Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference

Learning Probabilistic Relational Models Using Non-Negative Matrix Factorization

Anthony Coutant, Philippe Leray and **Hoel Le Capitaine**
(anthony.coutant, philippe.leray, hoel.lecapitaine)@univ-nantes.fr
LINA (UMR CNRS 6241), DUKe Research Group, École Polytechnique de l'Université de Nantes, France

Abstract

Probabilistic Relational Models (PRMs) are directed probabilistic graphical models representing a factored joint distribution over a set of random variables for relational datasets. While regular PRMs define probabilistic dependencies between classes' 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 variables with large 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 constrains us to learn them only from concerned individuals entity attributes and to assign the same cluster to a pair of individuals having the same attributes values. This constraint is actually based on a strong assumption which is not generalizable and can lead to an under usage of relationship data for learning. For these reasons, we relax this constraint in this paper and propose a different partition function learning approach based on relationship data clustering. We empirically show that this approach provides better results than attribute-based learning in the case where relationship topology is independent from involved entity attributes values, and that it gives close results whenever the attributes assumption is correct.

Introduction

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

Bayesian networks (Pearl 1988) are probabilistic graphical models for propositional datasets, where we assume individuals as i.i.d. Probabilistic Relational Models (Koller and Pfeffer 1998; Pfeffer and Koller 2000) (PRMs) extend Bayesian networks to relational data, releasing the i.i.d. assumption. Several problems have been addressed with PRM framework in the literature such as recommendation (Huang, Zeng, and Chen 2004) and clustering (Taskar, Segal, and Koller 2001). Recent work has also shown a use of PRM for Enterprise Architecture Analysis (Buschle et al. 2011).

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

PRM with Reference Uncertainty (Getoor et al. 2000; Getoor 2001; 2007)(PRM-RU) is an extension of PRM removing the need for exhaustive knowledge of relationships between individuals. Link uncertainty for a specific individual is represented as a random variable following a distribution over possible individuals it can be related to. However, in order to avoid large variables problems, the possible individuals are grouped thanks to the use of a *partition function* and the random variable simply follows a distribution over these groups. To the best of our knowledge, no work has been published to study the impact of partition function choice to learning performances. The only solution involves a determinism based on individuals inner information. Besides, partition functions, as defined in PRM-RU, must respect the constraint that individuals sharing the same inner description are grouped together.

This is a strong assumption which cannot be generalized in every case. As an example, students of same gender and age do not necessarily have close study patterns. Thus, given a set of attributes supposed to explain a relationship for one entity, parameters estimation of the model can lead to high variances inside each partition, resulting in poor parameters estimation results. Besides, it means that a relationship with no attribute cannot be used for learning.

In this paper, instead of building deterministic partition functions based on an attributes-oriented learning, we choose to use a clustering algorithm to learn relationship-based partition functions from data. We use for this task the so-called non-negative matrix factorization techniques (NMF) and empirically show that: 1) clustering based partition function learning provides more accurate models than currently used Cartesian product (CP) partition functions whenever relationships are not well explained by attributes; 2) clustering based partition functions provides close results to CP ones for cases where attributes perfectly describe the relationship and where CP is optimum.

The rest of the paper is organized as follows. We first describe PRM with Reference Uncertainty models, a defini-

Copyright © 2014, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

tion of their learning algorithm and discuss their weaknesses concerning partition functions. Then, we propose a brief overview of non-negative matrix factorization approaches and their use for clustering. Next, we describe our partition function learning approach using NMF and finally show experimental results.

Probabilistic Relational Models with Reference Uncertainty

Probabilistic Relational Models (PRMs) are an extension of Bayesian networks to learn from and infer in relational datasets with uncertainty.

Relational schema

In order to be compatible, a PRM and a dataset must respect the same underlying data structure, which is formalized under the concept of *relational schema*. A relational schema describes a set of classes $\mathcal{X}$. Every $\mathcal{X}_i \in \mathcal{X}$ is composed of a set of *descriptive attributes* and a set of *reference slots*. The set of descriptive attributes for a class $\mathcal{X}_i$ is denoted $\mathcal{A}(\mathcal{X}_i)$ and its set of reference slots is $\mathcal{R}(\mathcal{X}_i)$. The attribute A and the reference slot r of $\mathcal{X}_i$ are respectively denoted as $\mathcal{X}_i.A$ and $\mathcal{X}_i.r$. We call *slot chain* of size l, a sequence $(r_1, \dots, r_l) \in \mathcal{R}(\mathcal{X})^l$, where $\mathcal{R}(\mathcal{X})$ is the set of all the reference slots in the schema and in which for all $i \leq l$ we have $Ran[r_{i-1}] = Dom[r_i]$. An instance $\mathcal{I}$ of the schema is a set of individuals X_i for every class $\mathcal{X}_i \in \mathcal{X}$, each individual being assigned a set of valid values for its attributes and reference slots.

Probabilistic Relational Model

Given a relational schema, a Probabilistic Relational Model defines a probability distribution over its instances. PRMs are formally described in definition 1.

Definition 1 *(Pfeffer and Koller 2000) Probabilistic Relational Models (PRM) are composed of a qualitative dependency structure $\mathcal{S}$ and a set of parameters Θ. The structure defines, for every class $\mathcal{X}_i$ and every attribute $\mathcal{X}_i.A$ in the relational schema, a set of parents $Pa(\mathcal{X}_i.A)$. Each parent of $\mathcal{X}_i.A$ is of the form $\mathcal{X}_i.K.B$, where K is a slot chain with $Dom[K] = \mathcal{X}_i$ and $Ran[K] = \mathcal{X}_j \in \mathcal{X}$. Given a structure, Θ defines, for every attribute $\mathcal{X}_i.A$, a conditional probability distribution (CPD) $P(\mathcal{X}_i.A|Pa(\mathcal{X}_i.A))$*

PRMs allow us to learn probability distributions over individual attributes values of relational datasets. Assuming that learned probability distributions are stationary over considered instances, it is possible to instantiate a learned PRM Π from a new instance $\mathcal{I}$ of the same relational schema, in order to infer on its objects. Concretely, we consider the *relational skeleton* of $\mathcal{I}$, denoted $\sigma_r(\mathcal{I})$, which is composed of both objects lists for every class of the relational schema and value of reference slots for every object. From Π and $\sigma_r(\mathcal{I})$, it is possible to build a *ground Bayesian network* in which we can make regular Bayesian network inference for the objects of $\mathcal{I}$. PRMs can thus be seen as templates of joint probability distributions for any instance (or any relational skeleton) of the relational schema.

The problem with attributes-oriented PRMs is the requirement of a full specification of the relational skeleton for any instance to infer on. Thus, there can be no uncertainty on reference slot values, which make these PRMs unsuited for link-prediction tasks. This latter problem can however be tackled by the use of dedicated PRM extensions. We focus on one such extension in this article, called *PRM with Reference Uncertainty*.

PRM with Reference Uncertainty

We now consider the case where we do not have a full specification of reference slot values in instances we want to infer on. Given an instance $\mathcal{I}$, we extract its *object skeleton*, denoted $\sigma_o(\mathcal{I})$, consisting in the list of all the objects identifiers without any information about their descriptive attribute and reference slot values.

Probabilistic Relational Models with Reference Uncertainty (PRM-RU) are made to deal with situations whenever all objects are defined but links between them are uncertain, and thus define probability distributions for both attributes and reference slots values. Note that for a reference slot $\mathcal{X}_i.R$, where $Ran[\mathcal{X}_i.R] = \mathcal{X}_j$, R can take any value in $\sigma_o(\mathcal{I}(\mathcal{X}_j))$, the set of objects' identifiers of class $\mathcal{X}_j$ in $\mathcal{I}$. Thus, in order to define a distribution over these values, we could simply define a distribution over all objects identifiers of $\mathcal{X}_j$. This is however not recommended for several reasons. First, a learned PRM-RU would be difficult to instantiate and generalize for other instances. Then, it seems unreasonable to store and compute distributions over huge domains, since relational datasets may have millions individuals. Finally, we can doubt on the quality of learned models, since it is unlikely to have *sufficient statistics* for this task. To solve these problems, PRM-RU relies on the concept of objects partitioning, making the assumption that an object is chosen during a two-steps process where we first choose a subset of objects and then choose an object inside it. In order to build these subsets, every reference slot $\mathcal{X}_i.R$ with $Ran[\mathcal{X}_i.R] = \mathcal{X}_j$ is associated to a *partition function* ψ_R mapping individuals of $\mathcal{X}_j$ to a set of groups C_R. In addition, every reference slot $\mathcal{X}_i.R$ is associated to a *selector* attribute $\mathcal{X}_i.S_R$ taking values in C_R. This selector is finally treated as another attribute, having parents and a CPD given these parents. PRM-RU are formally defined in 2.

Definition 2 *(Getoor et al. 2001) A PRM with Reference Uncertainty (PRM-RU) is a PRM as described in definition 1. In addition, we add for every reference slot $\mathcal{X}_i.R$: a partition function ψ_R mapping the set of individuals $Ran[\mathcal{X}_i.R]$ to a set of groups C_R; a selector attribute $\mathcal{X}_i.S_R$ taking on values in C_R; a set of parents $Pa(\mathcal{X}_i.S_R)$; a CPD $P(\mathcal{X}_i.S_R|Pa(\mathcal{X}_i.S_R))$.*

The Figure 1(bottom) shows an example of PRM-RU defined from the relational schema in Figure 1(top).

Learning PRM-RU and PRM

PRM learning shares the same principles as for Bayesian networks. The structure learning is based on an iterative greedy search method where each iteration consists in: 1)

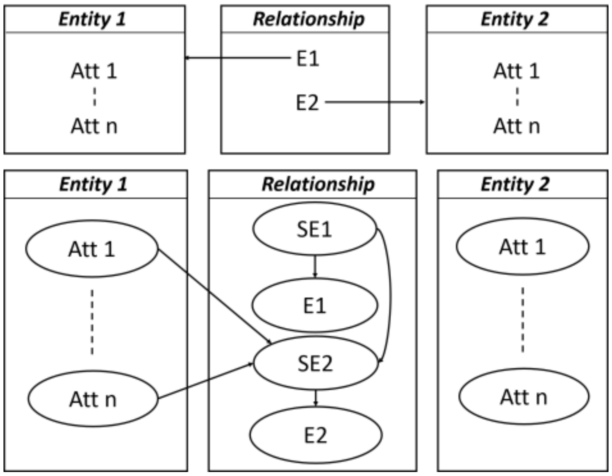


Figure 1: (top) Experiments relational schema. An arrow from A to B means that attribute A references class B. (bottom) PRM-RU structure to learn in experiments. An arrow from attribute A to attribute B means that A is a parent of B.

generating all *current* structure's neighbors; 2) learning parameters for each neighbor; 3) evaluating each neighbor's score using both its structure and parameters; 4) choosing the one maximizing it as new *current* structure for next iteration. However, unlike Bayesian networks, the information coming from the relational schema is used in PRMs learning to constrain the choice of possible parents for a node. Indeed, the greedy search algorithm for PRMs progressively checks dependencies between attributes separated by longer and longer slot chains, privileging the shortest ones first. The algorithm continues until convergence. In Bayesian networks and PRMs, the structure is a Directed Acyclic Graph (DAG) describing (in)dependencies between variables. Thus, neighbors of a specific model are obtained with three possible edges manipulation operators: *add*, *remove* and *reverse*. In PRM-RU, in addition, the partitions functions are also considered as part of the model structure. As a consequence, two new partition functions manipulation operators are added in PRM-RU structure learning: *refine* and *abstract*. The former (resp. latter) leads to more (less) detailed functions, with more (less) clusters.

Partition functions in PRM-RU

As seen before, learning a PRM-RU implies to learn partition functions, defined as a mapping of objects from one specific class to a specific set of clusters. To the best of our knowledge, existing literature about PRM-RU do not provide detailed information about partition function learning (Getoor et al. 2000; Getoor 2001; 2007). Cartesian product (CP) of some attributes values' sets are used in practice. More formally, CP partition functions are defined as follows: Let ψ_R be a partition function for a reference slot $\mathcal{X}_i.R$ of a PRM-RU, with $Ran[\mathcal{X}_i.R] = \mathcal{X}_j$. If it is a Cartesian product partition function, and if $\mathcal{P}_R \subseteq \mathcal{A}(\mathcal{X}_j)$ is the set of involved attributes for ψ_R, then for an instance $\mathcal{I}$, two individuals of $\mathcal{I}(\mathcal{X}_j)$ are assigned to the same cluster in the set of clusters C_R if and only if they have the same attributes values in $\mathcal{P}_R$.

In the CP partition function case, the *refine* (resp. *abstract*) operator adds (resp. removes) an involved attribute. Note that learning partition functions in this case simply boils down to executing k simple SELECT queries in the database. In the case where involved attributes are indexed, the complexity can be of $\mathcal{O}(k \times log(n))$.

In the definition of partitions functions for PRM-RU, it is required that two individuals having the same attributes values must be in the same partition. The CP partition function meets this requirement. However, this constraint induces that relationships can be fully explained by their involved entities' attributes, and thus that two individuals having the same inner description are related in a similar way to other individuals. This is actually a strong assumption which cannot be verified in every situation and can thus lead to poor results whenever the relationship is not clearly related to attributes values. As an example, two students having the same age and gender do not necessarily study the same subjects. Thus, the *study* relationship cannot be explained by grouping students by age and gender, and trying to define one single pattern of study for each group. Besides, we see that this assumption makes the relationship strongly dependent of the granularity of entities description. In addition, since the partition functions are determined without examining the relationship data, this information is not used.

These limitations are as much reasons for relaxing the constraint. We propose in this paper a completely different approach of partition functions learning, using clustering methods on the relationship data itself. We choose the so-called non-negative matrix factorization (NMF) approach to tackle the problem, since it is very generalizable, customizable and extended in many ways. The next section briefly describes NMF and its clustering effects.

Non-negative Matrix Factorization clustering

NMF problem formulation

Considering a matrix V of size $n \times m$, and a dimension value k, non-negative matrix factorization methods focus on finding non negative factors W of size $n \times k$ and H of size $k \times m$, such that, with respect to some dissimilarity measure Δ, we have $\Delta(V, W.H) \approx 0$. Generally, k is chosen small with respect to n and m. In the literature, most famous measures are the Frobenius norm, also known as the Euclidean norm, and KL Divergence, but many frameworks have been developed to compute NMF with other measures such as any Bregman divergence (Banerjee et al. 2004). Note that even if NMF leads to good results in practice, it has been proved it is an NP-hard problem (Vavasis 2009).

NMF for clustering

Experiments show that NMF provides good clustering results (Hoyer 2004) and NMF techniques for clustering have been widely studied in the literature. It has notably been proved (Ding et al. 2006) that NMF factorization with Euclidean distance is equivalent to K-Means clustering if we add the orthogonality constraint to H. Thus, this constraint is often encountered when dealing with NMF for clustering, especially as it guarantees a unique solution. In a clustering context, W can be interpreted as a degree of belonging of each clustered individual to each of the k clusters. Complementarily, H can be interpreted as a set of k clusters centroids description.

Learning PRM-RU with NMF methods

Let us consider an instance of an unvalued binary relationship R as a $n \times m$ matrix $M(R)$ where n is the number of individuals for the first involved entity and m is the number of individuals for the second one. Let $M(R)_{ij}$ be equal to 1 if there exist a link between i^{th} individual of entity 1 and j^{th} individual of entity 2, and 0 otherwise.

Since this relationship involves two entities, we must learn two different partition functions. Thus, in order to learn partition function for entity 1 (resp. entity 2), we apply NMF algorithm on $M(R)$ (resp. $M(R)^T$) to find, for each line of the matrix, the distribution over clusters for the corresponding individual.

Once NMF has been computed, individuals of the learning dataset are assigned to a cluster based on the maximum likelihood estimation of the corresponding W matrix line. Finally, once each individual is assigned to a cluster, regular parameters learning can be made. Note that, since the used NMF approach does not use the attributes values here, it is useless to explore the hypothesis space of involved attributes during structure learning. However, we could for example redefine the refine and abstract operators to make them increase or decrease the clusters number k.

After learning, in order to find the cluster $C(i)$ value of any new individual i, considering its relationship vector with distant entity's individuals v, and the learned factor matrix H, we calculate:

$$C(i) = \underset{j}{\operatorname{argmin}}\, \Delta(v, H_{j.}) \qquad (1)$$

where Δ is a dissimilarity measure, rationally equal or close to the one used for NMF learning.

NMF partition functions learning brings several advantages. First, since it does not depend on attributes values, we can still find some accurate dependencies between clusters and other nodes of the PRM-RU during structure learning, even if there are no attribute. In addition, one of the advantage of this method is to make full use of the relationship data, even when the relationship itself does not have any attribute. It is important to note that, beyond these specific cases tackled by our approach, if a relationship is perfectly described by involved entity attributes, it should also be found by NMF approach and then lead to close results as what would be obtained with CP partition functions learning.

In the next section, we empirically validate benefits of this method over CP partition function learning.

Experiments

In this section, we test our relational based partition learning approach on generated datasets, and compare its performance with existing attributes-oriented CP partition function learning.

Datasets generation

In our experiment, two datasets generation processes have been created: one aims at privileging results of clustering based learning, whereas the other aims at privileging Cartesian product learning. The relational schema of generated datasets consists in three classes: two entities called E1 and E2, and a relationship class called R. R only contains one reference slot to E1 and another one to E2. E1 and E2 classes contain the same parametrized number of binary descriptive attributes. We generate the same number of E1 and E2 individuals by independently drawing their attributes values from some defined distribution P(Ei.Aj) for the j^{th} attribute of Ei. We then assign E1 and E2 individuals to the corresponding set of clusters C1 and C2, in a different way depending on the dataset generation method: the NMF favorable method randomly assigns cluster value to E1 and E2 individuals, following some distributions P(S1) and P(S2) given as parameter; the NMF unfavorable method assigns cluster values according to E1 and E2 individuals' attributes' values, respecting the constraint of two individuals being in the same partition if they have similar inner description. After this step, we can generate R individuals. This is done by first drawing a cluster from C1 using P(S1) defined above. Then, draw a cluster from C2 using some P(S2|S1) given as parameter and finally uniformly choose an E2 individual from the selected C2 cluster. This individual is finally added to the typical vector of selected C1 cluster. Indeed, in this experiment, every E1 individual is supposed to have the exact same relationships vector to E2 individuals as other E1 individuals of the same cluster. In order to allow simple comparison between NMF and CP partition function learning, we must keep strict structure equality. Thus, the number of built clusters in generated datasets is always equal to 2^m where m is the number of binary attributes of entities classes.

Learning protocol

An experiment execution consists in first generating either a favorable or unfavorable dataset, and then splitting the dataset into 10 parts in order to make 10 folds cross validation learning, each time using 1/10th of dataset as testing set and the remaining 90% as training set. During an execution, 3 models are simultaneously learned: our model using NMF, the CP based model, and an optimistic model knowing the real assignations to clusters. The learned structure is the same for each model (cf. Figure 1(bottom)) and has been chosen since it maximizes the number of parameters that can be learned for R individuals (adding more edges is not allowed due to PRM-RU cycle constraints: see (Getoor 2007) for further details). For each cross-validation fold and for each of the 3 compared models, we keep the log likelihood calculated on the test set after learning on remaining data, and the purity clustering score for partitions functions. The NMF implementation used was the freely available one of (Pathak et al. 2007), minimizing KL-divergence, a common divergence measure for NMF problems.

An execution differs from the other by several parameters: the number of E1 and E2 binary attributes (which determines also the number of clusters) and the number of individuals of

E1 and E2. In order to avoid problems while learning NMF, the number of clusters asked for the E1 partition function can be updated to reflect the real number of clusters present in the training dataset. Indeed, even if the data generation process has been made to give individuals to each cluster, some selection for the training dataset can result in the missing of some clusters representatives. Not changing the number of clusters in this case could lead to drastically different mean-prototypes and thus poor learning results, especially if the number of clusters is high compared to the number of individuals.

Evaluation method

We evaluate an experiment execution by calculating the mean and standard deviation of the 10 log-likelihoods obtained for each learned model during the 10 fold cross-validation process. We also calculate for each execution the statistical significance, for the alternate hypothesis that our NMF model gives better results than the CP one. In order to compute this significance, we calculate for each experiment execution the z-score z with the following formula:

$$z = \frac{\mu_{NMF} - \mu_{CP}}{\sigma_{NMF}} \tag{2}$$

where μ_{NMF}, μ_{CP} and σ_{NMF} respectively represent for a set of cross-validation results: the mean of NMF results, the mean of CP results, and the standard deviation of NMF results. We then compute the left-tailed p-value of this z-score, which can be computed from the *cumulative standard normal distribution*:

$$P(Z \leq z) = \int_{-\infty}^{z} \frac{1}{2\pi} e^{\frac{-u^2}{2}} du \tag{3}$$

The one tailed p-value indicates the probability for NMF results to reject our null hypothesis stating that the mean of NMF results is not significantly lower than the mean of CP results. The lower the value, the more the NMF results are significantly lower.

Results

We run several experiments, in order to study the impact of different parameters on the obtained log-likelihood. More concretely, we made the number of individuals vary from 25 to 200, leading to an R matrix size from 625 to 40000. We also studied the impact of the number of attributes variation from 1 to 4, implying a number of clusters variation from 2 to 16. Log likelihoods' mean and standard deviation during 10 folds cross-validation process are given in Figure 2 for each combination of parameters and learning method, as well as statistical significance corresponding to previously explained p-value from z-score computation.

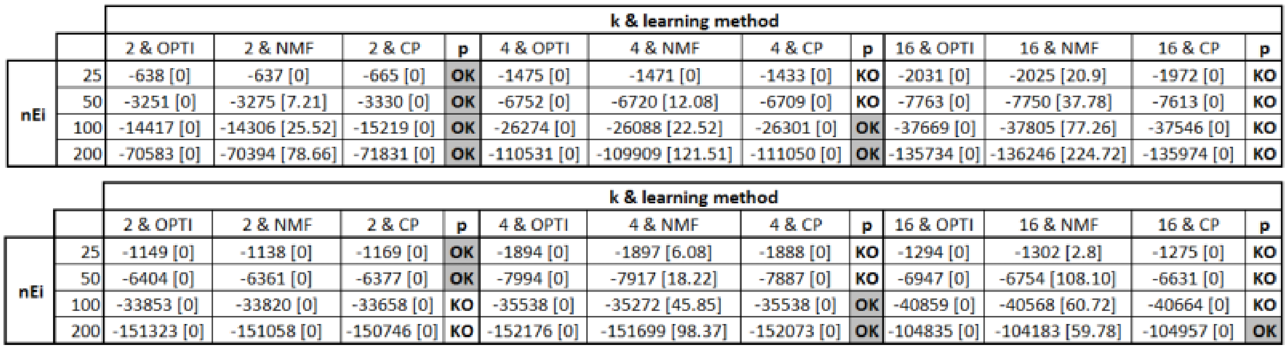
According to Figure 2 (top), we can see that our NMF model performs better than the CP one in a favorable case whenever clusters are not explained by entities' attributes. However, this is not true anymore whenever the number of clusters is high relatively to the number of entities individuals. In this case, it seems that CP methods can learn detailed enough parameters to fit the data. These results can be nevertheless explained by the experiment protocol itself. Indeed, the more we raise the number of clusters for a fixed number of entities, the more we take the risk of having some clusters not represented in some cross-validation fold. As a consequence, and even if we chose to change the cluster number depending on the learning instance, individuals on the test set of the unknown cluster would have poor likelihood results, which can explain the difference. An evidence of this in results can be seen thanks to the purity of NMF models which goes for the single time below 1 (0.95) for entity 1 only whenever there are 16 clusters in NMF favorable datasets. We can also see, according to Figure 2 (bottom) that our NMF model is not necessarily better or do not have significantly better accuracy in an unfavorable case whenever clusters are totally explained by attributes values. However, NMF results are neither significantly worse. This seems intuitive since if individuals having the same attributes values also have the same relationship patterns, then the NMF method should see it through the relationship matrix. Our model appears to be then more generalizable than Cartesian product one. We can note an exception in the unfavorable data for some cases whenever the NMF method is significantly better than Cartesian product. However, since significant p-values seem randomly distributed in the table, nothing can explain it better than chance.

Discussion

It is important to note that this experiment could be modified in some ways. First, the typical vectors assumption is a rather ideal hypothesis, ensuring some stability in data, allowing to focus on learning comparison of models. However, it can explain the very closeness of some results of both NMF and Cartesian product learning to the optimal one. It would be interesting to remove this assumption by also drawing E1 individuals during the relationship individuals generation step, in order for individuals of same cluster to have the same distribution of data over the other entity's clusters but not necessarily the same individuals. In addition, we should insist on the fact that data generation for this experiment is entity 1 oriented. Thus, similarity between E2 individuals of same clusters is not guaranteed, which can lead to some variability in results.

Conclusion and Perspectives

In this paper, we have proposed a relationship-oriented partition function learning algorithm, based on clustering methods for PRM with Reference Uncertainty models. We have seen that it allows for more accurate results whenever the relationship topology is not just a consequence of entity attributes values, and that it has comparable results to the currently used partition functions learning methods whenever attributes fully explain the relationships entities. Note that many clustering methods could be used to follow the approach. A recent overview on clustering can be found in (Aggarwal and Reddy 2014). We chose the so-called non-negative matrix factorization (NMF) approach here, since it is very generalizable, customizable and extended in many ways. We can cite as examples 3-factors NMF for co-clustering (Ding et al. 2006) or NMF with Laplacian regularization (Gu and Zhou 2009). Note that performance could



		k & learning method											
		2 & OPTI	2 & NMF	2 & CP	p	4 & OPTI	4 & NMF	4 & CP	p	16 & OPTI	16 & NMF	16 & CP	p
nEi	25	-638 [0]	-637 [0]	-665 [0]	OK	-1475 [0]	-1471 [0]	-1433 [0]	KO	-2031 [0]	-2025 [20.9]	-1972 [0]	KO
	50	-3251 [0]	-3275 [7.21]	-3330 [0]	OK	-6752 [0]	-6720 [12.08]	-6709 [0]	KO	-7763 [0]	-7750 [37.78]	-7613 [0]	KO
	100	-14417 [0]	-14306 [25.52]	-15219 [0]	OK	-26274 [0]	-26088 [22.52]	-26301 [0]	OK	-37669 [0]	-37805 [77.26]	-37546 [0]	KO
	200	-70583 [0]	-70394 [78.66]	-71831 [0]	OK	-110531 [0]	-109909 [121.51]	-111050 [0]	OK	-135734 [0]	-136246 [224.72]	-135974 [0]	KO

		k & learning method											
		2 & OPTI	2 & NMF	2 & CP	p	4 & OPTI	4 & NMF	4 & CP	p	16 & OPTI	16 & NMF	16 & CP	p
nEi	25	-1149 [0]	-1138 [0]	-1169 [0]	OK	-1894 [0]	-1897 [6.08]	-1888 [0]	KO	-1294 [0]	-1302 [2.8]	-1275 [0]	KO
	50	-6404 [0]	-6361 [0]	-6377 [0]	OK	-7994 [0]	-7917 [18.22]	-7887 [0]	KO	-6947 [0]	-6754 [108.10]	-6631 [0]	KO
	100	-33853 [0]	-33820 [0]	-33658 [0]	KO	-35538 [0]	-35272 [45.85]	-35538 [0]	OK	-40859 [0]	-40568 [60.72]	-40664 [0]	KO
	200	-151323 [0]	-151058 [0]	-150746 [0]	KO	-152176 [0]	-151699 [98.37]	-152073 [0]	OK	-104835 [0]	-104183 [59.78]	-104957 [0]	OK

Figure 2: Log likelihood means (and standard deviations) and results of significance test p for: (top) NMF-favorable data (bottom) CP-favorable data. Significance tests of the NMF $>$ CP hypothesis are noted OK if the p-value score is below 0.001.

also be improved by the use of fast NMF algorithms, as in (Wang et al. 2011).

It is important to note that our method is for now limited to binary relationships. This limitation could be removed, as for example through the use of Tensor Factorization techniques (Shashua and Hazan 2005).

References

Aggarwal, C. C., and Reddy, C. K. 2014. *Data Clustering: Algorithms and Applications*. CRC Press.

Banerjee, A.; Dhillon, I.; Ghosh, J.; Merugu, S.; and Modha, D. S. 2004. 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, 509–514. New York, NY, USA: ACM.

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

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

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

Getoor, L.; Friedman, N.; Koller, D.; and Taskar, B. 2001. Learning probabilistic models of relational structure. In *Proceedings of International Conference on Machine Learning (ICML)*, 170–177.

Getoor, L. 2001. *Learning Statistical Models from Relational Data*. Ph.D. Dissertation, Stanford.

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

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

Hoyer, P. O. 2004. Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.* 5:1457–1469.

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

Koller, D., and Pfeffer, A. 1998. Probabilistic frame-based systems. In *Proceedings of the National Conference on Artificial Intelligence*, 580–587. John Wiley & Sons LTD.

Pathak, S.; Haynor, D.; Lau, C.; and Hawrylycz, M. 2007. Non-negative matrix factorization framework for dimensionality reduction and unsupervised clustering. *Insight Journal*.

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

Pfeffer, A. J., and Koller, D. 2000. Probabilistic reasoning for complex systems. Technical report.

Shashua, A., and Hazan, T. 2005. Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, 792–799. New York, NY, USA: ACM.

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

Vavasis, S. A. 2009. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization* 20(3):1364–1377.

Wang, H.; Nie, F.; Huang, H.; and Makedon, F. 2011. Fast nonnegative matrix tri-factorization for large-scale data coclustering. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, 1553–1558. AAAI Press.