Learning Infinite Hidden Relational Models

Zhao Xu ZHAO.XU@CAMPUS.LMU.DE
Institute for Computer Science, University of Munich, Germany

Volker Tresp VOLKER.TRESP@SIEMENS.COM
Kai Yu KAI.YU@SIEMENS.COM
Corporate Technology, Siemens AG, Information and Communications, Munich, Germany

Hans-Peter Kriegel KRIEGEL@DBS.IFI.LMU.DE
Institute for Computer Science, University of Munich, Germany

Abstract

Relational learning analyzes the probabilistic constraints between the attributes of entities and relationships. We extend the expressiveness of relational models by introducing for each entity (or object) an infinite-state latent variable as part of a Dirichlet process (DP) mixture model. It can be viewed as a relational generalization of hidden Markov random field. The information propagates in the intern-connected network via latent variables, reducing the necessary for extensive structure learning. For inference, we explore a Gibbs sampling method based on the Chinese restaurant process. The performance of our model is demonstrated in three applications: the movie recommendation, the function prediction of genes and a medical recommendation system.

1. Introduction

Relational learning (Dzeroski & Lavrac, 2001; Raedt & Kersting, 2003; Wrobel, 2001; Friedman et al., 1999) is an object oriented approach that clearly distinguishes between entities (e.g, objects), relationships and their respective attributes and represents an area of growing interest in machine learning. Structural model selection in a relational system is however extensive due to the exponentially many features an attribute might depend on. From this point of view it is more advantageous to introduce for each entity a latent variable, which is only parent of the other attributes of the entity, and is the parent of attributes of relationships it participates. We refer it as *hidden relational model*, which can be viewed as a direct generalization of hidden Markov models used in speech or hidden Markov random fields used in computer vision (see, e.g. (Yedidia et al., 2005)). The ground network based on the model forms a relational network of latent variables, across which information can propagate. For example, the information of my grandfather can propagate to me via the latent variable of my father. Note, that there is no constraint to the relationships. That means a relationship can be binomial or multinomial, the number of entities involved in a relationship can be more than two. Since each entity class might have a different number of states in its latent variables, and the number varies with available information, it is natural to allow the model to determine the appropriate number of latent states in a self-organized way. This is possible by embedding the model in Dirichlet process (DP) mixture models, which can be interpreted as a mixture models with an infinite number of mixture components but where the model, based on the data, automatically reduces the complexity to an appropriate finite number of components. The combination of the hidden relational model and the DP mixture model is referred to as the *infinite hidden relational model* (IHRM), which can also be viewed as a generalization of nonparametric hierarchical Bayesian modeling to relational models (compare, (Xu et al., 2005)).

After presenting related work we will briefly introduce our preferred framework for describing relational models, i.e., the directed acyclic probabilistic entity relationship (DAPER) model. In Section 4 we describe the proposed models and in Section 5 we introduce a modified Chinese restaurant sampling process to ac-

Presented at the ICML Workshop on Open Problems in Statistical Relational Learning, Pittsburgh, PA, 2006.

commodate for the relational structure. Section 6 explains the inference. In the subsequent sections we describe experimental analysis on movie recommendation, function prediction of genes, and on a medical example. In Section 10 we will present conclusions.

2. Related Work

Our approach can be related to some existing work. (Getoor et al., 2000) refined probabilistic relational models with class hierarchies, which specialized distinct probabilistic dependency for each subclass. (Rosen-Zvi et al., 2004) introduced an author-topic model for documents. The model implicitly explored the two relationships between documents and authors and document and words. (Kemp et al., 2004) showed a relational model with latent classes, which strongly focuses on the discovery and interpretation of the clustering structure. Our model uses a similar latent structure but focuses on the improvement of predictive performance in the exploitation of relational information. In addition, their model only explores the relation between members in a single class, e.g. friendship between two persons, whereas IHRM represents and learns several entity classes and relational classes. Another extension of IHRM is that the relational attribute can be complex whereas in (Kemp et al., 2004), the relational attribute is restricted to simply model the existence of a relationship. (Carbonetto et al., 2005) introduced the nonparametric BLOG model, which specifies nonparametric probabilistic distributions over possible worlds defined by first-order logic. These models demonstrated good performance in certain applications. However, most are restricted to domains with simple relations. The proposed model goes beyond that by considering multiple related entities. In addition, the nonparametric nature allows the complexity of the model to be tuned by the model based on the available data set.

3. The DAPER Model

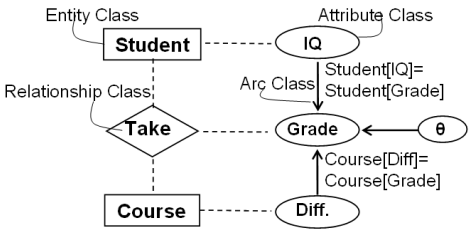

Figure 1. An example of DAPER model over university domain from (Heckerman et al., 2004).

The DAPER model (Heckerman et al., 2004) formulates a probabilistic framework for an entity relationship database model. The DAPER model consists of entity classes, relationship classes, attribute classes and arc classes, as well as local distribution classes and constraint classes. Figure 1 shows an example of a DAPER model for a universe of students, courses and grades. The entity classes specify classes of objects in the real world, e.g. Student shown as rectangles in Figure 1. The relationship class represents interaction among entity classes. It is shown as a diamond-shaped node with dashed lines linked to the related entity classes, e.g. the relationship Take(s, c) indicates that a student s takes a class c. Attribute classes describe properties of entities or relationships. Attribute classes are connected to the corresponding entity/relationship class by a dashed line. For example, associated with courses is the attribute class Course.Difficulty. The attribute class θ in Figure 1 represents the parameters specifying the probability of student's grade in different configurations. The arc classes shown as solid arrows represent probabilistic dependencies among corresponding attributes. For example, the solid arrow from Student.IQ to Course.Grade specifies the fact that student's grade probabilistically depends on student's IQ. For more details please refer to (Heckerman et al., 2004). A relationship class might have the special attribute Exist with Exist= 0 indicating that the relationship does not exist (Getoor et al., 2003). Given particular instantiations of entities and relationships a ground Bayesian network can be formed which consists of all attributes in the domain linked by the resulting arc classes.

4. Infinite Hidden Relational Models

4.1. Hidden Relational Models

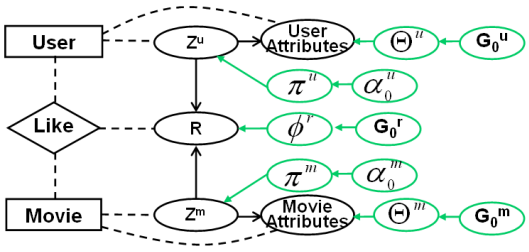

Figure 2. Infinite hidden relational model on movie recommendation.

An example of a hidden relational model is shown in Figure 2. The example shows a movie recommendation system with entity classes User and Movie and relationship class Like. Furthermore, there are the attributes UserAttributes, MovieAttributes and R

(rating) and various parameters and hyperparameters. The first innovation in our approach is to introduce for each entity a latent variable, in the example denoted as Z^u and Z^m. They can be thought of as unknown attributes of the entities and are the parents of both the entity attributes and the relationship attributes. The underlying assumption is that if the latent variable was known, both entity attributes and relationship attributes can be well predicted. The most important result from introducing the latent variables is that now information can propagate through the ground network of interconnected latent variables. Let us consider the prediction of the relationship attribute R. If both the associated user and movie have strong known attributes, those will determine the state of the latent variables and the prediction for R is mostly based on the entity attributes. In terms of a recommender system it is referred to content-based recommendation. Conversely, if the known attributes are weak, then the states of the latent variables for the user might be determined by the relationship attributes in relations to other movies and the states of those movies' latent variables. With the same argument, the prediction about movies can be done in the equivalent way. So by introducing the latent variables, information can globally distribute in the ground network defined by the relationship structure. This reduces the need for extensive structural learning, which is particularly difficult in relational models due to the huge number of potential parents. A similar propagation of information can be observed in hidden Markov models in speech systems or in the hidden Markov random fields used in image analysis (Yedidia et al., 2005). In fact the hidden relational model can be viewed as a generalization of both for relational structures. Note that the relational attribute R is not restricted to two states, e.g. the ratings ranged from 1 to 5. In addition, it is also possible that the number of entity classes participating a relationship is more than two.

We now complete the model by introducing the parameters. First we consider the user parameters. Assume that Z^u has K^u states and that $\pi^u = (\pi^u_1, \ldots, \pi^u_{K^u})$ are multinomial parameters with $P(Z^u = k) = \pi^u_k$ ($\pi^u_k \geq 0, \sum_k \pi^u_k = 1$). π^u is drawn from a Dirichlet prior with $\pi^u \sim \mathrm{Dir}(\cdot|\alpha^u_0/K^u, \ldots, \alpha^u_0/K^u)$. User attributes are assumed to be discrete and independent given Z^u. Thus, a particular user attribute A^u with S^u states is a sample from a multinomial distribution with $P(A^u = s) = \theta^u_s$

$$(\theta^u_1, \ldots, \theta^u_{S^u}) \sim G^u_0 = \mathrm{Dir}(\cdot|\beta^{u*}_1, \ldots, \beta^{u*}_{S^u}).$$

It is also convenient to re-parameterize

$$\beta^u_0 = \sum_{s=1}^{S^u} \beta^{u*}_s \quad \beta^u_s = \frac{\beta^{u*}_s}{\beta^u_0} \;\; s = 1, \ldots, S^u$$

and $\beta^u = \{\beta^u_1, \ldots, \beta^u_{S^u}\}$. In the application, we assume a neutral prior with $\beta^u_s = 1/S^u$, which represents our prior belief in the fact that the multinomial parameters should be equal. β^u_0 is a parameter indicating how strongly we believe that the prior distribution should be true. The parameters for the entity class Movie are defined in an equivalent way. Note, that for the relationship attribute R, there is a multinomial parameter ϕ^r for each of $K^u \times K^m$ configurations, and $\phi^r \sim G^r_0 = Dir(\cdot|\beta^r_0/S^r, \ldots, \beta^r_0/S^r)$, S^r is the number of states of R.

4.2. Infinite Hidden Relational Models

The latent variables play a key role in our model. In many applications, we would expect that for the latent variable of each entity class there is different number of states being suitable to the complexity of the data. Consider again the movie recommendation system. With little information about past ratings all users might look the same (movies are globally liked or disliked), with more information available, one might discover certain clusters in the users but with an increasing number of past ratings the clusters might show increasingly detailed structure ultimately indicating that everyone is an individual. It thus makes sense to permit an arbitrary number of latent states by embedding the model in a Dirichlet process mixture model. The combination is the *infinite hidden relational model*. The advantage is the model can decide itself about the optimal number of states for the latent variables. In addition, the model can now also be viewed as a direct generalization of a nonparametric hierarchical Bayesian approach (see, e.g. (Teh et al., 2004; Jordan, 2005; Tresp, 2006). For our discussion is suffices to say that we obtain an infinite hidden relational model by simply letting the number of states approach infinity, $K^u \to \infty$, $K^m \to \infty$. Although a model with infinite numbers of states and parameters cannot be represented, it turns out that sampling in such model is elegant and simple, as shown in the next section. In the Dirichlet mixture model, α_0 determines the tendency of the model to either use a large number or a small number of states in the latent variables, which is also apparent from the sampling procedures described below.

5. Sampling in the Infinite Hidden Relational Model

Although a Dirichlet process mixture model contains an infinite number of parameters and states, the sampling procedure only deals with a growing but finite representation. This sampling procedure is based on

the Chinese restaurant process (CRP) where a state of a latent variable is identified as a component, i.e., a table in a restaurant. We will now describe how the CRP is applied to the infinite hidden relational model. The procedure differs from the standard CRP by the sampling of the *relational attribute* where two CRP processes are coupled. Let the number of entity classes be C, and let G_0^c and α_0^c denote the base distribution and concentration parameter for entity class c, respectively. Then the sampling for all the entity and relation attributes is as follows:

1. For the first entity in each entity class c,
$$Z_1^c = 1; \quad \theta_1^c \sim G_0^c; \quad A_1^c \sim Mult(\cdot|\theta_1^c),$$
2. For each relationship class r between two entity classes, draw $\phi_{1,1}^r \sim G_0^r$ and $R_{1,1}^r \sim Mult(\cdot|\phi_{1,1}^r)$,
3. Assume that for each entity class c, N^c entitis have been generated, and K^c components appear, and N_k^c entities are assigned to the component k. The new entity $i = N^c + 1$:
   (a) $Z_i^c = k$ with probability $\frac{N_k^c}{N^c + \alpha_0^c}$ and:
      i. $A_i^c \sim Mult(\cdot|\theta_k^c)$,
      ii. for each relationship class r between entity classes c and c', $R_{i,j}^r \sim Mult(\cdot|\phi_{k,h}^r)$, with $j = 1, \ldots, N^{c'}$ and $h = Z_j^{c'}$,
   (b) $Z_i^c = K^c + 1$ with probability $\frac{\alpha_0^c}{N^c + \alpha_0^u}$;
      i. Draw $\theta_{K^c+1}^c \sim G_0^c$, $A_i^c \sim Mult(\cdot|\theta_{K^c+1}^c)$,
      ii. For each relationship class r between entity classes c and c', draw $\phi_{K^c+1,\ell}^r \sim G_0^r$ i.i.d. with component indices $\ell = 1, \ldots, K^{c'}$, and $R_{i,j}^r \sim Mult(\cdot|\phi_{k,h}^r)$, with $j = 1, \ldots, N^{c'}$ and $h = Z_j^{c'}$,
      iii. $K^c \leftarrow K^c + 1$.

One can see that the proposed model also draw relational attributes for the relation classes. In order to avoid a cluttering of notation, we only consider relations between two entity classes. The generalization to relations involving more than two entity classes is straightforward. The distribution G^r can take different forms for different relations.

6. Inference based on Gibbs Sampling

The previous procedure generates samples from the generative model. Now we consider sampling from a model given data, i.e. given a set of entity attributes, relational attributes. The goal is now to generate samples of the parameters θ^c, ϕ^r, and the latent variables Z^c, which allows us to then make predictions about unknown attributes. We exploit Gibbs sampling inference based on the CRP. Note, that since the attributes appear as children, unknown attributes can be marginalized out and thus removed from the model, greatly reducing the complexity. Although the DP mixture model contains an infinite number of states, in the Gibbs sampling procedure only a finite number of states is ever occupied, providing an estimate of the true underlying number of components (Tresp, 2006; Jordan, 2005).

In order to avoid a cluttering of notation, the Gibbs sampling inference is illustrated in the movie recommendation example. We assume that users are assigned to the first K^u states of Z^u and movies are assigned to the first K^m states of Z^m. We can do this without loss of generality by exploiting exchangeability. Note, that $K^u \leq U$ and $K^m \leq M$. If during sampling a state becomes unoccupied that state is removed from the model and indices are re-assigned. To simplify the description of sampling we will assume that this does not occur and that currently no state is occupied by exactly one item (just to simplify book keeping).

Gibbs sampling updates the assignment of users and movies to the states of the latent variable and resamples the parameters. In detail:

1. Pick a random user i. Assume that for N_k^u users, $Z^u = k$ without counting user i.
   (a) Then, we assign state $Z_i^u = k$ with probability proportional to
   $$P(Z_i^u = k|\{Z_{i'}^u\}_{i' \neq i}^{N^u}, D_i^u, \theta^u, \phi^r, Z^m) \propto$$
   $$N_k^u P(D_i^u|\theta_k^u, \phi_{k,*}^r, Z^m)$$
   (b) Instead, a new state $K^u + 1$ is generated with probability proportional to
   $$P(Z_i^u = K^u + 1|\{Z_{i'}^u\}_{i' \neq i}^{N^u}, D_i^u, \theta^u, \phi^r, Z^m) \propto$$
   $$\alpha_0^u P(D_i^u).$$
   (c) In the first case, the i-th user inherits the parameters assigned to state k: θ_k^u, $\phi_{k,1}^r, \ldots \phi_{k,K^m}^r$.
   (d) In the latter case: new parameters are generated following $P(\theta_{K^u+1}^u|D_i^u)$ and $P(\phi_{K^u+1,\ell}^r|D_i^u, Z^m), \ell = 1, \ldots, K^m$.
2. Pick a random movie j. Updates the latent variables of Z_j^m. The sampling is equivalent to the sampling of Z^u, above.

3. Occasionally (typically less often than the updates for the latent variables): Update the parameters, $\theta^u, \theta^m, \phi^r$ from posterior distribution based on all the data including sampled states for the latent variables.

In the algorithm, we used the following definitions (terms involving entity attributes or relationship attributes which are not known drop out of the equations)

$$P(D_i^u|\theta_k^u, \phi^r, Z^m) = P(A_i^u|\theta_k^u) \prod_{j=1}^{N^m} P(R_{i,j}|\phi^r_{k,Z_j^m})$$
$$P(\theta^u|D_i^u) \propto P(A_i^u|\theta^u)\, G_0^u(\theta^u)$$
$$P(\phi^r|D_i^u) \propto \prod_{j=1}^{N^m} P(R_{i,j}|\phi^r)\, G_0^r(\phi^r)$$

The algorithm easily generalizes to multiple relations as described in Section 8 and Section 9. A_i^u denotes all known attributes of user i. Definitions for the movies are equivalent. The most expensive term in the algorithm is in step 1 (a) which scales proportional to the number of known entity and relational attributes of the involved entity and is proportional to the number of occupied states.

We are currently exploring various sampling schemes and deterministic approximations. An extensive comparison will be available on the web shortly. The results reported in this paper were obtained by using the deterministic approximation described in (Tresp & Yu, 2004). First, we assume the number of components to be equal to the corresponding entities in the corresponding entity class. Then in the training phase each entity contributes to its own class only. Based on this simplification the parameters in the attributes and relations can be learned very efficiently. Note that this approximation can be interpreted as relational memory-based learning.

7. Experiment on MovieLens

We first evaluate our model on the MovieLens data (Sarwar et al., 2000). The task is to predict the preference of users. There are two entity classes (User and Movie) and one relationship class (Like). The User class has attribute classes such as Age, Gender, Occupation. The Movie class has attribute classes such as Published-year, Genres and so on. The relationship has an additional attribute R with two states: $R = 1$ indicates that the user likes the movie and $R = 0$ indicates otherwise. The model is shown as Figure 2. In the data set, there are totally 943 users and 1680 movies. We randomly select 765 users for training and 178 users for testing. In addition, user ratings on movies are originally recorded on a five-point scale. We transfer to be binary, *yes* if a rating higher than the average rating of the user, and vice versa. Model performance is evaluated using prediction accuracy. The base line system is content-based SVM where each training sample consists of a rating as a label and attributes of the corresponding user and movie as features. Prediction is then performed based on the learned SVM model. The experimental results are shown in Table 1. First we did experiments ignoring the attributes of the users and the items. We achieved an accuracy of 69.97% (E3). This is significantly better in comparison to approaches using one-sided collaborative filtering by generalizing across users (E1) leading to an accuracy of 64.22% or by generalizing across items (E2) leading to an accuracy of 64.66%. When we added information about the attributes of the users and the model, the prediction accuracy only improved insignificantly to 70.3% (E4): the reason is that the attributes are weak predictors of preferences as indicated by the bad performance of the baseline system (54.85% accuracy, E5) which is solely based on the attributes of the users and the items.

Table 1. The prediction accuracy of user preference

Method	Accuracy(%)
E1: Collaborative filtering 1	64.22
E2: Collaborative filtering 2	64.66
E3: Infinite hidden relational model without attributes	69.97
E4: Infinite hidden relational model	70.3
E5: Content based SVM	54.85

8. Experiment on Medical Data

The second experiment is concerned with a medical domain. The proposed model is shown in Figure 3(a). The domain includes three entity classes (Patient, Diagnosis and Procedure) and two relationship classes (Make: physician is making a diagnosis and Take:patient taking a procedure). A patient typically has both multiple procedures and multiple diagnoses. The Patient class has several attribute classes including Age, Gender, PrimaryComplaint. The DiagnosisAttributes contain the class of the diagnosis as specified in the ICD-9 code and the ProcedureAttributes contain the class of the procedure as specified in the CPT4 code. The relationship class Make (resp. Take) is modeled as existence uncertainty, thus has additional attribute with two states, $R^{pa,pr} = 1$ means that the patient received the procedure and $R^{pa,pr} = 0$ indicates otherwise. In the data, there are totally 14062 patients, 703 diagnoses and 367 procedures. We ran-

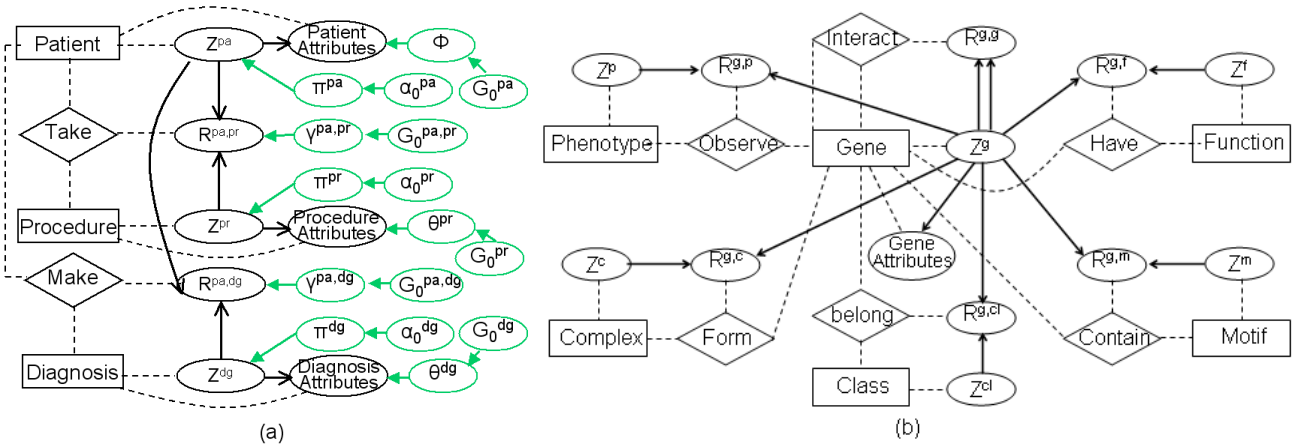

Figure 3. Infinite hidden relational model for (a) a medical database and (b) a gene database.

domly select 9979 patients as training and the others as testing. The infinite hidden relational model contains three DPs, one for each entity class. The DPs are coupled via the two relations. We compare our approach with two models. The first one is a relational model using reference uncertainty (Getoor et al., 2003) without a latent variable structure. The second comparison model is a content based Bayesian network. In this model, only the attributes of patients and procedures is considered. We test model performances by predicting the application of procedures. ROC curve is used as evaluation criteria. In the experiment we selected the top N procedures recommended by the various models. Sensitivity indicates how many percent of the actually being performed procedures were correctly proposed by the model. (1-specificity) indicates how many of the procedures that were not actually performed were recommended by the model. Along the curves, the N was varied from left to right as $N = 5, 10, \ldots, 50$.

In the experiment we predict a relation between a patient and a procedure *given her first procedure*. The corresponding ROC curves (averaged over all patients) for the experiments are shown in Figure 4. The infinite hidden relational model (E3) exploiting all relational information and all attributes gave best performance. When we remove the attributes of the entities, the performance degrades (E2). If, in addition, we only consider the one-sided collaborative effect, the performance is even worse (E1). (E5) is the pure content-based approach using the Bayesian network. The results show that entity attributes are a reasonable predictor but that the performance of the full model cannot be achieved. (E4) shows the results of relational model using reference uncertainty, which gave good results but did not achieve the performance of the infinite hidden relational model. Figures 5 shows the corresponding plots for a selected class of patients; patients with prime complaint *respiratory problem*. The results exhibit similar trends.

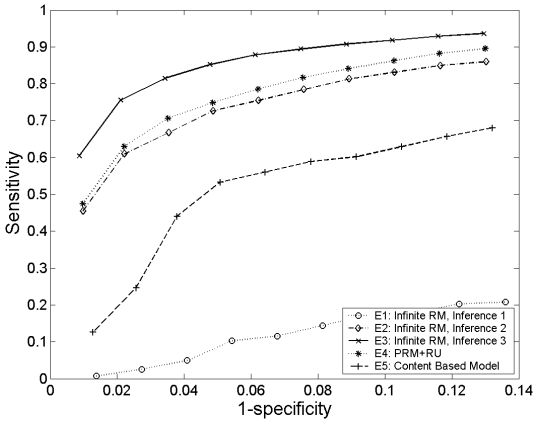

Figure 4. ROC curves for predicting procedures.

9. Experiment on Gene Data

The third evaluation is performed on the yeast genome data set of KDD Cup 2001 (Cheng et al., 2002). The task is to predict gene function based on information at the gene level and at the protein level. The data set consists of two relational tables that are produced from the original seven relational tables. One table specifies a variety of properties of genes or proteins. These properties include *chromosome, essential, phenotype, motif, class, complex and function*. *Chromosome* expresses the chromosome on which the gene appears.

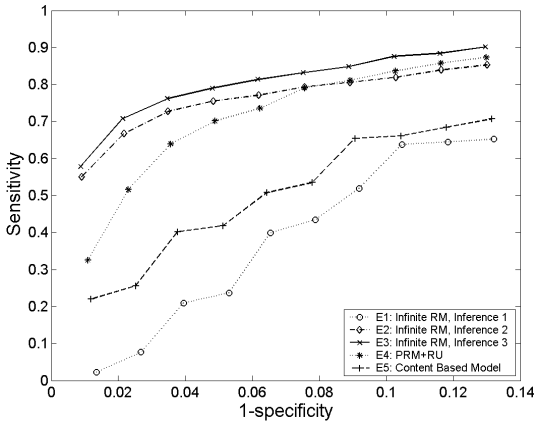

Figure 5. ROC curves for predicting procedures on a subset of patients with prime complaint *respiratory problem.*

Essential specifies whether organisms with a mutation in this gene can survive. *Phenotype* represents the observed characteristics of organisms with differences in this gene. *Class* means the structural category of the protein for which this gene codes. *Motif* expresses the information about the amino acid sequence of the protein. The value of property *complex* specifies how the expression of the gene can complex with others to form a larger protein. The other table contains the information about interactions between genes. A gene typically has multiple complexes, phenotypes, classes, motifs and functions, respectively but only one property essential and one property chromosome. An example gene is shown in Table 2. To keep the multi-relational nature of the data, we restore the original data structure. There are six entity classes (Gene, Complex, Phenotype, Class, Motif and Function) and six relationship classes (Interact: genes interact with each other, Have: genes have functions, Observe: phenotype are observed for the genes, Form: which kinds of complex is formed for the genes, Belong: genes belong to structural classes, Contain: genes contain characteristic motifs). Gene class has attribute classes such as Essential, Chromosome, etc. The attributes of other entity classes are not available in the data set. A hidden attribute is added into each entity class. All relationships are modeled as existence uncertainty. Thus each relationship class has additional attribute R with two states. The state of R indicates whether the relationship exists or not. The task of function prediction of genes is therefore transformed to the relationship prediction between genes and functions. The data set totally contains 1243 genes. A subset (381 genes) is withheld for testing in the KDD Cup 2001. The remaining 862 genes are provided to participants. In the data, there are 56 complexes, 11 phenotypes, 351 motifs, 24 classes and 14 functions. There are two main challenges in the gene data set. First, there are many types of relationships. Second, there are large numbers of objects, but only a small number of known relationships.

Table 2. An example gene

Attribute	Value
Gene ID	G234070
Essential	Non-Essential
Class	1, ATPases 2, Motorproteins
Complex	Cytoskeleton
Phenotype	Mating and sporulation defects
Motif	PS00017
Chromosome	1
Function	1, Cell growth, cell division and DNA synthesis 2, Cellular organization 3, Cellular transport and transprotmechanisms
Localization	Cytoskeleton

The proposed model applied to the gene data is shown in Figure 3(b). The existence of any relationship depends on the hidden states of the corresponding entities. The information about a variety of relationships of Gene is propagated via the hidden attribute of Gene. The model is optimized using 862 genes, and is applied on the testing data. The experiment results are shown in Table 3. There were 41 groups that participated in the KDD Cup 2001 contest. The algorithms include naive Bayes, k-nearest neighbor, decision tree, neural network, SVM, and Bayesian networks, etc. and technologies such as feature selection, boosting, cross validation, etc., were employed. The performance of our model is comparable to the best results. The winning algorithm is a relational model based on inductive logic programming (Krogel & Wrobel, 2001). As far as we know, that is best result so far. The infinite hidden relational model is only slightly worse (probably not significantly) if compared to the winning algorithm.

Table 3. Prediction of gene functions (%)

Model	Accuracy	True Positive Rate
Infinite model	93.18	72.8
Kdd cup winer	93.63	71.0

In the second set of experiments, we investigated the influence of a variety of relationships on the prediction of functions. We perform the experiments by ignoring a specific kind of known relationships. The result is shown in Table 4. The value of importance is proportional to the difference on the prediction accuracy. When a specific type of known relationship is ignored, the lower accuracy indicates higher importance of this type of relationship. One observation is that the most important relationship is *Complex*,

specifying how genes complex with another genes to form larger proteins. The second one is the interaction relationships between genes. This coincide with the lesson learned from KDD Cup 2001 that protein interaction information is less important in function prediction. This lesson is somewhat surprising since there is a general belief in biology that the knowledge about regulatory pathways is helpful to determine the functions of genes.

Table 4. The importance of a variety of relationships in function prediction of genes

Ignored relationships	Accuracy(%)	Importance
Complex	91.13	197
Interaction	92.14	100
Class	92.61	55
Phenotype	92.71	45
Attributes of gene	93.08	10
Motif	93.12	6

10. Conclusions and Extensions

We have introduced the infinite hidden relational model. The model showed encouraging results on a number of data sets. We hope that infinite hidden relational model will be a useful addition to relational modeling by allowing for flexible inference in a relational network reducing the need for extensive structural model search. Implicitly, we have assumed a particular sampling scheme, i.e., that entities are independently sampled out of unspecified populations. In this context our model permits generalization but it might fail if this assumption is not reasonable or if the sampling procedure changes in the test set. We have focussed on an explicit modeling of the relation between *pairs* of entities but our model can easily be generalized if more than two entities are involved in a relation. As part of our future work we will explore and compare different approximate inference algorithms.

References

Carbonetto, P., Kisynski, J., de Freitas, N., & Poole, D. (2005). Nonparametric bayesian logic. *Proc. 21st UAI*.

Cheng, J., Hatzis, C., Hayashi, H., Krogel, M., Morishita, S., Page, D., & Sese, J. (2002). KDD Cup 2001 report. *SIGKDD Explorations*, *3*, 47–64.

Dzeroski, S., & Lavrac, N. (Eds.). (2001). *Relational data mining*. Berlin: Springer.

Friedman, N., Getoor, L. Koller, D., & Pfeffer, A. (1999). Learning probabilistic relational models. *Proc. 16th IJCAI* (pp. 1300–1309). Morgan Kaufmann.

Getoor, L., Friedman, N., Koller, D., & Taskar, B. (2003). Learning probabilistic models of link structure. *Journal of Machine Learning Research*, *3*, 679–707.

Getoor, L., Koller, D., & Friedman, N. (2000). From instances to classes in probabilistic relational models. *Proc. ICML 2000 Workshop on Attribute-Value and Relational Learning: Crossing the Boundarie*.

Heckerman, D., Meek, C., & Koller, D. (2004). *Probabilistic models for relational data* (Technical Report MSR-TR-2004-30). Microsoft.

Jordan, M. I. (2005). Dirichlet processes, chinese restaurant processes and all that. *Tutorial at NIPS 2005*.

Kemp, C., Griffiths, T., & Tenenbaum, J. R. (2004). *Discovering latent classes in relational data* (Technical Report AI Memo 2004-019).

Krogel, M.-A., & Wrobel, S. (2001). Transformation-based learning using multirelational aggregation. *Proc. 11th ILP* (pp. 142–155). Springer.

Raedt, L. D., & Kersting, K. (2003). Probabilistic logic learning. *SIGKDD Explor. Newsl.*, *5*, 31–48.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P. (2004). The author-topic model for authors and documents. *Proc. 20th UAI* (pp. 487–494). AUAI Press.

Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2000). Analysis of recommender algorithms for e-commerce. *Proc. ACM E-Commerce Conference* (pp. 158–167). ACM.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2004). *Hierarchical dirichlet processes* (Technical Report 653). UC Berkeley Statistics.

Tresp, V. (2006). Dirichlet processes and nonparametric bayesian modelling. *Online tutorial*.

Tresp, V., & Yu, K. (2004). An introduction to nonparametric hierarchical bayesian modelling. In *Proc. hamilton summer school on switching and learning in feedback systems*, 290–312. Springer.

Wrobel, S. (2001). Inductive logic programming for knowledge discovery in databases. In S. Dzeroski and N. Lavrac (Eds.), *Relational data mining*, 74–101. Springer.

Xu, Z., Tresp, V., Yu, K., Yu, S., & Kriegel, H.-P. (2005). Dirichlet enhanced relational learning. *Proc. 22nd International Conference on Machine Learning (ICML 2005)* (pp. 1004–1011). ACM.

Yedidia, J., Freeman, W., & Weiss, Y. (2005). Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, *51*, 2282–2312.