Learning Probabilistic Relational Models using co-clustering methods

Anthony Coutant, Philippe Leray, Hoel Le Capitaine

To cite this version:

HAL Id: hal-00819031

https://hal.science/hal-00819031

Submitted on 29 Apr 2013

Learning Probabilistic Relational Models using co-clustering methods

Anthony Coutant
LINA (UMR CNRS 6241),
KOD/GRIM Research Groups
Ecole Polytechnique de
l'Université de Nantes, France
anthony.coutant@univ-nantes.fr

Philippe Leray
LINA (UMR CNRS 6241),
KOD Research Group
Ecole Polytechnique de
l'Université de Nantes, France
philippe.leray@univ-nantes.fr

Hoel Le Capitaine
LINA (UMR CNRS 6241),
GRIM Research Group
Ecole Polytechnique de
l'Université de Nantes, France
hoel.lecapitaine@univ-nantes.fr

ABSTRACT

Probabilistic Relational Models (PRM) are probabilistic graphical models which define a factored joint distribution over a set of random variables in the context of relational datasets. While regular PRM define probabilistic dependencies between objects' descriptive attributes, an extension called *PRM with Reference Uncertainty* (PRM-RU) allows in addition to manage link uncertainty between them, by adding random variables called *selectors*. In order to avoid problems due to large variables domains, selectors are associated with partition functions, mapping objects to a set of clusters, and selectors' distributions are defined over the set of clusters. In PRM-RU, the definition of partition functions constrain us to learn them using flat (i.e. non relational) clustering algorithms. However, many relational clustering techniques show better results in this context. Among them, co-clustering algorithms, applied on binary relationships, focus on simultaneously clustering both entities objects to use as much information available from the relationship as possible. In this paper, we present a work in progress about a new extension of PRM, called *PRM with Co-Reference Uncertainty*, which associates, to each class containing reference slots, a single selector and a single *co-partition function* learned using a co-clustering algorithm.

1. INTRODUCTION

Many machine learning approaches assume the individuals on their datasets as being independent and identically distributed (i.i.d.). However, multiple problems in real life break this hypothesis. For example, the probablity of a person to develop some genetical diseases is influenced by the history of his family's medical issues.

Bayesian networks [19] are probabilistic graphical models which rely on the i.i.d. assumption. Probabilistic Relational Models [15, 20] (PRM) extend Bayesian Networks to relational datasets defined by a relational schema. Several problems have been addressed with PRM framework in the litterature such as recommendation [14] and clustering [23]. A recent book has also been written on the use of PRM for Enterprise Architecture Analysis [3].

In *regular PRM*, the learning task aims at finding general probabilistic dependencies between classes attributes values, using information about both objects inner informations and relationships between them. In this case, relationships between objects are supposed to be known and are not part of the learned model.

PRM with Reference Uncertainty [11, 8, 9](PRM-RU) is an extension of regular PRM removing the need for exhaustive knowledge of relationships between objects. Link uncertainty for a specific object is represented as a random variable following a distribution over possible objects it can be related to. However, in order to avoid large variables problems, the possible objects are first regrouped in clusters thanks to the use of a *partition function* and the random variable simply follows a distribution over the partition function's clusters. Learning these functions can clearly be made with clustering algorithms. Nevertheless, the partition functions' definition constrain us to use flat clustering algorithms.

In the clustering litterature, more and more work are focusing on relationships data, implying several heterogeneous entities, rather than flat (i.e. non relational) and homogeneous datasets. Results [5, 7, 12] showed that, when considering relationships data, represented for example as their co-occurence matrices, it was often more efficient to cluster simultaneously both individuals and features. In the case of binary relationships, the methods following this principle are called *co-clustering* or *bi-clustering* and are of great interest in our PRM-RU learning problem.

In this article, we propose a new PRM extension, called *PRM with Co-Reference Uncertainty*, which associates, for each class containing reference slots, a single selector variable linked to a mapping function from every tuple of objects from reference slots' ranges to a set of co-clusters. We call these mapping functions *co-partition functions* and learn them using a family of co-clustering algorithms based on non-Negative Matrix Tri-Factorization (NMTF) techniques.

The rest of the paper is organised as follows. Section 2 describes the PRM with Reference Uncertainty models, a definition of their learning algorithm and discuss their weakness concerning partition functions. Section 3 then proposes a brief overview of co-clustering approaches and motivates the particular choice of NMTF based algorithms. Finally, section 4 describes our new *PRM with Co-Reference Uncertainty* model and its learning process.

2. PROBABILISTIC RELATIONAL MODELS WITH REFERENCE UNCERTAINTY

2.1 Relational Schema

A relational schema of a relational model $\mathcal{M}$ describes a set of n classes $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$. Every $\mathcal{X}_i \in \mathcal{X}$ is composed of a set of descriptive attributes $\mathcal{A}_i = \{\mathcal{A}_i^1, \ldots, \mathcal{A}_i^m\}$ and a set of reference slots $\mathcal{R}_i = \{\mathcal{R}_i^1, \ldots, \mathcal{R}_i^l\}$ relating the class to each other. We call *slot chain* of size l, a sequence $(r_1, \ldots, r_l) \in \mathcal{R}^l = \{\mathcal{R}_1 \cup \ldots \cup \mathcal{R}_n\}^l$ in which $Ran[r_{i-1}] = Dom[r_i]$. We note $\mathcal{I}(\mathcal{M})$ an instance of $\mathcal{M}$, that is a set of objects x_i for every class $X_i \in \mathcal{X}$, each object being assigned a set of specific values for its class' attributes and reference slots.

2.2 Regular PRM

Given a relational schema $\mathcal{M}$, we are interested in finding probabilistic dependencies between its attributes. Let V_A be a set of random variables, each one being defined on the domain of a different attribute of $\mathcal{M}$. Regular PRM models are defined below.

Definition 1. *[20] A regular PRM (Probabilistic Relational Model) is a pair* $(\mathcal{G}, \Theta)$ *where* $\mathcal{G}$ *corresponds to its graph structure and* Θ *to its set of parameters. The graph* $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ *is a directed acyclic graph (DAG) containing a node* $N_i \in \mathcal{N}$ *for every random variable* $V_i \in V_A$ *and an edge* $e \in \mathcal{E}$ *for every direct probabilistic dependencies between two random variables (two nodes can only be connected in* $\mathcal{G}$ *if there exist a slot chain between the corresponding attributes in* $\mathcal{M}$*). Every node* $N_i \in \mathcal{N}$ *is then associated to a parameter* $\theta_i \in \Theta$ *which describe the probability distribution of the corresponding random variable conditionally to the random variables of its parent in the graph* $\mathcal{G}$.

Regular PRM allow to use information about objects relationships in order to predict the values of these objects attributes. However, they are based on the assumption that links between objects are known for different instances. Thus, it is not possible to make link prediction with regular PRM. In this case, we can rely on an extension called *PRM with Reference Uncertainty.*

2.3 PRM-RU

In order to manage link uncertainty, we must consider more random variables than in regular PRM. Let $\mathcal{V}_R$, the set of random variables being associated to the set of reference slots of $\mathcal{M}$. Thus, given a specific instance $\mathcal{I}$ of $\mathcal{M}$, there exist for each reference slot $\mathcal{R}_{ij}$ in the relational schema, with $Ran[\mathcal{R}_{ij}] = \mathcal{X}_k \in \mathcal{X}$, a random variable $\mathcal{V}_{Rij} \in \mathcal{V}_R$ which can take values in $\mathcal{I}(\mathcal{X}_k)$, that is the set of unique identifiers of the whole objects set of type $\mathcal{X}_k$ for $\mathcal{I}$. It is important to note that variables of $\mathcal{V}_R$ directly depend on $\mathcal{I}$ and can have huge domains. Indeed, it is common to see millions (and far more) of individuals for a specific entity in a database. This leads to several problems. First, we cannot define the variables and their distributions for any instance if we keep this variables set as is. Then, it seems unreasonable to store and compute distributions over huge domains. Finally, we can be skeptical about the quality of learning this kind of random variables, since it is unlikely that we will have *sufficient statistics* for this task. As a consequence, we should replace $\mathcal{V}_R$ by a new set $\mathcal{V}_{SR}$, describing synthetical distribution over reduction of reference slots domains.

Definition 2. *[10] A PRM with Reference Uncertainty (PRM-RU)* Π *defined from* $\mathcal{M}$ *is a PRM as described in definition 1. In addition, we add for every variable* v *in* $\mathcal{V}_{SR}$*, linked to the reference slot* $\mathcal{R}_{ik}$*, a selector node in* $\mathcal{G}$ *associated to a partition function* ψ_p*. In PRM-RU, the variable* v *is defined over a set of* c *clusters* $C = \{C_{p1}, \ldots, C_{pc}\}$ *and the associated partition function defines a mapping from the objects of* $Ran[\mathcal{R}_{ik}]$ *to* $Dom[\psi_p] = Dom[v] = C$*. The selector nodes are each linked to a parameter* $\theta_i \in \Theta$ *describing the probability distribution of the corresponding random variable conditionally to the random variables of its parent in the graph* $\mathcal{G}$.

For an instance $\mathcal{I}(\mathcal{M})$, a PRM-RU describes a factored joint distribution which can be written as:

$$P(\mathcal{I}|\Pi) = \prod_{\mathcal{X}_i \in \mathcal{X}} \prod_{x \in \mathcal{I}(\mathcal{X}_i)} \prod_{\mathcal{A}_{ij} \in \mathcal{A}_i} P(x.\mathcal{A}_{ij} \,|\, Pa(x.\mathcal{A}_{ij})) \prod_{\substack{\mathcal{R}_{ij} \in \mathcal{R}_i, \\ \mathcal{X}_k = Ran[\mathcal{R}_{ij}] \\ x_k = x.\mathcal{R}_{ij}}} \frac{P(x.S(\mathcal{R}_{ij}) = \psi[x_k] \,|\, Pa(x.S(\mathcal{R}_{ij})))}{|\mathcal{I}(\mathcal{X}_k[\psi[x_k]])|}, \tag{1}$$

where $x.S(\mathcal{R}_{ij})$ is the value of the selector of the reference slot $\mathcal{R}_{ij}$ for x, $\psi[x_k]$ corresponds to $\Psi_{\mathcal{R}_{ij}}[x_k]$ that is the cluster assigned to x_k by the partition function of $\mathcal{R}_{ij}$, and $|\mathcal{I}(\mathcal{X}_k[\psi[x_k]])|$ is the number of objects in $\mathcal{I}$ of type $\mathcal{X}_k$ and assigned to the same cluster as x_k by the partition function of $\mathcal{R}_{ij}$.

The figure 1(b) shows an example of PRM-CRU defined from the relational schema in figure 1(a).

2.4 Learning PRM-RU

PRM-RU learning algorithm shares the same principles as for Bayesian networks. The structure learning is then based on an iterative greedy search method where each iteration consists in 1) generating all possible neighbours; 2) evaluating score of every PRM resulting from neighborhood search; 3) choose the model maximizing it. However, unlike Bayesian networks, the information coming from the relational schema is used in PRM-RU learning to constrain the choice of possible parents for a node. Indeed, the neighborhood generation will favor the nodes which are "close" to each other for the parenting relationships. The closeness of two nodes is defined here as the length of the slot chain between them. As usual, the algorithm continues until convergence. In

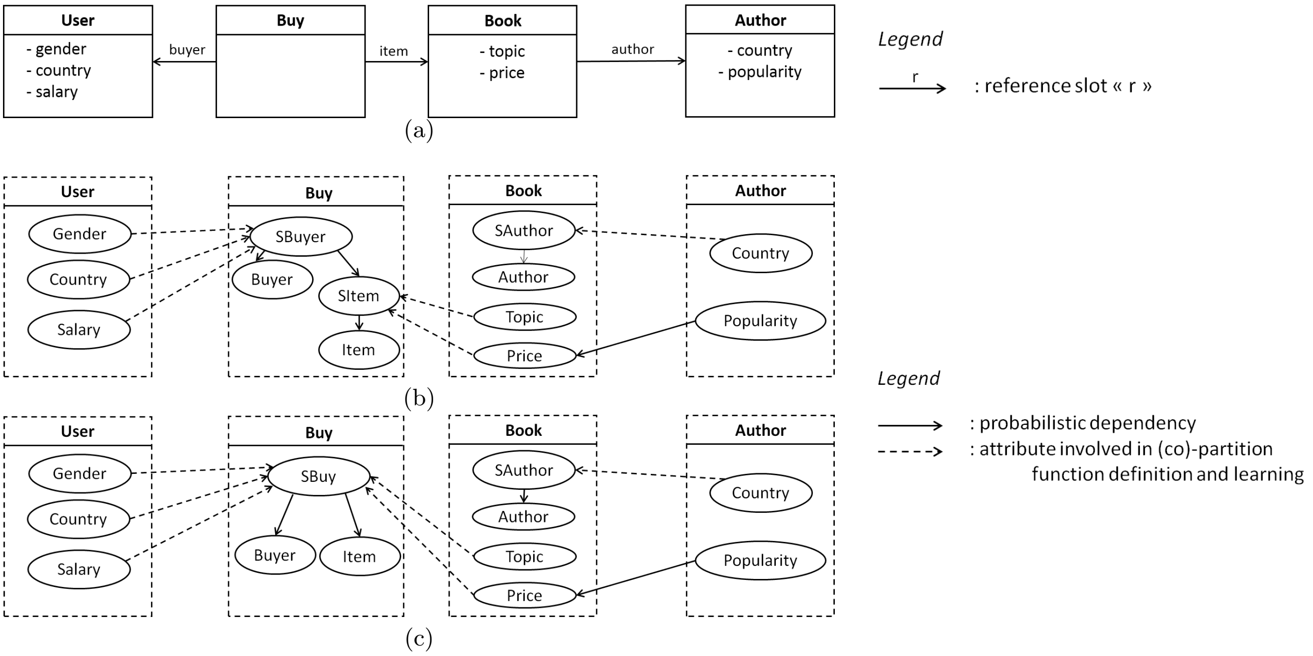

Figure 1: (a) Relational Schema (b) Example of PRM-RU structure (c) Corresponding PRM-CRU structure

Bayesian networks, the neighbourhood of a specific model is obtained with three possible operators: add, remove and reverse an edge. In PRM-RU, these operators are still available (constrained by the size of slot chains between parents and children). In addition, two new operators are added to update partition functions: refine and abstract. The former re-runs the partition function learning using one more attribute, potentially leading to more detailed functions. The latter does the complete opposite. These operators are subject to constraints limiting the choice of partition functions attributes in only one class corresponding to the range of the corresponding reference slot.

2.5 Limits of clustering in PRM-RU

PRM are made to deal with fundamentally relational data, making the information "flow" all over the model when reasoning with them. However, the concept of partition function as described in PRM-RU is not relational. Indeed, this concept is described as a mapping from a set of objects of one specific class to a set of clusters, only using information about this set of objects, as if they were isolated from the other classes' individuals. Existing literature about PRM-RU do not provide detailed information about partition function learning [11, 8, 9]. An often mentionned point is about using Cartesian product of every involved attributes values' sets. However, it does not seem to be efficient if the dataset is big. The partition function definition suggests they are rather obtained through the use of regular, flat clustering algorithm. This case could be the most we can do whenever we are in a class with only one reference slot. However, for classes describing relationship between several entities, and containing more than one slot, we can try to learn more interesting partition functions taking into consideration all the related objects together with the relationship information and not only each entity's object separately. In order to do that, we propose to use advantages of a set of clustering algorithms which have known great success in the clustering community, called *co-clustering algorithms*, instead of regular ones. This approach is described in the following section.

3. CO-CLUSTERING AND PRM

3.1 Brief overview of co-clustering approaches

In the clustering community, co-clustering techniques, also called bi-clustering, have become more and more popular and have shown better results than their flat clustering counterparts on many experiments involving non i.i.d. datasets. We can cite examples from document mining [5], gene mining [13], and image processing [21] contexts. Taking a relationship between two entities, materialized for example as a co-occurence matrix, the idea behind co-clustering approaches is to simultaneously cluster the two entities sets, using both the information of the relationship itself and eventually the inner information of entities.

Several approaches have been used to address this problem. Some methods are based on information theory. Dhillon et al. [6] see the relationship between the two entity sets as a joint distribution between random variables in the entity's sets domains, and addresses the co-clustering problem as finding the co-clusters which approximates this joint distribution the best, *i.e.* which leads to the least mutual information loss from the original one. Banerjee et al. [1] generalize the evaluation of the best co-clusters by using a distorsion measure between the approximate joint distribution and the original one, taking this measure in a family of distorsion measure called *Bregman divergences*. Shan and Banerjee [22] enlarge this hard co-clustering assignment problems into soft assignment ones, using a generative model.

Different methods use relationship matrix (Laplacian matrix of a graph or directly the relationship matrix) factorization techniques. We can cite methods based on Singular Value Decomposition [16] (SVD) or spectral clustering approaches [5]. Other interesting factorization techniques are based on non Negative Matrix Factorization approaches (NMF approach is well studied in [18] and we can see some example for data-mining in [2, 4]). They focus on finding an approximation of the original relationship matrix R_{12} of n objects of type $\mathcal{X}_1$ and m objects of type $\mathcal{X}_2$, under the form of a product of two low dimension matrices $G_1^{(n\times c)}.G_2^{(c\times m)}$ where c represents the number of clusters and G_k capture information about assignation of $\mathcal{X}_k$ individuals to clusters set $\mathcal{C}_k$. They then solve the following optimization problem:

$$\min \|R_{12} - G_1 G_2^T\|,\ s.t.\, G_1 \geq 0, G_2 \geq 0 \tag{2}$$

Recent method [7] have shown the interest of adding a third matrix to absorb the different scales of R_12, G_1 and G_2, also allowing to relax the constraint of the same number of clusters for both entities. The methods based on a 3-factor decomposition are called non-Negative Matrix Tri-Factorization (NMTF) and solve the following optimization problem:

$$\min \|R_{12} - G_1 S G_2^T\|,\ s.t.\, G_1 \geq 0, G_2 \geq 0, S \geq 0 \tag{3}$$

For the co-clustering problem, the $S \geq 0$ constraint is relaxed.

3.2 Integrating co-clustering in PRM learning process

Even if many approaches exist to deal with co-clustering, most of them do not fit perfectly with the richness of information available in PRM. Indeed, most of co-clustering algorithm do not use inner objects data in addition to relationship information, whereas we use both information to learn a PRM. As a consequence, this is one main constraint for the selection of a co-clustering algorithm to integrate with PRM.

Some recent work [12, 26, 24, 25] regularize the matrices factorization for co-clustering tasks, as described above, through the adding of entities information by the mean of intra-classes affinity matrices. Beyond the fact that it provides the desired advantage of intra-class information use, these co-clustering algorithms can take into consideration the topological space (or manifold) of every entity, avoiding the trouble of always considering simple Euclidean spaces. These techniques show good results in experimentations. For these reasons, co-clustering algorithms operating on manifolds are well suited for our PRM learning problem. We briefly describe their principle in the following subsection.

3.3 Co-clustering with Laplacian regularization

Let R_{12} be a relationship matrix between two entities X_1 and X_2. The two entities X_k with $k \in \{1,2\}$ are described as sets of objects $X_k = \{x_k^1, \ldots, x_k^{n_k}\}$ where n_k is the number of objects of the entity k. We can describe information between objects of the entity X_k under the form of a square affinity matrix W_k. Let now define a clusters set C_k for every X_k. The objective of co-clustering is then to find the functions f_k mapping every object $x_i^k \in X_k$ to a cluster $c_j^k \in C_k$ for every $k \in [1,2]$. From matrices R_{12}, W_1 and W_2, the algorithm targets to find the best result matrices G_k and S minimizing:

$$\begin{aligned} J &= \|R - G_1 S G_2{}^T\|_F^2 + \sum_{k\in\{1,2\}} \lambda_k \mathbf{tr}\left[G_k{}^T L_k G_k\right] \\ s.t.\ & G \geq 0 \end{aligned} \tag{4}$$

where $\|.\|_F$ is the Frobenius norm, $L_k = D_k - W_k$ is the graph Laplacian corresponding to X_k and D_k is the diagonal degree matrix corresponding to X_k. Two results are then useful for us. First, the matrices G_k can be interpreted after normalization as a posterior distribution of clustering for every $x_i^k \in X_k$. Then, the S matrix can allow us to calculate $P(\mathbf{C}_1\mathbf{C}_2)$, where $\mathbf{C}_1$ and $\mathbf{C}_2$ are random variables respectively defined on C_1 and C_2.

We can see that integrating this kind of co-clustering algorithm into PRM learning process requires the calculation of the right intra-class similarity matrices. The details of this integration is the object of the following section.

4. PRM WITH CO-REFERENCE UNCERTAINTY

We propose here to take benefit from co-clustering approaches in order to learn PRM models using even more relational information than those used by PRM-RU. We define a new model based on this idea called *PRM with Co-Reference Uncertainty* (PRM-CRU) described below.

4.1 Definition

Keeping the same definition of a relational model $\mathcal{M}$ as before, we update the set of random variables $V^S(\mathcal{M}) = \mathcal{V}_A \cup \mathcal{V}_{SR}$ considered. We now define the synthetical random variables $\mathcal{V}_{SR}$ corresponding to reference slots $\mathcal{R}$ of $\mathcal{M}$ as variables describing distributions over reduction of cartesian product of several reference slots domains. We now associate for every class $\mathcal{X}_i \in \mathcal{X}$ at most one partition function, which is a co-partition function if the class contains two reference slots. The formal definition of PRM-CRU is given below.

DEFINITION 3. *A PRM-CRU Π defined from $\mathcal{M}$ is a PRM as described in definition 1. In addition, we add for every variable $v \in \mathcal{V}_{SR}$, linked to an entity $\mathcal{X}_i \in \mathcal{X}$ with $|\mathcal{R}_i| \in \{1,2\}$, a co-selector node in $\mathcal{G}$ associated to a co-partition function ψ_i. In PRM-CRU, the variable v is defined over a set of c co-clusters $C = \{C_{i1}, \ldots, C_{ic_i}\}$ and the associated partition function defines a mapping from $\mathcal{I}(Ran[\mathcal{R}_{i1}]) \times \ldots \times \mathcal{I}(Ran[\mathcal{R}_{ir_i}])$ to $Dom[\psi_i] = Dom[v] = C$. The selector nodes are each linked to a parameter $\theta_i \in \Theta$ describing the probability distribution of the corresponding random variable conditionally to the random variables of its parent in the graph $\mathcal{G}$.*

The figure 1(c) shows an example of PRM-CRU defined from the relational schema in figure 1(a). It is important to note that, if $\mathcal{M}$ contains at least one entity with more than two reference slots, it is not possible to create a PRM-CRU on it. Relaxing this constraint will be the object of future work.

4.2 Learning PRM-CRU

PRM-CRU models do not need a change of learning algorithm structure, relatively to PRM-RU learning, but a redefinition of the **refine** and **abstract** operators which content will strongly depend on the choosen co-clustering algorithm. First, it is important to remind that many co-clustering algorithm only focus on relationship matrices (for binary cases) or equivalent in more dimension. For these algorithms, adding or removing attributes from concerned classes by the refine and abstract operators will not change anything. Thus, in this first case, learning a PRM-CRU boils down to first learning every co-partition functions once and for all, and then use only edges operators during greedy search. Secondly, for co-clustering methods using both relationship and entities inner information, the input taken for the latter is under the form of similarity matrices. In this case, the refine and abstract method must trigger a three steps procedure: 1) update the set of attributes $\mathcal{P}_i$; 2) compute the new similarity matrix for the updated entity's objects (no need to recompute these matrices for other classes of the relationship); 3) relaunch the co-clustering algorithm with updated similarities.

More formally, let consider a specific partition function ψ_i in Ψ linked to the entity $\mathcal{X}_i \in \mathcal{X}$ with $|\mathcal{R}_i| = 2$ (there is no partition function ψ_i if $|\mathcal{R}_i| = 0$; the partition function works as in PRM-RU in the case where $|\mathcal{R}_i| = 1$). Let $X = Ran[\mathcal{R}_{i1}] \cup Ran[\mathcal{R}_{i2}]$, the set of classes being the range of at least one reference slot of $\mathcal{X}_i$. Let then $A = \bigcup\limits_{\forall \mathcal{X}_i \in X} \mathcal{A}_i$ the set of potential attributes for ψ_i and $\mathcal{P}_i \subset A$, the set of current used attributes by ψ_i. Given a fixed number of clusters k, the refine operator will add an attribute $a \in A - \mathcal{P}_i$ to $\mathcal{P}_i$ from class $\mathcal{X}_i \in X$, recalculate the similarity matrix $S_{\mathcal{I}(\mathcal{X}_i)}$ and re-runs the corresponding co-clustering algorithm to find k new clusters from all the similarity matrices of involved classes instances. Similarly, the abstract operator will remove an attribute $a \in A - \mathcal{P}_i$ from $\mathcal{P}_i$, recalculate the corresponding similarity matrix and finally re-runs the co-clustering algorithm. Unlike for the PRM-RU case, only one constraint still hold on the two operators call: the one forbidding a refine call if every potential attribute has been added ($\mathcal{P}_i = A$). Indeed, a co-clustering algorithm can still work without any attribute since it always has the relationship matrix to work on.

4.3 Discussion

PRM-CRU are aimed to offer more accurate clusters to the PRM learning algorithm than in PRM-RU. The objective is to improve the probabilistic likelihood $P(\mathcal{I}|\mathcal{G}, \Psi, \Theta)$ we can reach with Greedy Search heuristic, since we can more accurately fit the data with co-clustering, than with flat clustering methods, in numerous situations. This gain comes at a cost, however, due to the limitation of co-clustering techniques themselves. Indeed, we can not currently work with relational schema with classes having more than two reference slots, and classes with exactly two reference slots can not have more than one descriptive attribute (otherwise, it would not be able to have a single matrix to represent the relationship data).

5. CONCLUSION AND PERSPECTIVES

In this paper, we have presented a work in progress about a new extension of PRM, called *PRM with Co-Reference Uncertainty*, which defines a single selector variable for each class containing reference slots. This variable is associated to a *co-partition function*, which is learned thanks to a co-clustering algorithm. We have particularly described algorithms based on NMF techniques and have motivated the interest of using more particularly NMTF techniques with Laplacian regularization.

It is important to note that, in its current state, our work can not be applied to all datasets since it does not support every relational schema. The main limitation is due to the use of co-clustering algorithms which can only be used on binary relationships expressed as matrices. Since a relational schema can involve more complex classes with more reference slots, it is important to work on generalizing this method for the case of any n-ary relationship. This could be done for example with tensor decomposition techniques [17], which seems promising. We will focus on this generalization in future work.

Another interesting task on which we will focus on is the analysis of the set of co-partition functions obtained by learning the PRM-CRU. By finding a consensus on the multiple co-partition functions obtained, we can try to reverse the problem addressed in this article: making multi-relational clustering with PRM learning algorithm after having made PRM learning using relational clustering.

6. REFERENCES

[1] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 509–514, New York, NY, USA, 2004. ACM.

[2] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences*, 101(12):4164–4169, 2004.

[3] M. Buschle, J. Ullberg, U. Franke, R. LagerstrÃűm, and T. Sommestad. A tool for enterprise architecture analysis using the prm formalism. In P. Soffer and E. Proper, editors, *Information Systems Evolution*, volume 72 of *Lecture Notes in Business Information Processing*, pages 108–121. Springer Berlin Heidelberg, 2011.

[4] M. Cooper and J. Foote. Summarizing video using non-negative similarity matrix factorization. In *Multimedia Signal Processing, 2002 IEEE Workshop on*, pages 25–28, 2002.

[5] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 269–274, New York, NY, USA, 2001. ACM.

[6] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages

89–98, New York, NY, USA, 2003. ACM.

[7] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 126–135, New York, NY, USA, 2006. ACM.

[8] L. Getoor. *Learning Statistical Models from Relational Data*. PhD thesis, Stanford, 2001.

[9] L. Getoor. *Introduction to statistical relational learning*. The MIT press, 2007.

[10] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of relational structure. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 170–177, 2001.

[11] L. Getoor, D. Koller, B. Taskar, and N. Friedman. Learning probabilistic relational models with structural uncertainty. In *Proceedings of the AAAI Workshop on Learning Statistical Models from Relational Data*, pages 13–20, 2000.

[12] Q. Gu and J. Zhou. Co-clustering on manifolds. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 359–368, New York, NY, USA, 2009. ACM.

[13] D. Hanisch, A. Zien, R. Zimmer, and T. Lengauer. Co-clustering of biological networks and gene expression data. *Bioinformatics*, 18(suppl 1):S145–S154, 2002.

[14] Z. Huang, D. D. Zeng, and H. Chen. A unified recommendation framework based on probabilistic relational models. In *in Fourteenth Annual Workshop on Information Technologies and Systems (WITS*, pages 8–13, 2004.

[15] D. Koller and A. Pfeffer. Probabilistic frame-based systems. In *Proceedings of the National Conference on Artificial Intelligence*, pages 580–587. JOHN WILEY & SONS LTD, 1998.

[16] M. Lee, H. Shen, J. Z. Huang, and J. S. Marron. Biclustering via sparse singular value decomposition. *Biometrics*, 66(4):1087–1095, 2010.

[17] B. London, T. Rekatsinas, B. Huang, and L. Getoor. Multi-relational weighted tensor decomposition. In *NIPS Workshop on Spectral Learning*, 2012.

[18] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[19] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[20] A. J. Pfeffer and D. Koller. *Probabilistic reasoning for complex systems*. Stanford University Stanford, CA, 2000.

[21] G. Qiu. Image and feature co-clustering. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 991–994 Vol.4, 2004.

[22] H. Shan and A. Banerjee. Bayesian co-clustering. In *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*, pages 530–539, 2008.

[23] B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2*, IJCAI'01, pages 870–876, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[24] H. Wang, H. Huang, and C. Ding. Simultaneous clustering of multi-type relational data via symmetric nonnegative matrix tri-factorization. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 279–284, New York, NY, USA, 2011. ACM.

[25] H. Wang, F. Nie, H. Huang, and F. Makedon. Fast nonnegative matrix tri-factorization for large-scale data co-clustering. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1553–1558. AAAI Press, 2011.

[26] J. Yoo and S. Choi. Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on stiefel manifolds. *Information Processing and Management*, 46(5):559 – 570, 2010.