A Bayesian Nonparametric Model for Joint Relation Integration and Domain Clustering

Dazhuo Li, Fahim Mohammad, Eric Rouchka
Department of Computer Engineering and Computer Science
University of Louisville
Louisville, KY 40292, USA
{dazhuo.li, m.fahim, eric.rouchka}@louisville.edu

Abstract—Relational databases provide unprecedented opportunities for knowledge discovery. Various approaches have been proposed to infer structures over entity types and predict relationships among elements of these types. However, discovering structures beyond the entity type level, e.g. clustering over relation concepts, remains a challenging task. We present a Bayesian nonparametric model for joint relation and domain clustering. The model can automatically infer the number of relation clusters, which is particularly important in novel cases where little prior knowledge is known about the number of relation clusters. The approach is applied to clustering various relations in a gene database.

Keywords—relational learning; clustering; Bayesian nonparametric

I. INTRODUCTION

The internal organization of the cell comprises many levels, including DNA, gene transcripts, proteins, metabolome. Genome databases have been built to store information related to biological components and their interactions across all these levels. For example, a gene database may contain a set of relations each having two participating domains (Relation, domain and entity are formally defined in Section II): Have(Gene, Function), Form(Gene, Complex), Belong(Gene, Class), Observe(Gene, Phenotype), Contain(Gene, Motif) and Interact(Gene, Gene), where function corresponds to a specific biological function, complex refers to a multi-protein complex, class represents a broadly defined set of similarly functioning proteins, phenotype describes observable traits, and motif refers to a common sequence pattern shared between sets of genes.

Genome databases offer unprecedented opportunity to improve our understanding of biological systems. There has been a great deal of previous work on integrative analysis of biological data for protein function prediction and regulatory network reconstruction. An essential step of learning from biological database is to infer the clustering structures over its domains and use the structures to predict associations between elements in the domains. For example, the functions of a gene may be predicted based on other genes with which the gene share a similar expression pattern, or based on other genes forming similar complexes; Similarly, to predict if a ligand binds to a protein, one may compare the ligand to all the other ligands known binding to the protein; or compare the protein to all known target proteins of the ligand. Recent relational learning model in machine learning includes the Infinite Relational Model (IRM) [1] and the Infinite Hidden Relational Model (IHRM) [2]. Both IRM and IHRM simultaneously cluster a variety of domains in a relation database, which may contain an arbitrary collections of relations, each of which may take any number of domains.

Presented a set of relations and domains, both IRM and IHRM would make a relation congruence assumption: each domain has one clustering to be inferred from the observed database; i.e. for each domain, similar elements w.r.t one relation would also be similar w.r.t any other relation involving this domain. For instance, given the gene database mentioned here, the models would assume that if a set of genes is inferred to have similar functions (w.r.t. Have(Gene, Function)), then these genes would also form similar complex (w.r.t. Form(Gene, Complex)) and contain similar motifs (w.r.t Contain(Gene, Motif)), etc. To make inference based on the models, it is important to make sure the relation congruence assumption being satisfied, which is usually achieved by employing expert knowledge to guide the selection of a subset of relations. When little knowledge about the degree of congruence between the relations is known a priori, it is rather appealing to test the statistical significance of the hypothesis by employing some mechanism of model selection and comparison. Many variable selection methods have been developed in traditional machine learning settings, however, to our best knowledge, few have been adapted
III. THE RELATIONAL OVER RELATIONS (ROR) MODEL

Let \( \mathbf{R} = \{R^1, \ldots, R^f\} \) represent a set of relations. Let \( \mathbf{T} = \{T^1, \ldots, T^j\} \) be a set of entity types involved in these relations. Let \( \mathbf{z} = [z_1, \ldots, z_f] \) be a vector of latent variables, where \( z_i = k \) indicates the \( i \)th relation \( R^i \) is assigned to the \( k \)th relation cluster. The task is to infer \( \mathbf{z} \). In ROR, a constraint is added among relations, which requires

\[
\bigcap_{i=1}^{f} \mathcal{T}(R^i) \neq \emptyset
\]

i.e. there exists a non-empty subset of \( \mathbf{T} \) that is involved in every relation in \( \mathbf{R} \).

In ROR, how the relations are clustered depends on the clustering of entities involved among these relations. Ideally, one may associate with each entity type a vector of latent variables representing a clustering assignment over the entity domain. However, this is true only when it is known \textit{a priori} that the same clustering assignment is shared by different relations, e.g., the same clustering of genes are shared through all relations in the gene database. In the proposed work, it is this hypothesis which the ROR model will be testing. Thus, the same clustering assignment over an entity domain is shared by different relations if and only if these relations are in the same relation cluster \textit{a posteriori}. More formally, with a little abuse of notations, define

\[
\mathbf{R}^k = \{R^i | z_i = k\}
\]

\[
\mathbf{T}^k = \mathcal{T}(\mathbf{R}^k) = \{\mathcal{T}(R^i) | R^i \in \mathbf{R}^k\}
\]

\[
\mathbf{U}^k = \{u^{kj} | T^j \in \mathbf{T}^k\}
\]

\[
\mathbf{U} = \{\mathbf{U}^k | k = 1, \ldots, K\}
\]

where \( K \) is the number of relation clusters; and \( u^{kj} \) represents, in the context of relation cluster \( k \), the clustering assignment over the domain of entity type \( T^j \). That is, assume the domain of \( T^j \) contains \( d \) elements, then \( u^{kj} = [u^{k1}_1, \ldots, u^{k1}_d] \) with \( u^{k1}_l = 1 \) indicates the \( l \)th element in the domain of \( T^j \) is assigned to the \( k \)th cluster. The ROR defines a generative model for the observed relations, the latent clustering of relations and the latent clusterings over entity type domains as

\[
\Pr(\mathbf{R}, \mathbf{z}, \mathbf{U}) = \Pr(\mathbf{R}|\mathbf{z}, \mathbf{U})\Pr(\mathbf{z}, \mathbf{U})
\]

\[
= \prod_{k=1}^{K} \Pr(\mathbf{R}^k|\mathbf{U}^k)\Pr(\mathbf{z}, \mathbf{U})
\]

where the relations are assumed conditionally independent given the clusterings, and the clusterings for each type under each relation cluster are also independent.
The prior on the cluster assignment vectors, \( \Pr(z) \) and \( \Pr(U) \), are described in Section III-A. The model on the relation data given the clustering assignments, \( \Pr(R^k|U^k) \), are described in Section III-B.

### A. Generating Clusterings with Chinese Restaurant Process

Chinese Restaurant Process (CRP) [4] defines an exchangeable distributions on partitions such that the joint distribution is invariant to the order in which observations are assigned to clusters. The name comes from a metaphor where there is a Chinese restaurant with an infinite number of tables, each of which can seat an infinite number of customers. The first customer enters the restaurant and sits at the first table. The rest of customers enter the restaurant one by one and sit at a table with other customers, or a new table by itself. In general, the \( N + 1 \)th customer either sits an already occupied table \( a \) with probability proportional to the number of already seated dinners \( n_a \), or sits at an empty new table with probability proportional to \( \gamma \). There is not a priori distinction between the unoccupied tables.

In CRP, the distribution over clusters for object \( i \), conditioned on the cluster assignments of objects \( 1, \ldots, i-1 \) is

\[
\Pr(z_i = a | z_1, \ldots, z_{i-1}) = \begin{cases} \frac{n_a}{i-1+\gamma} & n_a > 0 \\ \frac{\gamma}{i-1+\gamma} & a : \text{new cluster} \end{cases}
\]

where \( n_a \) is the number of objects already assigned to cluster \( a \), and \( \gamma \) is a parameter.

### B. Generating Relations with Infinite Relational Model

Given \( R^k \) defined by Equation 2, i.e. \( R^k \) is the \( k \)th relation cluster, then the distribution over \( R^k \) are induced by IRM:

\[
R^k \sim \text{IRM}(R^k)
\]

More specifically, let \( R \in R^k \), and \( u, v, \in U^k \), then the entry in \( R \) at an arbitrary index \([m,n] \), \( R_{m,n} \), is generated by the process:

\[
\begin{align*}
\mathbf{u} & \sim \text{CRP}(\gamma) \\
\mathbf{v} & \sim \text{CRP}(\gamma) \\
\theta_{a,b} & \sim \text{Beta}(\beta, \beta) \\
R_{m,n} & \sim \text{Bernoulli}(\theta_{u_m,v_n})
\end{align*}
\]

where \( u_m, v_n \) are latent variables in \( u, v \) respectively. For illustrative purposes, the discussion assumes that \( R \in \mathbb{R}^{d_1 \times d_2} \) with \( d_1, d_2 \) being the cardinality of the entity domains. Alternatively, if \( R \in \mathbb{R}^{d_1 \times d_2} \), then

\[
\begin{align*}
\theta_{a,b} & \sim \text{Gaussian}(0, \sigma^2_\theta) \\
R_{m,n} & \sim \text{Gaussian}(\theta_{u_m,v_n}, \sigma^2_R)
\end{align*}
\]

**input** \( R = \{R^1, \ldots, R^I\} \)

**initialize** Each \( R^i \) is in its own relation cluster, denoted by \( \mathbf{R}^i \). The number of relation clusters \( c = I \).

while \( c > 1 \) do

Find the pair \( (\mathbf{R}^i, \mathbf{R}^j) \) with the highest \( r_k \) and union them, i.e. \( \mathbf{R}^k = \mathbf{R}^i \cup \mathbf{R}^j \), \( \tau_k = (\tau_i, \tau_j) \), \( c = c - 1 \), and delete the two relation clusters \( \mathbf{R}^i, \mathbf{R}^j \).

end while

return An ROR mixture model where each tree node is a mixture component modeled by IRM.

Figure 1: A tree based approximation to the ROR model.

Several other possible parametric forms for the distribution \( f(R_{m,n}|\theta_{u_m,v_n}) \) are logistic with mean \( 1/(1 + \exp(-\theta_{u_m,v_n})) \); and Poisson with mean and variance \( \theta_{u_m,v_n} \).

### IV. Approximate Inference

Given the full set of observations \( \mathbf{R} \), the goal is to infer an optimal clustering \((z, U)\) and ultimately the posterior distribution of these latent variables. According to Bayes’ theorem, this posterior distribution is

\[
\Pr(z, U|\mathbf{R}) \propto \Pr(\mathbf{R}, z, U)
\]

where \( \Pr(\mathbf{R}, z, U) \) is the generative model described in Section III. Posterior samples can be drawn using a MCMC algorithm. In this section, an alternative approach is designed, which incorporates tree based approximations for DPM [3] and MCMC sampling method for IRM [1].

The algorithm is shown in Figure 1. It starts with each relation in its own cluster, and repeatly merge a pair of relation clusters yielding the highest posterior probability of merge, until all relations are merged into one relation cluster. Given a pair of relation cluster \( \mathbf{R}^i \) and \( \mathbf{R}^j \), the posterior probability of merging them, is defined as

\[
r_k = \frac{\pi_k \text{IRM}(\mathbf{R}^i \cup \mathbf{R}^j)}{\Pr(\mathbf{R}^i \cup \mathbf{R}^j|\tau_k)}
\]

where \( \text{IRM}(\mathbf{R}^k) \), \( \Pr(\mathbf{R}^i \cup \mathbf{R}^j|\tau_k) \), and \( \pi_k \) are estimated as follows. The marginal likelihood of an IRM,

\[
\text{IRM}(\mathbf{R}^k) = \sum_{U^k} \Pr(\mathbf{R}^k|U^k)\Pr(U^k),
\]

is neither analytically tractable nor suitable for numerical integration. It is estimated by the harmonic mean approach [5], which provides a simple approximation to a marginal likelihood using the MCMC posterior.
samples. The marginal probability of the relations at node \( k \) given a tree topology \( \tau_k \) is defined recursively

\[
\Pr(R^i \cup R^j | \tau_k) = \pi_k \Pr(R^i | \tau_j) \Pr(R^j | \tau_j) + (1 - \pi_k) \Pr(R^i | \tau_j) \Pr(R^j | \tau_j)
\]

where \( \tau_k \) is simply a topology with left subtree \( \tau_i \) and right subtree \( \tau_j \). A prior for agglomerative clustering is proposed in [3], which has similar property to the CRP prior:

\[
\begin{align*}
\alpha_k &= 1 & d_k &= \gamma \quad \text{if } \tau_k \text{ is a leaf} \\
\beta_k &= \frac{\gamma \Gamma(N_k)}{d_k} & d_k &= \gamma \Gamma(N_k) + d_i d_j \quad \text{else}
\end{align*}
\]

where \( \gamma \) is the concentration hyperparameter, and \( \Gamma(\cdot) \) is the Gamma function.

V. RESULTS

A. Dataset

This section demonstrates that the ROR jointly discovers the degree of associations among relations and the clustering structures of each entity domains involved in these relations, using the yeast gene data set from KDD Cup 2001 [6]. The table used in this study contains a variety of “attributes” of genes or proteins, including Function, Complex, Class, Phenotype, Motif, Localization, Essential and Chromosome. Each gene may be related to multiple elements from an attribute type, e.g. a gene may have multiple Functions, contain several Motifs, or belong to more than one Class. Rather than the original big table scheme, this relationship is better expressed in relation forms, e.g. Have(Gene, Function), Contain(Gene, Motif) and Belong(Gene, Class). In this example, the focus is on binary relations, though the model requires no restriction on a relation’s arity (e.g. a 3-arity relation Relation(Gene, Function, Phenotype)).

Information about the restored relational gene database is described in Table I: the Relation column specifies that, for each of the eight relations, the entity type (or domain) to which Gene is related; Since in this example scheme, each domain except Gene appeared in one and only one relation, the corresponding relation name are often omitted in our discussion. The Domain Size column specifies the size (number of possible elements) of the corresponding domain; In the Relation Type column, \( m : m \) and \( m : 1 \) represent “many-to-many” and “many-to-one” relationship respectively between participating domain; The last column shows several example elements from each domain. There are overall 862 genes used in this data.

<table>
<thead>
<tr>
<th>Relation(Gene, X)</th>
<th>Domain X</th>
<th>Relation Type</th>
<th>Domain X Value Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>14</td>
<td>( m : m )</td>
<td>Cell growth, cell division, cell rescue, defense, cell death and aging, ionic homeostasis, metabolism.</td>
</tr>
<tr>
<td>Complex</td>
<td>54</td>
<td>( m : m )</td>
<td>Acetolactate synthase, anaphase promoting complex, CAMP-dependent protein kinase, nucleosomal protein complex.</td>
</tr>
<tr>
<td>Class</td>
<td>24</td>
<td>( m : m )</td>
<td>Actin related proteins, actins, adaptins, molecular chaperones, motorproteins, nucleases.</td>
</tr>
<tr>
<td>Phenotype</td>
<td>11</td>
<td>( m : m )</td>
<td>Carbohydrate and lipid biosynthesis defects, cell cycle defects, mating and sporulation defects.</td>
</tr>
<tr>
<td>Motif</td>
<td>351</td>
<td>( m : m )</td>
<td>PS00012, PS00013, PS00014, PS00017, PS00018.</td>
</tr>
<tr>
<td>Localization</td>
<td>15</td>
<td>( m : 1 )</td>
<td>Cell wall, cytoplasm, cytoskeleton, endosome, integral membrane, lipid particles, mitochondria.</td>
</tr>
<tr>
<td>Essential</td>
<td>3</td>
<td>( m : 1 )</td>
<td>Essential, non-Essential, ambiguous-Essential.</td>
</tr>
<tr>
<td>Chromosome</td>
<td>16</td>
<td>( m : 1 )</td>
<td>1, 2, ..., 16.</td>
</tr>
</tbody>
</table>

B. Clustering Domains

Figure 2 displays the observed data in a heatmap, where a black entry (value of 1) indicates a known relationship existing between the corresponding elements in the data matrix. The elements assigned to different clusters are separated by a blue line. Figures in the top are the clustering inferred by assuming that none of the relations are congruent \( a \) \( priori \) to each other, while figure in the bottom are the results assuming all relations being congruent \( a \) \( priori \). In the top ones, very few clusters are discovered: when using Have(Gene, Function) or Observe(Gene, Phenotype), only two gene clusters are discovered; when using any of the other relations, all genes are assigned to the same cluster. In the bottom, there are more inferred gene clusters (6 gene clusters); Furthermore, more clusters are discovered for domains other than the gene domain as well. This phenomena is as expected because this relation data set are highly sparse, i.e. most of the pairs (gene, element) are either not related or their relations are missing, therefore genes (or other domains) do not have enough information to differentiate with each other. Combining multiple relations encourages information to be shared. The results from the ROR model, shown in Figure 3 and discussed in Section V-C, provides
Figure 2: (Top) Domain clusterings inferred by assuming relations are independent a priori. (Bottom) Domain clusterings inferred by assuming all relations are dependent a priori. Entries colored black represent the existing of a relationship between the two involved elements. Blue lines separates domain clusters.

a guideline for choosing relations for different study: while Form(Gene, Complex) provides more information for predicting the function of genes (the relation Have(Gene, Function)) than Contain(Gene, Motif) does, the latter relation describes genes from a much different perspective than other relations.

C. Clustering Relations

To apply the proposed model to this database, each relation is associated with a latent variable \( z_i \) indicating the group to which the \( i \)th relation (or equivalently, relation name) is assigned to. Meanwhile, for each relation group \( k \) and for each type \( j \) involved in the relation group, there is a latent variable associated with each element in that domain. For example, assume the first two relations in Table I (Have(Gene, Function) and Form(Gene, Complex)) both are assigned to the first relation group, then \( z_1 = z_2 = 1 \). There is a latent vector \( u \) for Gene, indicating the clustering of elements in the domain; Similarly, both Function and Complex have their corresponding latent vectors. Further, if relation Contain(Gene, Motif) and At(Gene, Chromosome) both are assigned to another relation group, then there is be another latent vector \( v \) for Gene, which is independent of \( u \). A goal of the model is to infer these latent variables.

Figure 3(a) shows the degree of associations among the relations inferred from ROR given the observed data. It can be seen that Have(Gene, Function) are highly related to most of the relations except At(Gene, Chromosome) and Contain(Gene, Motif). This indicates that, given the observations contained in the data set, besides Have(Gene, Function) itself, Form(Gene, Complex) gives the most information about whether a gene has a particular function, followed by Belong(Gene, Class), Observe(Gene, Phenotype), Loc(Gene, Localization) and Is(Gene, Essential). This result is concordant with the discovery using the same data set in [2], where the author showed that the accuracy of gene function prediction deceased the most if Complex, Class are removed from their model, while decreased the least when Motif are removed from the model.

A potential alternative method for comparing relations could be comparing the independently inferred clusterings of common domains shared in the relations, using clustering comparison methods such as Adjusted Rand Index [7]. To compare the effectiveness of this approach to the ROR, a clustering of the Gene domain is inferred independently based on each of the eight relations. The adjusted Rand Index (ARI) between these Gene clusterings are plotted in Figure 3(b), where almost all ARI values are zero, suggesting that the concordance between these relations is almost by chance. This conclusion is contradict to known studies on the same data set [2], [6].

VI. DISCUSSION AND FUTURE WORK

We proposed a Bayesian nonparametric model for joint relation clustering and domain clustering. Given a set of relations and domains, the model tries to answer the important questions of which relations should be integrated for domain clustering. We demonstrated its usage with a gene database. We showed that integrative analysis provides better domain clusterings, but a systematic way of selecting relations is important. The hierarchical clustering of relations inferred from the proposed model provides a guideline for the procedure of integration.

The harmonic mean identity is used to approximate
the marginal likelihood:

$$IRM(R^k) \approx \left\{ \frac{1}{S} \sum_{s=1}^{S} \Pr(R^k|U^k(s)) \right\}^{-1}$$

where $U^{k(1)}, \ldots, U^{k(S)}$ are $S$ samples drawn from the posterior distribution $\Pr(U^k|R^k)$. The decision of using the harmonic mean estimator is due to its simplicity. However, the estimator can have infinite variance. A stabilized version of the estimator is described in [8]. In [9], the proposed path sampling approach generalizes the thermodynamic integration originated from theoretical physics and involves a sequence of intermediate distributions bridging prior and posterior. The estimation of marginal likelihood remains a central problem in Bayesian inference. The proposed model requires that at least one common entity type is involved in all the relations (Equation 1). This assumption allows one to construct efficient probabilistic models much simpler than it would have been. While the model has profound applications, the assumption can be violated in many situations. In such case, expert knowledge and more sophisticated model will be necessary, though the proposed work may serve as a building block for more complicated model. These challenging tasks will be the focus of future work.

ACKNOWLEDGMENT

This work was supported in part by DOE grant DE-EM0000197 and NIH Grants P20RR16481, P20RR16481S1 and P30ES014443. Its contents are solely the responsibility of the authors and do not represent the official views of DOE, NCRR, NIEHS, or NIH.

REFERENCES


