Proc. 17th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2013), Lecture Notes in Computer Science, Vol. 7819, 147-159, 2013

An Extension of the Infinite Relational Model Incorporating Interaction Between Objects

Iku Ohama[1], Hiromi Iida[1], Takuya Kida[2], and Hiroki Arimura[2]

[1] Corporate R&D Division, Panasonic Corporation, Osaka 571-8501, Japan
{ohama.iku,iida.hiromi}@jp.panasonic.com
[2] Division of Computer Science, Hokkaido University, Hokkaido 060-0814, Japan
{kida,arim}@ist.hokudai.ac.jp

Abstract. The *Infinite Relational Model* (IRM) introduced by Kemp *et al.* (Proc. AAAI2006) is one of the well-known probabilistic generative models for the co-clustering of relational data. The IRM describes the relationship among objects based on a stochastic block structure with infinitely many clusters. Although the IRM is flexible enough to learn a hidden structure with an unknown number of clusters, it sometimes fails to detect the structure if there is a large amount of noise or outliers. To overcome this problem, in this paper we propose an extension of the IRM by introducing a *subset mechanism* that selects a part of the data according to the interaction among objects. We also present posterior probabilities for running collapsed Gibbs sampling to learn the model from the given data. Finally, we ran experiments on synthetic and real-world datasets, and we showed that the proposed model is superior to the IRM in an environment with noise.

Keywords: relational data, co-clustering, generative model, subset selection.

1 Introduction

A *relational data* among *m* objects and *n* objects is a bipartite network on a set of *m* vertices and another set of *n* vertices, which describes the relationships among objects in social, physical, and other phenomena. Equivalently, a relational data is represented by a matrix with *m* rows and *n* columns. For example, POS data is a relational data between customers and items, and a friend list of a social network service (SNS) such as the Facebook is a relational data among users.

With the emergence of such large amounts of relational data, there has been an increase in the interest in methods that can efficiently discover hidden interaction patterns among objects from given relational data. For example, enterprises involved in e-commerce and SNS might want to know about the following relationships:

- Which type of items does a customer purchase using e-commerce?
- Which other users are in a relationship with a SNS user?
- To which user does another user re-tweet when communicating on Twitter?

Clustering methods are among the most effective approaches to obtain answers to such questions, and several methods have been proposed so far [5, 3, 4, 16]. The *Infinite Relational Model* (IRM) [11] is a well-known and important generative model that represents processes for generating relational data. Co-clustering based on the model can produce a proper set of clusters that summarizes the relationships among objects. Moreover, the number of clusters is automatically estimated from the input data, even when the cluster structure and its size are unknown.

However, the IRM might fail to detect unknown structures when the data has a large amount of noise or the model can describe only a part of the data. Owing to the use of infinite clustering based on the Dirichlet Process (DP) [6], the IRM works to some extent, but it finds many small clusters to adapt itself to contradicting data. In fact, the problem of the co-clustering of real-world datasets is often difficult, because the data are noisy or sparse. For example, a spam blog that leaves comments randomly on other blogs has too many links. Such a noisy blog makes it difficult to analyze the relationship among blogs. Moreover, an inactive blog, which the author is not eager to write, has very few links. Such an insignificant blog also becomes an obstacle in finding important clusters. As we show later in Section 5, co-clustering with the IRM on such ill-formed data finds ineffective clusters.

To handle these ill-formed data, we incorporated a *subset selection mechanism* into the IRM and proposed a new relational model. In our model, the *relevance* of each object is parameterized by an individual Bernoulli parameter. The relevance indicates the degree of confidence with which an object forms informative relations coming from the latent cluster structure. For example, for POS data, an active customer tends to generate relevant relations with many items, as done by a well-known item as well. Their relevance becomes comparatively high in our model. Then, either a *relevant relation* or an *irrelevant relation* is generated stochastically for pair-wise objects according to the interaction of their relevance parameters.

Our contributions in this paper are summarized as follows:

- We proposed a new generative model, which is an extension of the IRM and incorporates a subset selection mechanism, whereby a subset of the relational data is determined by the interaction of the objects' relevances. By estimating the relevance of each object from the data, we diminished the effect of the irrelevant relations and performed co-clustering accurately.
- We derived posterior probabilities for running the Collapsed Gibbs Sampling [12] in order to infer the parameters of the model.
- We performed experiments on synthetic and real-world datasets. The experimental results for the synthetic datasets showed that our model significantly improved the performance of co-clustering compared with the IRM. For the real-world datasets, our model could successfully find major categories as clusters from the datasets. An estimated relevance of object can be viewed as the popularity or representativeness of the object within a cluster.

Therefore, the proposed method is effective in analyzing noisy relational data.

1.1 Related works

Hoff *et al.* [8] discussed an ill-formed problem with clustering vector data. They introduced a background distribution that describes irrelevant elements within the vector data, so that their model can find cluster robustly against noise based on a relevant subset of the data.

Ishiguro *et al.* [10] extended the IRM with a similar idea. They introduced switch variables to indicate whether an object is relevant for cluster analysis, or is an irrelevant troublesome one. In their model, only relationships among relevant objects are analyzed. That is, their model is an object-wise subset model. However, in some cases, it would not be preferable to select subset of objects for clustering target. For example, when we utilize co-clustering results for recommendation, we want to suggest the nearest cluster for any object. In our new model, the clustering target is selected in a relation-wise manner.

2 Relational Data and the Infinite Relational Model

In this section, we first define the relational data discussed in this paper. Then, we discuss the IRM, a generative model for co-clustering relational data.

Let $T^1 = \{O_i^1\}_{i=1}^{N1}$ and $T^2 = \{O_j^2\}_{j=1}^{N2}$ be the sets of objects. We define the relational data between T^1 and T^2 as $R : T^1 \times T^2 \to \{0,1\}$. If $R(i,j) = 1(0)$, we say that there is a *link* (*non-link*) between O_i^1 and O_j^2 [3]. For a purchase dataset, T^1 and T^2 are the sets of customers and items, respectively. We can represent customer i's purchase of item j by $R(i,j) = 1$, while $R(i,j) = 0$ indicates that customer i have not bought item j. The co-clustering problem on relational data is to estimate cluster assignments $\boldsymbol{z}^1 = \{z_i^1\}_{i=1}^{N1} \in C^1$ and $\boldsymbol{z}^2 = \{z_j^2\}_{j=1}^{N2} \in C^2$ based on given data $\boldsymbol{R}$, where $C^1 = \{1,2,\cdots,K\}$ and $C^2 = \{1,2,\cdots,L\}$ are the sets of cluster indices for T^1 and T^2, respectively.

The IRM proposed by Kemp *et al.* [11] is a generative model for relational data that can co-cluster objects based on the similarities of the relationships among the objects. In the IRM, the Dirichlet Process (DP) [6] is used as a prior distribution for the number of clusters. The DP is a nonparametric stochastic process that can be viewed as an infinite-dimensional Dirichlet distribution, and can generate any-dimensional multinomial distributions. Therefore, the IRM can adaptively estimate the number of clusters for the observed data. The generative model of the IRM is described as follows:

$$z_i^1 \,|\, \gamma^1 \sim \mathrm{CRP}(\gamma^1), \qquad z_j^2 \,|\, \gamma^2 \sim \mathrm{CRP}(\gamma^2), \tag{1}$$

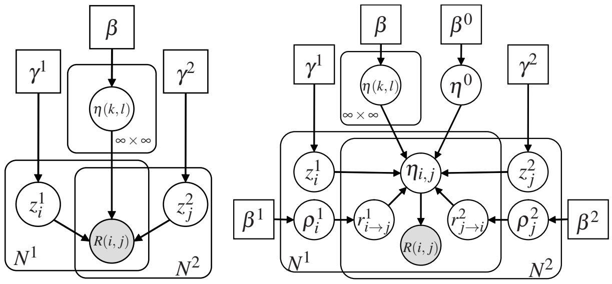
$$\eta(k,l) \,|\, \beta \sim \mathrm{Beta}(\beta,\beta), \tag{2}$$

$$R(i,j) \,|\, \eta(z_i^1,z_j^2) \sim \mathrm{Bernoulli}(\eta(z_i^1,z_j^2)), \tag{3}$$

where CRP$(\cdot)$ is the Chinese Restaurant Process (CRP) [2], which is one of the well-known constructive algorithms of DP; Beta$(\cdot,\cdot)$ is the beta distribution; and Bernoulli$(\cdot)$ is the Bernoulli distribution, respectively. Figure 1a shows the IRM graphically.

We will briefly review the above process. First, the cluster assignments z_i^1 and z_j^2 are given by CRPs (Eq. (1)), where γ^1 and γ^2 are the concentration parameters of the DP that controls the number of clusters to be generated. We denote the cluster assignments

[3] We focus on a 2-type ($T^1 \times T^2$) binary relationship in this paper, although several variations of relationships can be considered straightforwardly, such as discrete/continuous-valued relations and multi-type relations represented by a tensor.



(a) IRM (b) rdIRM

Fig. 1: Graphical representations of the generative models. Circle nodes denote variables, square nodes denote constants, shaded nodes denote observations, and round-edged squares indicate the dimensions of variables.

for all objects other than object i as $\boldsymbol{z}^1_{-i}$. When $\boldsymbol{z}^1_{-i}$ is given, the conditional probability $P(z_i^1 = k^* \mid \boldsymbol{z}^1_{-i}, \gamma^1)$ that z_i^1 is assigned to the cluster k^* by CRP is given as follows:

$$P(z_i^1 = k^* \mid \boldsymbol{z}^1_{-i}, \gamma^1) \propto \begin{cases} m^1_{-i,k^*} & (\text{if } m^1_{-i,k^*} > 0), \\ \gamma^1 & (\text{if } k^* \text{ is new cluster}), \end{cases} \tag{4}$$

where m^1_{-i,k^*} is the number of objects other than object i that are assigned to the cluster k^*. As Eq. (4) shows, the assignment z_i^1 basically depends on the probability proportional to the number m^1_{-i,k^*} of objects that belong to each cluster. However, new clusters are generated at the rate γ^1. Assume that $K \times L$ clusters ($C^1 = \{1, 2, \cdots, K\}$ and $C^2 = \{1, 2, \cdots, L\}$) have been generated for $T^1 \times T^2$. Then, from Eq. (2), a Bernoulli parameter $\eta(k,l)$ is given according to the beta prior for each pair of clusters $C^1 \times C^2$. The parameter $\eta(k,l)$ indicates the intensity of the relationship between an object in the cluster k and an object in the cluster l. Finally, the relation $R(i,j)$ is generated from the corresponding Bernoulli trial (Eq. (3)).

3 The Relevance-Dependent Infinite Relational Model

In this section, we present our new model, called *the Relevance-Dependent Infinite Relational Model* (rdIRM).

In real-world relationships, whether each relation is intentionally generated depends on the objects related to the relation. In the case of a purchase, a customer who knows about a large number of items will have a certain opinion about whether he needs these items. As a result, this customer will generate very important relations that are relevant to decide his cluster assignment. In contrast, a customer who knows only about a few items will have vague opinions. Thus, relations that are generated by this customer would be irrelevant. That is, such an irrelevant relation should not affect the co-clustering. For the items, it is reasonable to consider that similar properties exist in terms of popularity.

To model the above situation, for each object O_i^1 and O_j^2, we introduce *relevance parameters* $\rho_i^1, \rho_j^2 \in [0,1]$ that indicate the degree of confidence to generate the relevant relations. Then, we consider a generative mechanism in which each relation $R(i,j)$ between objects is generated from a mixture of the distribution inherent in a cluster $\eta(k,l)$ (foreground distribution) and the distribution common to the entire data η^0 (background distribution). We can construct such a mechanism as follows:

$$r_{i \to j}^1 \sim \text{Bernoulli}(\rho_i^1), \qquad r_{j \to i}^2 \sim \text{Bernoulli}(\rho_j^2),$$
$$r_{i,j} = f(r_{i \to j}^1, r_{j \to i}^2), \qquad \eta_{i,j} = r_{i,j} \times \eta(z_i^1, z_j^2) + (1 - r_{i,j}) \times \eta^0,$$

where $f(\cdot,\cdot)$ is an arbitrary Boolean function that returns 1 or 0. The above mechanism enables us to embed a *relevance-dependent subset selection* into the relational model: only the informative (relevant) relations are generated from the foreground distribution $\eta(k,l)$, and the background distribution η^0 describes the non-informative (irrelevant) part of the relational data. For example, when f is a logical sum, it corresponds to that we make the mixture rate as $1-(1-\rho_i^1)(1-\rho_j^2)$. When f is a logical product, the mixture rate becomes $\rho_i^1 \times \rho_j^2$. The other logical functions work similarly.

To summarize, the generative process for the rdIRM is defined as follows:

$$z_i^1 \,|\, \gamma^1 \sim \text{CRP}(\gamma^1), \qquad z_j^2 \,|\, \gamma^2 \sim \text{CRP}(\gamma^2), \tag{5}$$
$$\eta(k,l) \,|\, \beta \sim \text{Beta}(\beta,\beta), \qquad \eta^0 \,|\, \beta^0 \sim \text{Beta}(\beta^0,\beta^0), \tag{6}$$
$$\rho_i^1 \,|\, \beta^1 \sim \text{Beta}(\beta^1,\beta^1), \qquad \rho_j^2 \,|\, \beta^2 \sim \text{Beta}(\beta^2,\beta^2), \tag{7}$$
$$r_{i \to j}^1 \,|\, \rho_i^1 \sim \text{Bernoulli}(\rho_i^1), \qquad r_{j \to i}^2 \,|\, \rho_j^2 \sim \text{Bernoulli}(\rho_j^2), \tag{8}$$
$$r_{i,j} = f(r_{i \to j}^1, r_{j \to i}^2), \qquad \eta_{i,j} = r_{i,j} \times \eta(z_i^1, z_j^2) + (1 - r_{i,j}) \times \eta^0, \tag{9}$$
$$R(i,j) \,|\, \eta_{i,j} \sim \text{Bernoulli}(\eta_{i,j}). \tag{10}$$

Figure. 1b graphically represents this model.

Now, we will briefly explain the rdIRM process. First, the cluster assignments $\boldsymbol{z}^1$ and $\boldsymbol{z}^2$ are given as in the original IRM, (Eq. (5)). Second, the foreground distribution $\eta(k,l)$ and the background distribution η^0 are independently given from a beta prior (Eq. (6)). Third, the relevances ρ_i^1 and ρ_j^2 for O_i^1 and O_j^2, respectively, are given from beta priors (Eq. (7)). Fourth, the two switches $r_{i \to j}^1$ and $r_{j \to i}^2$ are given by a Bernoulli trial with corresponding relevances (Eq. (8)). Fifth, either the foreground $\eta(k,l)$ or the background η^0 is selected by the interaction of $r_{i \to j}^1$ and $r_{j \to i}^2$ via logical function f (Eq. (9)). Finally, the relation $R(i,j)$ is generated from the selected probability (Eq. (10)).

The difference between our rdIRM and the original IRM is that we modeled a generative process of noisy relationships by introducing objects' relevances and their interaction mechanism. That is, our rdIRM can co-cluster relational data based on a subset of relations that are relevant to underlying cluster structures.

When f is a logical sum, a relevant relation can be generated when at least one of the related objects O_i^1 or O_j^2 has high relevance. This models situations in which the relevant relationship between objects can be generated by a one-sided request, such as sending an e-mail or following a hyperlink on the Internet. When f is a logical product, the relevant relation is generated only when the objects cooperate with each other. This models situations in which an object that wants to have a relevant relation with another can be constrained from doing so. Of course, we can employ other logical functions for other interaction models.

4 Inference

We use the Collapsed Gibbs Sampler [12] to infer the parameters of the rdIRM [4]. Given $r_{i,j}$, the relational data $\boldsymbol{R}$ are separated into a foreground part and a background part; thus, the relevances ρ_i^1, ρ_j^2 and the link probabilities $\eta(k,l), \eta^0$ can be integrated out. Therefore, the inference of the rdIRM is performed by sampling the assignments $\boldsymbol{z}^1, \boldsymbol{z}^2$ and the switches $\boldsymbol{r}^1, \boldsymbol{r}^2$ one after the other. In this section, we only show the derived posteriors for running the Gibbs sampling below, because of the space limitation.

4.1 Sampling Cluster Assignments $\boldsymbol{z}^1, \boldsymbol{z}^2$

Because z_j^2 can be sampled in the same way as z_i^1, we concentrate on z_i^1. We can assume that the switch variables $\boldsymbol{r}$ ($\boldsymbol{r}^1$ and $\boldsymbol{r}^2$) have already been given before taking a sample of z_i^1, so that the cluster assignments are influenced only by the foreground part of the observations. Therefore, the conditional posterior for $z_i^1 = k^*$ is derived as follows:

$$P(z_i^1 = k^* \,|\, \boldsymbol{z}_{-i}^1, \boldsymbol{z}^2, \boldsymbol{r}, \boldsymbol{R}, \beta, \gamma^1) \propto \begin{cases} m_{-i,k^*}^1 \prod_{l \in C^2} \frac{B(m_r^{+i}(k^*,l)+\beta, \overline{m}_r^{+i}(k^*,l)+\beta)}{B(m_r^{-i}(k^*,l)+\beta, \overline{m}_r^{-i}(k^*,l)+\beta)} & (\text{if } m_{-i,k^*}^1 > 0), \\ \gamma^1 \prod_{l \in C^2} \frac{B(m_r^{+i}(k^*,l)+\beta, \overline{m}_r^{+i}(k^*,l)+\beta)}{B(\beta,\beta)} & (\text{if } m_{-i,k^*}^1 = 0), \end{cases} \tag{11}$$

Here, we use $B(\cdot,\cdot)$ to denote the beta function. Symbols m_r ($\overline{m}_r$) denote the numbers of links (non-links) in the foreground part of the observation, and are computed as follows:

$$m_r^{+i}(k^*,l) = \sum\sum_{\substack{s \in T^1, j \in T^2: \\ z_s^1 = k^* (z_i^1 := k^*), \\ z_j^2 = l}} (R(s,j) \times r_{i,j}), \; \overline{m}_r^{+i}(k^*,l) = \sum\sum_{\substack{s \in T^1, j \in T^2: \\ z_s^1 = k^* (z_i^1 := k^*), \\ z_j^2 = l}} ((1 - R(s,j)) \times r_{i,j}),$$

$$m_r^{-i}(k^*,l) = \sum\sum_{\substack{s \in T^1, j \in T^2: \\ z_s^1 = k^* (s \neq i), \\ z_j^2 = l}} (R(s,j) \times r_{i,j}), \; \overline{m}_r^{-i}(k^*,l) = \sum\sum_{\substack{s \in T^1, j \in T^2: \\ z_s^1 = k^* (s \neq i), \\ z_j^2 = l}} ((1 - R(s,j)) \times r_{i,j}).$$

Note that if $r_{i,j} = 1$ for all (i,j), Eq. (11) is equivalent to the original IRM's sampler.

4.2 Sampling Switch Variables $r_{i \to j}^1, r_{j \to i}^2$

As the sampling of $r_{j \to i}^2$ is done in the same way as the sampling of $r_{i \to j}^1$, we concentrate on $r_{i \to j}^1$. Given $\boldsymbol{z}^1$ and $\boldsymbol{z}^2$, we have a finite number $K \times L$ of clusters. Thus, the conditional posterior for $r_{i \to j}^1$ is derived as follows:

$$\begin{aligned} & P(r_{i \to j}^1 \,|\, \boldsymbol{z}^1, \boldsymbol{z}^2, \boldsymbol{r}_{-(i \to j)}^1, \boldsymbol{r}^2, \boldsymbol{R}, \beta, \beta^0, \beta^1) \\ & \propto P(R(i,j) \,|\, r_{i \to j}^1, \boldsymbol{r}_{-(i \to j)}^1, \boldsymbol{r}^2, \boldsymbol{R}_{-(i,j)}, \beta^0)^{1 - f(r_{i \to j}^1, r_{j \to i}^2)} \\ & \times P(R(i,j) \,|\, \boldsymbol{z}^1, \boldsymbol{z}^2, r_{i \to j}^1, \boldsymbol{r}_{-(i \to j)}^1, \boldsymbol{r}^2, \boldsymbol{R}_{-(i,j)}, \beta)^{f(r_{i \to j}^1, r_{j \to i}^2)} \\ & \times P(r_{i \to j}^1 \,|\, \boldsymbol{r}_{i \to (-j)}^1, \beta^1), \end{aligned} \tag{12}$$

[4] Approximative approaches such as variational inference [7] are preferable for handling large scale data. However, for the sake of accuracy, we used a sampling approach in this paper.

where $\boldsymbol{R}_{-(i,j)}$ denotes the whole set of $\boldsymbol{R}$ excluding $R(i,j)$. Similarly, $\boldsymbol{r}^1_{-(i\to j)}$ denotes the whole set of $\boldsymbol{r}^1$ without $r^1_{i\to j}$, and $\boldsymbol{r}^1_{i\to(-j)}$ denotes a vector of $r^1_{i\to t}$s that are related to object i without $r^1_{i\to j}$. The terms on the right-hand side of Eq. (12) are computed as follows:

$$
\begin{aligned}
P(R(i,j)\,|\,r^1_{i\to j},\boldsymbol{r}^1_{-(i\to j)},\boldsymbol{r}^2,\boldsymbol{R}_{-(i,j)},\beta^0) &= \frac{(m_{\overline{r}}^{-(i,j)}+\beta^0)^{R(i,j)}(\overline{m}_{\overline{r}}^{-(i,j)}+\beta^0)^{1-R(i,j)}}{m_{\overline{r}}^{-(i,j)}+\overline{m}_{\overline{r}}^{-(i,j)}+2\beta^0}, \\
P(R(i,j)\,|\,\boldsymbol{z}^1,\boldsymbol{z}^2,r^1_{i\to j},\boldsymbol{r}^1_{-(i\to j)},\boldsymbol{r}^2,\boldsymbol{R}_{-(i,j)},\beta) &= \frac{(m_r^{-(i,j)}(k,l)+\beta)^{R(i,j)}(\overline{m}_r^{-(i,j)}(k,l)+\beta)^{1-R(i,j)}}{m_r^{-(i,j)}(k,l)+\overline{m}_r^{-(i,j)}(k,l)+2\beta}, \\
P(r^1_{i\to j}\,|\,\boldsymbol{r}^1_{i\to(-j)},\beta^1) &= \frac{(n_{\boldsymbol{r}^1_i}^{-(i,j)}+\beta^1)^{r^1_{i\to j}}(n_{\overline{\boldsymbol{r}}^1_i}^{-(i,j)}+\beta^1)^{1-r^1_{i\to j}}}{N^2-1+2\beta^1},
\end{aligned}
$$

where $m_{\overline{r}}^{-(i,j)}$ and $\overline{m}_{\overline{r}}^{-(i,j)}$ denote the numbers of links and non-links, respectively, such that $r_{s,t}=0$ for all pairs $(s,t)\neq(i,j)$; $m_r^{-(i,j)}(k,l)$ and $\overline{m}_r^{-(i,j)}(k,l)$ denote the numbers of links and non-links, respectively, such that $z^1_s=k$, $z^2_t=l$ and $r_{s,t}=1$ for all pairs $(s,t)\neq(i,j)$; and $n_{\boldsymbol{r}^1_i}^{-(i,j)}$ and $n_{\overline{\boldsymbol{r}}^1_i}^{-(i,j)}$ denote the numbers of $r^1_{i\to t}=1\{t\neq j\}$ and $r^1_{i\to t}=0\{t\neq j\}$, respectively, within $\boldsymbol{r}^1_{i\to(-j)}$. Specifically, these counts are computed as follows:

$$
\begin{aligned}
n_{\boldsymbol{r}^1_i}^{-(i,j)} &= \sum_{t\in T^2:t\neq j}\left(r^1_{i\to t}\right), \qquad n_{\overline{\boldsymbol{r}}^1_i}^{-(i,j)} = \sum_{t\in T^2:t\neq j}\left(1-r^1_{i\to t}\right), \\
m_{\overline{r}}^{-(i,j)} &= \sum\sum_{\substack{s\in T^1,t\in T^2:\\(s,t)\neq(i,j)}}\left(R(s,t)\times(1-f(r^1_{s\to t},r^2_{t\to s}))\right), \\
\overline{m}_{\overline{r}}^{-(i,j)} &= \sum\sum_{\substack{s\in T^1,t\in T^2:\\(s,t)\neq(i,j)}}\left((1-R(s,t))\times(1-f(r^1_{s\to t},r^2_{t\to s}))\right), \\
m_r^{-(i,j)}(k,l) &= \sum\sum_{\substack{s\in T^1,t\in T^2:\\z^1_s=k,z^2_t=l,\\(s,t)\neq(i,j)}}\left(R(s,t)\times f(r^1_{s\to t},r^2_{t\to s})\right), \\
\overline{m}_r^{-(i,j)}(k,l) &= \sum\sum_{\substack{s\in T^1,t\in T^2:\\z^1_s=k,z^2_t=l,\\(s,t)\neq(i,j)}}\left((1-R(s,t))\times f(r^1_{s\to t},r^2_{t\to s})\right).
\end{aligned}
$$

5 Experiments

In this section, we present our experimental results. To clarify the effectiveness of our subset selection mechanism, the performance of our rdIRM is compared with that of the original IRM. Through all the experiments, we assumed that the priors of all the binary variables in the generative models were uniform (Beta$(1.0,1.0)$). In addition, we estimated the concentration parameters γ^1,γ^2 for the DPs assuming Gamma priors by sampling method presented in [8].

5.1 Experiments on Synthetic Datasets

We prepared 12 synthetic datasets. First, in accordance with the generative model of our rdIRM, we created five synthetic datasets, Data1(0.0), Data1(0.2), Data1(0.5), Data1(0.8), and Data1(1.0), where the numbers in parentheses indicate the background link probabilities η^0 for the datasets. We set the logical function f for the rdIRM to be a logical sum. The cluster assignments $\boldsymbol{z}^1$ and $\boldsymbol{z}^2$ were independently generated from fixed-dimensional multinomial distributions. The parameter values used for generating the datasets were $N^1 = N^2 = 200$, $\beta = (0.5, 0.5)$, and $\beta^1 = \beta^2 = (4.0, 3.0)$; the number of clusters were set as $K = 4$ and $L = 5$, and the parameters for the multinomials were $\boldsymbol{\pi}^1 = (0.4, 0.3, 0.2, 0.1)$ and $\boldsymbol{\pi}^2 = (0.33, 0.27, 0.20, 0.13, 0.07)$ for T^1 and T^2, respectively. Next, we also created five synthetic datasets in a similar manner (from Data2(0.0) to Data2(1.0)), except that we set the logical function f to be a logical product and we set both β^1 and β^2 to be $(4.0, 2.0)$. Finally, we created two datasets without background influences, (Data1(NULL) and Data2(NULL)). We applied the logical sum version of the rdIRM to Data1 and the logical product version to Data2.

We used three measures to evaluate clustering performance. One was the Adjusted Rand Index (ARI) [9], which is widely used for computing the similarity between true and estimated clustering results. The ARI takes a value in the range 0.0 – 1.0, and takes a value of 1.0 when a clustering result is completely equivalent to the ground truth. Another was the number of erroneous estimated clusters (EC). We computed the average of these measures for the two sets T^1 and T^2. The rest was the test data log likelihood (TDLL), which indicates the predictive robustness of a generative model; we hid 1.0% of the observation during inference (keeping it small so that the latent cluster structure did not change), and measured the averaged log likelihood such that a hidden entry would take the actual value. A larger value is better, and a smaller one means that the model overfits the data. Finally, we repeated the experiment 10 times for each dataset using different random seeds to find an overall average.

Table 1 lists the computed measures. In the case of every dataset, except Data1(NULL) and Data2(NULL), we confirmed that the rdIRM outperformed the IRM. In particular, the rdIRM maintained good performance for sparse ($\eta^0 \approx 0.0$) or dense ($\eta^0 \approx 1.0$) data. We also list in Table 2 the maximum a posteriori (MAP) estimations of the background probability $\bar{\eta}^0$ and the estimated ratios of the foreground for synthetic datasets, except Data1(NULL) and Data2(NULL). The ground truths of the foreground ratios (FRs) are 0.8197 for Data1 and 0.4622 for Data2. As the table shows, the rdIRM performs well in estimating the ground truths.

5.2 Experiments with Real-World Datasets

We applied the rdIRM to two real-world datasets. One was the "MovieLens" dataset[5], which contains a large number of user ratings of movies on a five-point scale. In our experiment, we created a binary relational dataset with a threshold that yields $R(i, j) = 1$ for ratings higher than 3 points and $R(i, j) = 0$ for all other ratings. That is, a relational value $R(i, j) = 1$ indicates that user i likes movie j. There are a total of 943 users and 1,682 movies in the dataset, and 3.5% of the relations are links. The other dataset was the "animal-feature" dataset [14], which includes relations between 50 animals and 85 features. Each feature is rated on a scale of 0–100 for each animal. We prepared the binary data with a threshold that yields $R(i, j) = 1$ for all ratings higher than the average

[5] http://movielens.umn.edu/

Table 1: ARI, EC, and TDLL on synthetic datasets.

	ARI		EC		TDLL	
Dataset	IRM	rdIRM	IRM	rdIRM	IRM	rdIRM
Data1(NULL)	**1.000**	0.999	**0.000**	0.030	-0.302	**-0.261**
Data1(0.0)	0.712	**0.999**	0.678	**0.022**	-0.410	**-0.315**
Data1(0.2)	0.806	**1.000**	0.480	**0.010**	-0.432	**-0.363**
Data1(0.5)	0.868	**0.993**	0.270	**0.090**	-0.459	**-0.405**
Data1(0.8)	0.834	**0.999**	0.388	**0.013**	-0.462	**-0.385**
Data1(1.0)	0.806	**0.999**	0.435	**0.025**	-0.425	**-0.330**
Data2(NULL)	**1.000**	0.996	**0.000**	**0.000**	-0.316	**-0.232**
Data2(0.0)	0.629	**0.980**	1.053	**0.020**	-0.424	**-0.196**
Data2(0.2)	0.627	**0.913**	0.735	**0.105**	-0.576	**-0.431**
Data2(0.5)	0.759	**0.930**	0.488	**0.105**	-0.614	**-0.526**
Data2(0.8)	0.724	**0.917**	0.738	**0.097**	-0.558	**-0.438**
Data2(1.0)	0.644	**0.981**	0.910	**0.083**	-0.390	**-0.183**

Table 2: Estimated background probabilities ($\bar{\eta}^0$) and the FRs.

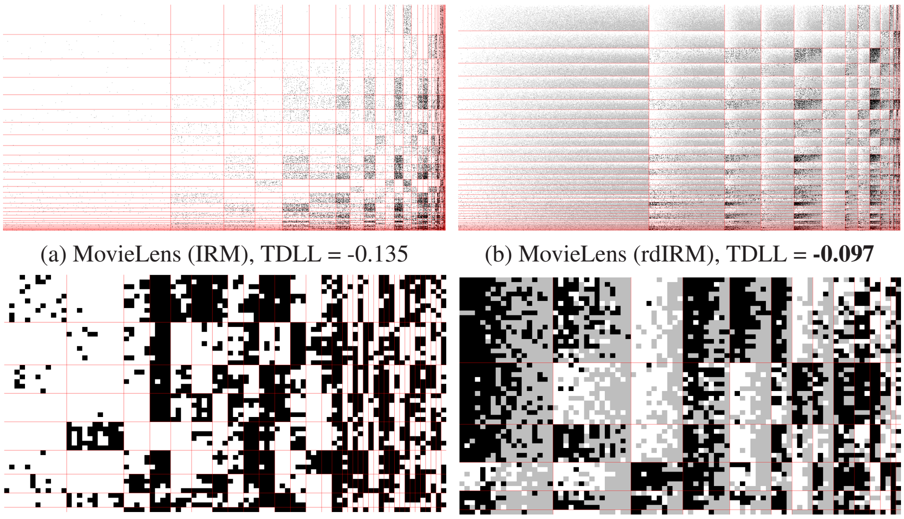
Dataset	$\bar{\eta}^0$	FR
Data1(0.0)	0.0085	0.8484
Data1(0.2)	0.1970	0.8462
Data1(0.5)	0.4531	0.8588
Data1(0.8)	0.7674	0.8607
Data1(1.0)	0.9876	0.8611
Data2(0.0)	0.0022	0.4884
Data2(0.2)	0.2139	0.4548
Data2(0.5)	0.5033	0.4658
Data2(0.8)	0.7845	0.4397
Data2(1.0)	0.9872	0.4654

of the entire set of ratings (20.79). That is, we used the relational value $R(i, j) = 1$ ($R(i, j) = 0$) to indicate that animal i has (does not have) feature j. In this dataset, 36.8% of the relations are links.

We used a logical sum version of the rdIRM for the MovieLens dataset and a logical product version for the animal-feature dataset. Our reason to use the former was that a user can watch any movie according to his or her preference, and similarly, movies are usually promoted independent of the users. Therefore, it seemed natural that the foreground (relevant relations) for the MovieLens dataset should be generated as per either the user's relevance ρ_i^1 or the movie's relevance ρ_j^2. On the other hand, animal features are acquired through evolution based on the specific type of animal. For example, aquatic features such as "swims" or "water" cannot be acquired by terrestrial animals. Therefore, the type of animal limits the features that it can acquire, and conversely, the type of feature limits the types of animals that are related to that feature. Therefore, we used the logical product version of the rdIRM for the animal-feature dataset.

Figure 2 shows the clustering results and the computed TDLL for these real-world datasets. Figure 3 shows color maps for the estimated foreground probabilities $\bar{\eta}(k,l)$. The background probabilities η^0 that the rdIRM estimated were 0.0000 for the MovieLens dataset and 0.0036 for the animal-feature dataset. It can be seen that the original IRM organized many non-informative cluster-blocks, because the IRM considered that all the relations were relevant for cluster analysis. In contrast, the rdIRM found more vivid cluster structures owing to the use of our subset selection mechanism, which selects an informative subset of relations via the interaction of the objects' relevances. The computed TDLLs show that the rdIRM predicts hidden entries more robustly than does the original IRM for both datasets.

The left side of Table 3 lists the examples of the movie clusters produced by the rdIRM for the MovieLens dataset. In the columns for the number of links and $\bar{\rho}_j^2$, it can be seen that $\bar{\rho}_j^2$ tends to increase with the number of links. This means that we can regard the relevances as an indication of the popularity of the movies within the cluster. On the other hand, the original IRM treats all the links and non-links as relevant, so that



(a) MovieLens (IRM), TDLL = -0.135

(b) MovieLens (rdIRM), TDLL = **-0.097**

(c) animal-feature (IRM), TDLL = -0.393

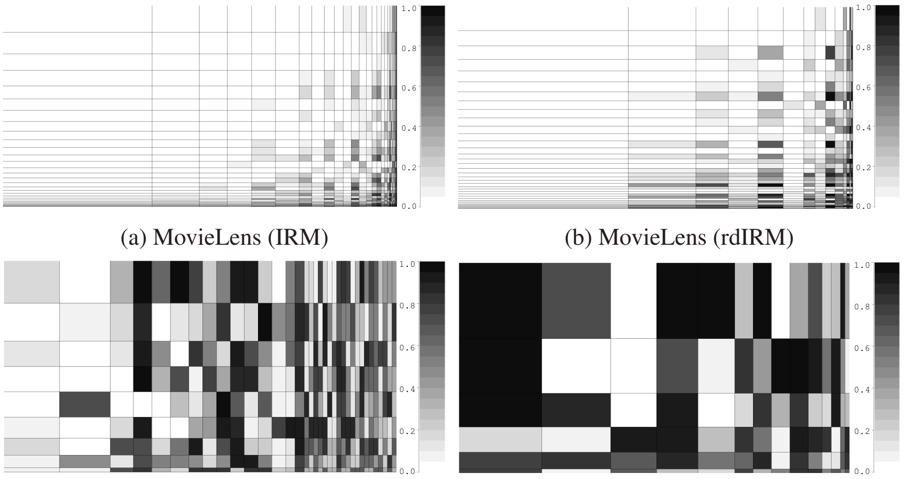
(d) animal-feature (rdIRM), TDLL = **-0.213**

Fig. 2: Clustering results for the real-world datasets. Black and white dots indicate links and non-links, respectively. In the rdIRM's results, gray dots indicate the areas that were estimated as background (irrelevant to cluster). Note that the objects within each cluster are sorted by descending order of the estimated relevances $\bar{\rho}_i^1$ and $\bar{\rho}_j^2$. "TDLL" is the computed test data log likelihood for each dataset.

the differences of the popularity of movies popularity affect the cluster assignment. The right side of Table 3 lists the examples of the feature clusters obtained by the rdIRM for the animal-feature dataset. As with the results for the MovieLens dataset, we can see that the estimated ρ_j^2 tends to increase with the number of links. One interesting result produced by the rdIRM is that representative features such as "swims," "water," "paws," "nestspot" and "meet" were found to have high relevance in their clusters. From these results, we can say that the relevances estimated by the rdIRM indicate the popularities or representativeness of the objects. Consequently, the rdIRM finds clusters in terms of major categories by introducing the relevance-dependent subset selection mechanism.

6 Conclusions

In this paper, we proposed a new probabilistic relational model called the Relevance-Dependent Infinite Relational Model (rdIRM), which is suitable for noisy relational data analysis. The rdIRM parameterizes objects' relevances and incorporates a relevance-dependent subset selection mechanism, so that the rdIRM can estimate objects' relevances, and can co-cluster noisy relational data selecting only relevant relations that are informative for co-cluster analysis.



(a) MovieLens (IRM)

(b) MovieLens (rdIRM)

(c) animal-feature (IRM)

(d) animal-feature (rdIRM)

Fig. 3: The estimated foreground link probabilities $\bar{\eta}(k,l)$.

Our experiments with synthetic datasets confirmed that the rdIRM can find proper clusters in a noisy relational data, especially, in sparse or dense data. Moreover, our experiments on real-world datasets confirmed that the clusters obtained by the rdIRM represent major categories and that the estimated relevances can be viewed as the popularity or representativeness of the objects.

Our future research plans include extending the rdIRM so that it can also estimate the logical function f, which was given statically in this paper. We are also interested in applying our relevance-based subset selection mechanism to more advanced relational models, such as the mixed (or multiple) membership models [1, 13], the hierarchical structure models [15], and the time-varying models [7].

References

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, 2008.

[2] D. Aldous. Exchangeability and related topics. In *Ecole d'Ete de Probabilities de Saint-Flour XIII*, pages 1–198. 1985.

[3] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proc. SIGKDD*, pages 269–274, 2001.

[4] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proc. SIGKDD*, pages 89–98, 2003.

[5] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proc. SIGKDD*, pages 126–135, 2006.

[6] T. S. Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

Table 3: Examples of the clusters obtained by the rdIRM. The first column lists the object (Title/Feature). The second column lists the number of links related to the object (LNKS). The third column lists the estimated relevance ($\bar{\rho}_j^2$). The fourth column lists the cluster indices that were obtained by the original IRM (ZIRM). The left side tables are for the MovieLens dataset. The right side tables are for the animal-feature dataset.

Movie cluster 6 (contains 6 movies.)			
Title	LNKS	$\bar{\rho}_j^2$	ZIRM
Star Wars	501	0.9111	28
Return of the Jedi	379	0.5534	9
Independence Day	228	0.0921	25
Star Trek	220	0.1905	25

Movie cluster 7 (contains 40 movies.)			
Title	LNKS	$\bar{\rho}_j^2$	ZIRM
Silence of the Lambs	344	0.9132	26
Pulp Fiction	294	0.7598	26
Usual Suspects	232	0.6233	20
Alien	223	0.5164	20
Terminator	217	0.5608	20
Seven(Se7en)	167	0.3376	15

Movie Cluster 2 (contains 35 movies.)			
Title	LNKS	$\bar{\rho}_j^2$	ZIRM
W.W. & the Chocolate F.	189	0.7196	27
Birdcage	154	0.4762	17
Truth About Cats & Dogs	148	0.3386	17
Happy Gilmore	74	0.0360	2
Kingpin	73	0.1196	2

Feature cluster 1 (contains 10 features.)			
Feature	LNKS	$\bar{\rho}_j^2$	ZIRM
swims	10	0.9808	2
water	10	0.9808	2
coastal	8	0.9231	2
arctic	9	0.8846	2
flippers	7	0.8077	2

Feature cluster 5 (contains 15 features.)			
Feature	LNKS	$\bar{\rho}_j^2$	ZIRM
paws	19	0.9615	27
nestspot	31	0.9423	20
claws	19	0.9038	22
small	23	0.7885	21

Feature cluster 6 (contains 8 features.)			
Feature	LNKS	$\bar{\rho}_j^2$	ZIRM
meat	20	0.9808	17
fierce	21	0.9231	17
hunter	17	0.8846	17
stalker	10	0.4808	16
scavenger	6	0.1538	1
flys	1	0.0769	1

[7] Wenjie Fu, Le Song, and Eric P. Xing. Dynamic mixed membership blockmodel for evolving networks. In *Proc. ICML*, pages 329–336, 2009.

[8] P. D. Hoff. Subset clustering of binary sequences, with an application to genomic abnormality data. *Biometrics*, 61(4):1027–1036, 2005.

[9] L. Hubert and P. Arabie. Comparing partitions. *J. of Classification*, 2(1):193–218, 1985.

[10] K. Ishiguro, N. Ueda, and H. Sawada. Subset infinite relational models. *J. Mach. Learn. Res. - Proceedings Track*, 22:547–555, 2012.

[11] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proc. AAAI*, volume 1, pages 381–388, 2006.

[12] J. S. Liu. The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem. *J. Am. Stat. Assoc.*, 89(427):958–966, 1994.

[13] M. Mørup, M.N. Schmidt, and L.K. Hansen. Infinite multiple membership relational modeling for complex networks. In *Proc. MLSP*, pages 1–6, 2011.

[14] D. N. Osherson, J. Stern, O. Wilkie, M. Stob, and E. E. Smith. Default probability. *Cognitive Science*, 15(2):251–269, 1991.

[15] D. Roy, C. Kemp, V. Mansinghka, and J. Tenenbaum. Learning annotated hierarchies from relational data. In *Proc. NIPS*, pages 1185–1192, 2006.

[16] M. M. Shafiei and E. E. Milios. Latent dirichlet co-clustering. In *Proc. ICDM*, pages 542–551, 2006.