,F) = 2$).

The granularity of a HDD increases with the redundancy of $\mathcal{H}$ (i.e. the average number of h_i covering any example) and more precisely with the number and diversity of hypotheses h_i. Still, a HDD does not involve in any way the conclusions associated to hypotheses h_i; this suggests that the relevance of a HDD is potentially independent from the relevance of $\mathcal{H}$ (see section 4.3).

Still, the structure of the boolean space does not reflect the structure of the problem domain. A hypothesis h_i usually covers less than half the problem space: $\pi_i(E) = 1$ is thus less frequent than $\pi_i(E) = 0$, whilst 1 and 0 play equivalent roles in the boolean space.

3.4 Projection onto $\mathbb{N}^d$

We therefore consider more complex hypotheses. Let h_i now be a disjunction of formulae in $\mathcal{L}_h$, with $h_i = s_{i,1} \vee \ldots \vee s_{i,n_i}$, and let $\pi_i(E)$ (section 3.1) be now defined as the number of formulae $s_{i,j}$ covering E. This allows π to map the problem domain $\mathcal{L}_h$ onto a richer metric space, that of integer vectors $\mathbb{N}^d$. The corresponding HDD is naturally defined as:

$$dist(E,F) = \sqrt{\sum_i (\pi_i(E) - \pi_i(F))^2}$$

The ordered structure of $\mathbb{N}$ reflects a logical structure on the problem domain. Let h_i^M denote the $M - of - N$ hypothesis constructed from the disjunctive h_i, defined as: $E \prec h_i^M$ iff E is covered by at least M formulae $s_{i,j}$. One easily shows that h_i^{M+1} is covered by h_i^M. The set of hypotheses $\{h_i^M$, for $M = 1..n_i\}$, is a sequence of nested hypotheses which can be viewed as neighborhoods, or balls, of increasing specificity; π_i thereby corresponds to a "dimension" of the problem domain, and the coordinate $\pi_i(E)$ of E on this dimension precisely gives the rank of the most specific ball E belongs to.

4 Distance Induction based on Disjunctive Version Space

This section is devoted to learning a HDD from examples expressed as definite or constrained clauses.

4.1 Principle

The presented mechanism relies on the disjunctive version space (DiVS) approach; more details on *DiVS* in attribute-value and first-order logic languages are respectively found in [17] and [18]. The elementary step in *DiVS* consists of characterizing the most general hypothesis $D(E, F)$ covering example E and discriminating example F, where E and F satisfy distinct target concepts.

In attribute-value languages, $D(E, F)$ simply is the disjunction of the maximally general selectors[2] covering E and rejecting F:

Table 2: Hypothesis D(E,F) and corresponding mapping

	Initial description				*Mapping*
	Atom	*Size*	*Type*	*El. charge*	π
E	*carbon*	*small*	*22*	*3.45*	*3*
F	*carbon*	*large*	*17*	*5.22*	*0*
I	*oxygen*	*small*	*18*	*7.11*	*2*

$$D(E,F) = [Size = small] \vee [Type > 17] \vee [El.\ charge < 5.22]$$

Given the user-supplied number d of dimensions, $\mathcal{H}$ is iteratively constructed by setting $h_i = D(E_i, F_i)$, where E_i and F_i are randomly selected in the training set such that they satisfy distinct target concepts.

```
Construction of $\mathcal{H} = \{h_1, \ldots, h_d\}$
    For $i = 1$ to $d$,
        Randomly select $E_i$ and $F_i$ in the training set
            with $Class(E_i) \neq Class(F_i)$
        Construct $h_i$ discriminating $E_i$ from $F_i$.
```

For any further example I, the coordinate $\pi_i(I)$ on dimension $D(E_i, F_i)$ is computed as the number of selectors in $D(E_i, F_i)$, satisfied by I. E_i and F_i respectively get the highest and lowest coordinates on this dimension.

4.2 DISTILL

DiVS has been extended and adapted to first order logic via the *STILL* algorithm [18]. Due to space limitations, *STILL* will only be illustrated on a short example. Let E and F be definite clauses; let $\mathcal{C}$ be constructed from E by turning any occurence of a term t_i in E into a distinct variable X_j, and let substitution θ be defined as $\theta = \{X_j / t_i\}$.

$$\begin{array}{lll} E : tc(e) & : - & atom(e, a, oxy, 18), atom(e, b, carbon, 22), cc(a, b) \\ F : {}^{-}tc(f) & : - & atom(f, c, carbon, 24), atom(f, d, hydr, 3) \\ \mathcal{C} : \ tc(X) & : - & atom(X', Y, Z, T), atom(X'', U, V, W), cc(R, S) \end{array}$$

[2] We restrict ourselves to selectors $[att = V]$, where V denote a discrete value or a numerical interval. Selector $[att = (a, +\infty)]$ is written $[att > a]$ for the sake of convenience.

A constrained clause $G\gamma$ in the chosen language belongs to the set $D(E, F)$, iff either G or γ discriminate F. G is discriminant iff it includes a discriminant predicate (e.g. *cc*). Otherwise, G subsumes F and the set of substitutions mapping G onto F is denoted Σ; then, γ is discriminant iff it is incompatible with all substitutions in Σ, or equivalently belongs to all $D(\theta, \sigma)$ within an equivalent attribute-value representation:

Table 3: Attribute-value reformulation and (part of) a discriminant constraint

	X	X'	Y	Z	T	X''	U	V	W	$T-W$
θ	e	e	a	oxy	18	e	b	$carbon$	22	-4
σ_1	f	f	c	$carbon$	24	f	c	$carbon$	24	0
σ_2	f	f	d	$hydr$	3	f	d	$hydr$	3	0
σ_3	f	f	c	$carbon$	24	f	d	$hydr$	3	21
σ_4	f	f	d	$hydr$	3	f	c	$carbon$	24	-21

$$D(\theta, \sigma_1) = [Z = oxygen] \vee [T < 24] \vee [W < 24] \vee [T - W < 0]$$

The disjunctive hypothesis $D(E, F)$ discriminating E from F is therefore completely described by the set of discriminant predicates, and the disjunctive constraints $D(\theta, \sigma)$ for σ ranging in Σ. This characterization gets intractable on really relational domains (e.g. $|\Sigma|$ goes up to 40^{40} in the mutagenesis problem). *STILL* therefore constructs a polynomial approximation of $D(E, F)$, noted $D_\eta(E, F)$, by only considering η substitutions $\sigma_1, ..\sigma_\eta$ randomly sampled in Σ. The construction of $D_\eta(E, F)$ is in $\mathcal{O}(\eta \times V^2)$, where V denotes the maximal number of arguments in an example.

Deciding whether $D_\eta(E, F)$ covers a further instance I is similarly intractable, as it requires to explore the set Σ' of substitutions mapping C onto I. A polynomial approximation of the covering test is similarly provided by considering only K substitutions randomly selected in Σ'.

The coordinate of I on dimension $D_\eta(E, F)$ is the number of discriminant predicates involved in I, augmented with the maximal value of $C\tau \star D_\eta(E, F)$, taken over K substitutions τ randomly selected in Σ'. And $C\tau \star D_\eta(E, F)$ is the minimum number of selectors in $D(\theta, \sigma_j)$ satisfied by τ, for $j = 1 \ldots \eta$. Finally, the distance between any two examples has complexity $\mathcal{O}(d \times K \times \eta \times V^2)$.

4.3 Experimentation

This approach is evaluated on the well-studied mutagenesis problem [13, 21]. Table 4.(a) reports the best results obtained by *FOIL*, *PROGOL* and *STILL* [20, 18]. *FOIL* and *PROGOL* have been evaluated via 10-fold crossvalidation; *STILL* was evaluated in a similar way, only including 25 runs (with different random seeds) instead of 10, as recommended for evaluating stochastic processes. Run times (in seconds) are measured on HP-735 workstations.

DISTILL is evaluated from the average predictive accuracy of the 1-NN classifier based on *dist*, via the same protocol as *STILL*. The experiments focus on the influence of the number d of constructed hypotheses, varied in 10..100. The two other parameters of *DISTILL*, inherited from *STILL*, are set to their default value ($\eta = 300$ and $K = 3$).

Another experimentation goal is to study what happens if the provided examples are not classified at all, by removing the test $Class(E) \neq Class(F)$ in the construction of $\mathcal{H}$ (section 4.1). The corresponding algorithm is termed *UNDISTILL*, for *Unsupervised Distance Induction.*

Tables 4.b and 4.c respectively give the results obtained by *DISTILL* and *UNDISTILL* (with run times in seconds on a HP-710).

Table 4: Predictive accuracy on the 188-compound problem

(a) Reference results

System	Accuracy	Time
FOIL	86 ± 3	.5
PROGOL	88 ± 2	40 950
STILL	93.6 ± 4	< 120

(b) *DISTILL*

D	Accuracy	Time
10	88.6 ± 4.8	7
30	93.6 ± 5	19
50	94.7 ± 3.7	31
70	**96.7** ± 4.3	43
90	95.3 ± 2.4	56

(c) *UNDISTILL*

D	Accuracy	Time
10	86.7 ± 6.9	6
30	94.2 ± 3.8	19
50	93.3 ± 3.8	31
70	93.3 ± 5.3	44
90	**94.7** ± 2.6	56

It was conjectured that the relevance of $\mathcal{H}$ was not a necessary condition to derive a relevant HDD (section 3.3); one is nevertheless surprised that *DISTILL* and *UNDISTILL* obtain comparable results. In retrospect, it appears that hypotheses are used to make distinctions on the problem domain: the soundness of these distinctions does not matter provided they allow for a sufficiently precise scattering of the problem domain.

Practically, the good performances of UNDISTILL suggest that distance induction does not depend on the noise of the data, and can be employed for supervised learning.

5 Conclusion

Rather than syntactically comparing two examples, we propose to compare the way these respectively behave with respect to a set of hypotheses. Hypothesis-driven distances strongly depend on the selection of the hypotheses: HDDs typically bring no further information if these hypotheses are concise and intelligible (section 3.3). We therefore used a disjunctive version space approach: a set of d hypotheses is constructed as the maximally general hypotheses discriminating d pairs of examples (E_i, F_i). E_i and F_i are randomly selected in *UNDISTILL*, and they are further required to satisfy distinct target concepts in *DISTILL*.

Experimental validation shows that both *DISTILL* and *UNDISTILL* supersede other ILP learners on the mutagenesis dataset, for $d \geq 30$. Incidentally, this confirms that a stochastic bias (meant as the selection of E_i and F_i) can be a sound alternative to knowledge-demanding biases.

Further work will consider how the set of hypotheses can be pruned or augmented. Other perspectives are offered by coupling this distance with standard data analysis algorithms (e.g. k-means or factorial analysis) to achieve conceptual clustering or graphical representation of the data.

This approach will also be experimented on other and larger datasets, facing with the multiple challenges of knowledge discovery in data bases.

References

1. A. Aamodt and E.Plaza. Case-based reasoning : Foundational issues, methodological variations, and system approaches. *AICOM*, 7(1), 1994.
2. G. Bisson. Learning in FOL with a similarity measure. In *Proceedings of* 10^{th} *AAAI*, 1992.
3. A. Cornuejols. Analogy as minimization of description length. In G. Nakhaeizadeh and C. Taylor, eds, *Machine Learning and Statistics : The interface.* Wiley, 1996.
4. C. DeCaestecker. Incremental concept formation with attribute selection. In K. Morik, editor, *Proc. of EWSL 1989*, pages 49–58. Pitman, London, 1989.
5. P. Domingos. Rule induction and instance-based learning: A unified approach. In *Proceedings of IJCAI-95*, pages 1226–1232. 1995.
6. R.O. Duda and P.E. Hart. *Pattern Classification and scene analysis.* John Wiley and sons, Menlo Park, CA, 1973.
7. W. Emde and D. Wettscherek. Relational instance based learning. In L. Saitta, editor, *Proceedings of ICML-96*, pages 122–130, 1996.
8. U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–34. MIT Press, 1996.
9. D. Fisher. Iterative optimization and simplification of hierarchical clusterings. Technical report, Vanderbilt University, TR CS-95-01, 1995.
10. D. Gentner. Structure mapping : A theoretical framework for analogy. *Cognitive Science*, 7:155–170, 1983.
11. D. Heath, S. Kasif, and S. Salzberg. Induction of oblique decision trees. In *Proceedings of IJCAI-93*, pages 1002–1007. Morgan Kaufmann, 1993.
12. J. D. Kelly and L. Davis. A hybrid genetic algorithm for classification. In *Proceedings of IJCAI-91*, pages 645–650. Morgan Kaufmann, 1991.
13. R.D. King, A. Srinivasan, and M.J.E. Sternberg. Relating chemical activity to structure: an examination of ILP successes. *New Gen. Comput.*, 13, 1995.
14. S. Muggleton. Inverse entailment and PROGOL. *New Gen. Comput.*, 13:245–286, 1995.
15. S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19:629–679, 1994.
16. J.R. Quinlan. Learning logical definition from relations. *Machine Learning*, 5:239–266, 1990.
17. M. Sebag. Delaying the choice of bias: A disjunctive version space approach. In L. Saitta, editor, *Proceedings of ICML-96*, pages 444–452. 1996.
18. M. Sebag and C. Rouveirol. Tractable induction and classification in FOL. In *Proceedings of IJCAI-97*, to appear.
19. M. Sebag and M. Schoenauer. A rule-based similarity measure. In S. Wess, K.-D. Althoff, and M. M. Richter, eds, *Topics in Case-Based Reasonning*, volume 837 of *LNCS*, pages 119–130. Springer Verlag, 1994.
20. A. Srinivasan and S. Muggleton. Comparing the use of background knowledge by two ILP systems. In L. de Raedt, editor, *Proceedings of ILP-95.* Katholieke Universiteit Leuven, 1995.
21. *ESPRIT Project LTR 20237 ILP*^{2}. PPR-1, 1997.