Subset Infinite Relational Models

Katsuhiko Ishiguro **Naonori Ueda** **Hiroshi Sawada**
NTT Communication Science Laboratories, NTT Corporation. 619-0237 Kyoto, Japan.
{ishiguro.katsuhiko , ueda.naonori , sawada.hiroshi} @lab.ntt.co.jp

Abstract

We propose a new probabilistic generative model for analyzing sparse and noisy pairwise relational data, such as friend-links on social network services and customer records in online shops. Real-world relational data often include a large portion of non-informative pairwise data entries. Many existing stochastic blockmodels suffer from these *irrelevant* data entries because of their rather simpler forms of priors. The proposed model incorporates a latent variable that explicitly indicates whether each data entry is relevant or not to diminish bad effects associated with such irrelevant data. Through experiments using synthetic and real data sets, we show that the proposed model can extract clusters with stronger relations among data within the cluster than clusters obtained by the conventional model.

1 Introduction

Analysis of pairwise relational data, such as the customer records of purchases in online shops, friend-links on social networks, or bibliographic citations between scientific articles, is useful in many aspects. Many statistical models for relational data have been presented [Liben-Nowell and Kleinberg, 2003, Clauset et al., 2008, Zhu et al., 2009, Erosheva et al., 2004] in the literature. Among them, the stochastic block model (SBM) [Nowicki and Snijders, 2001] and the infinite relational model (IRM) [Kemp et al., 2006] perform simultaneous clustering on the row and column dimensions of a given pairwise relational data matrix. For example, in the case of customer records, the row and column correspond to users and items, respectively. The row and column clusters are interpreted as latent user groups and item topics, respectively. SBM requires specifying the number of clusters in advance, while IRM automatically estimates the number of clusters depending on the observed data. The usefulness of these models has already been established.

However, the real-world relational data are often noisy and sparse. For example, relations between users and shop items in online purchase records is very sparse: there are thousands of items and users, but only very few item-user pairs are observed in records. In the case of friend-links in Social Network Services (SNS) such as Twitter, the problem of sparseness again holds. Furthermore, we can find some *noisy* observations due to spam accounts that randomly follow unknown users for advertisement or other purposes. When applying the conventional block models to such data, as shown later, we often obtain unexpected clusters with *weak* relations among pairwise relational data within the cluster.

In this paper, we address the problems of noisiness and sparseness of real-world relational data. The number of these *irrelevant* data entries (elements of the observed relational data matrix) that do not strongly relate to the other data entries is huge in the dataset. As a result, they may obscure the interesting part of the observations. Assume each user is a data point located in a sparse and high-dimensional feature space: its feature vector is an observed relation between many other users. In such a space, every data point is similarly distant from all other data points because most of the feature vectors are sparse or uniformly distributed. This means we cannot distinguish the users because of high-dimensional *irrelevant* features. Also, since SBM and IRM are optimized to maximize the posterior probability of all clustering and parameters based on all the observations, the irrelevant data entries make it difficult to extract the core structure of small *relevant* and *interesting* clusters.

Specifically, we extend IRM so that only relevant rows and columns of a given relational data matrix can automatically be extracted for clustering. Intuitively, in the case of user-item purchase records, not all users and items but some of them are supposed to be analyzed for clustering. The proposed model estimates the *relevancy* of each row and column depending on the observations. If a row (column) is estimated as *irrelevant* to form a cluster, then the row (column) is excluded from the IRM process. We only perform

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX. Copyright 2012 by the authors.

the IRM on those who are classified as *relevant*; therefore, the IRM analysis on the core relations are not disturbed by noise observations. This relevancy and IRM-based clustering are seamlessly connected in the probabilistic generative model, and we can obtain its inference algorithm as a straightforward Gibbs sampler. Our proposed model is inspired by the subset clustering model [Hoff, 2005, Guan et al., 2011] that explicitly excludes irrelevant elements from feature vectors for better clustering of these vectors.

The rest of this paper is organized as follows. In the next section, we introduce the IRM as the baseline of our method, and reveal its drawbacks. We present a new model as a solution for the problem in the third section, and explain the inference procedure in the fourth section. The fifth section is devoted to experimental evaluations, and the final section concludes the paper.

2 Infinite Relational Models

We first explain the infinite relational model (IRM) [Kemp et al., 2006], which estimates an unknown number of hidden clusters from relational data. In IRM, the Dirichlet process (DP) is used as a prior for clusters of an unknown number, and is denoted as $\mathrm{DP}(\alpha, G_0)$ where $\alpha > 0$ is a parameter and G_0 is a base measure. We write $G \sim \mathrm{DP}(\alpha, G_0)$ when a distribution $G(\theta)$ is sampled from DP. We can implement DP by using either a stick-breaking process [Sethuraman, 1994] or a Chinese restaurant process (CRP) [Blackwell and MacQueen, 1973] as a marginalized form of stick-breaking process. In this paper, we employ the CRP representation of DP. CRP itself is a probability of partitioning of N objects. Let $z_i = k, i \in \{1, \ldots, N\}, k \in \{1, \ldots, K\}$ denote that the ith object is assigned to the kth partition (cluster) among the total K partitions. Then the CRP is represented as the following equations:

$$\mathrm{CRP}(z_{1:N}|\alpha) = \alpha^K \frac{\prod_{k=1}^{K} (n_k - 1)!}{\prod_{i=1}^{N} (\alpha + i - 1)}, \tag{1}$$

$$p(z_i = k|z_{\setminus i}, \alpha) = \begin{cases} \frac{n_{k\setminus i}}{n-1+\alpha} & n_{k\setminus i} > 0, \\ \frac{\alpha}{N-1+\alpha} & n_{k\setminus i} = 0. \end{cases} \tag{2}$$

Equation 1 shows the joint probability of K partitions. Equation 2 represents the probability of the object i being allocated to the partition k given $K - 1$ partitions. n_k denotes the number of objects assigned to the partition k, and $n_{k\setminus i}$ denotes the same number excluding object i.

The IRM is an application of the DP for relational data. Let us first assume a binary two-place relation on the set of objects $D = \{1, \ldots, i, \ldots, N\}$ as $D \times D \rightarrow \{0, 1\}$. For simplicity, we assume two-place relations throughout the paper, but the extension for more high-dimensional data is straightforward. The IRM divides the set of N objects into multiple clusters based on the matrix of observed relational data $\boldsymbol{X} = \{x_{i,j} \in \{0, 1\};\ 1 \leq i, j \leq N\}$. The IRM is able to infer the number of clusters at the same time because it uses DP as a prior distribution of the cluster partition. A data entry $x_{i,j} \in \{0, 1\}$ denotes the existence of a relation between a row object i and a column object j $(i, j \in \{1, 2, \ldots, N\})$. In the case of SNS friend-links, if there is (not) a friend-link from user i to user j, then $x_{i,j} = 1$ (0). We allow asymmetric relations $x_{i,j} \neq x_{j,i}$ throughout the paper. The probabilistic generative model of the IRM is as follows:

$$\theta_{k,l}|c_{k,l}, d_{k,l} \sim \mathrm{Beta}\left(c_{k,l}, d_{k,l}\right), \tag{3}$$

$$z_i|\alpha \sim \mathrm{CRP}\left(\alpha\right), \tag{4}$$

$$x_{i,j}|\boldsymbol{Z}, \{\theta\} \sim \mathrm{Bernoulli}\left(\theta_{z_i,z_j}\right). \tag{5}$$

In Eq. (3), $\theta_{k,l}$ is the strength of a relation between the objects in clusters k and l. We sample a cluster index of the object i, $z_i = k, k \in \{1, 2, \ldots, \}$ using the CRP as in Eq. (4). Generating the observed relational data $x_{i,j}$ follows Eq. (5) conditioned by the cluster assignments $\boldsymbol{Z} = \{z_i\}_{i=1}^N$ and the strengths θ. We call this model a "one-domain" model since object indices i and j point to the same domain (in the case of SNS, both i and j denote a user). A graphical model of one-domain IRM is illustrated in Fig. 2(A).

Let us assume the case where relation is defined between objects in different domains, namely $D_1 \times D_2 \rightarrow \{0, 1\}$ where $D_1 = \{1, \ldots, i, \ldots, N_1\}$ and $D_2 = \{1, \ldots, j, \ldots, N_2\}$. For such data, we define a "two-domain" IRM as follows:

$$\theta_{k,l}|c_{k,l}, d_{k,l} \sim \mathrm{Beta}\left(c_{k,l}, d_{k,l}\right), \tag{6}$$

$$z_{1,i}|\alpha_1 \sim \mathrm{CRP}\left(\alpha_1\right), \tag{7}$$

$$z_{2,j}|\alpha_2 \sim \mathrm{CRP}\left(\alpha_2\right), \tag{8}$$

$$x_{i,j}|\boldsymbol{Z}_1, \boldsymbol{Z}_2, \{\theta\} \sim \mathrm{Bernoulli}\left(\theta_{z_{1,i},z_{2,j}}\right). \tag{9}$$

In the above equations, i indexes the object of the first domain D_1, and j indexes the object of the second domain D_2. In the case of an online purchase record, the first domain corresponds to a user list, and an object i denotes a specific user i. The second domain corresponds to a list of shop items, and an object j denotes a specific item j. The data entry $x_{i,j}$ represents a relation between the user i and the item j: namely, the purchase record. In Eq. (6), $\theta_{k,l}$ is the strength of a relation between the cluster k of the first domain and the cluster l in the second domain. $z_{1,i}$ in Eq. (7) and $z_{2,j}$ in Eq. (8) denotes the cluster assignments in the first domain and the second domain, respectively. The main difference between the one-domain IRM and two-domain IRM is that the object cluster assignments of two domains are generated from independent CRP Eq. (7) and Eq. (8). It means that each domain may have a different number of clusters.

One drawback of IRM for highly noisy and sparse relational data, which is often the case, is the CRP prior itself. As we can see from Eq. (1) and Eq. (2), the CRP prior naively "counts" the cluster assignments. Therefore, the counts of irrelevant objects of $\boldsymbol{X}$ affect the posterior of $\boldsymbol{Z}$.

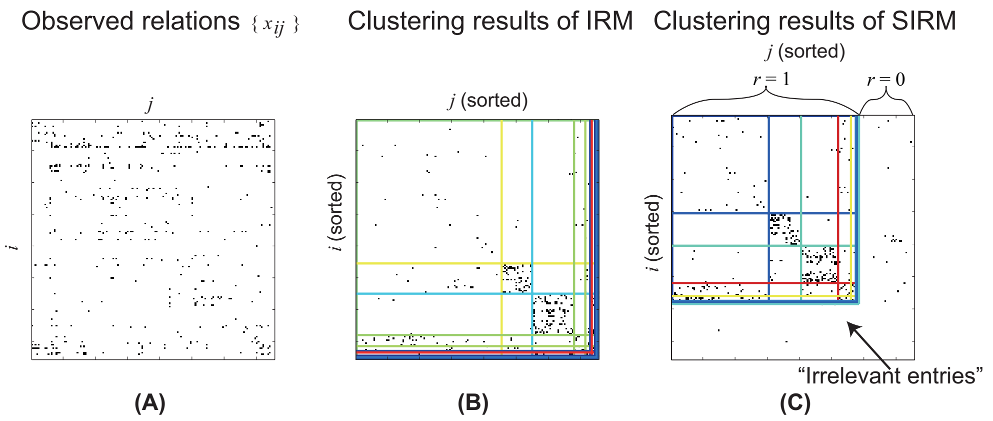

Figure 1: Illustrative example of IRM and SIRM (best viewed in color). (A): Observed relationships of Enron (Aug.) dataset (see the Experiments section for details). One black spot indicates existence of a link from a row object i to a column object j. (B): Clustering result of dataset by IRM. Color lines indicate the boundaries of object clusters, and each rectangle block represents a cluster-cluster relationship that abstracts individual data entries within the block. The object indices are sorted. (C): Clustering result of the same dataset by proposed SIRM. The lower right region that is NOT surrounded by color lines represents "irrelevant" data entries. SIRM automatically excludes these irrelevant data entries, and finds a more detailed cluster structure within "relevant" objects in the upper left region.

Another problem is minute clusters. IRM assumes that every object must be assigned to one of K clusters. According to the stick-breaking construction of DP, the mixing ratio of K clusters tends to follow the power law. Thus a typical clustering result of IRM looks like Fig. 1(B): many minute clusters in the lower right area. This is practically problematic because it makes it difficult to discern whether each minute cluster truly extracts a property of given relational data or just fits to noise.

3 Subset Infinite Relational Models

We introduce an extension of IRM called the Subset Infinite Relational Model (SIRM) to deal with the aforementioned problems of irrelevant data entries. The term "subset" indicates that we only focus on hidden structure among the relevant objects that are the subset of whole objects.

This SIRM is inspired by the subset clustering models in [Hoff, 2005]. Similar models also have been proposed in [Hoff, 2006, Guan et al., 2011]. Our assumption is that relevant data entries follow some cluster-dependent properties, and irrelevant ones have no such dependencies. According to this assumption, a column (or row) data entry of an irrelevant object will distribute in the same manner regardless to the object in its counterpart. Such objects are not useful for data clustering nor do they characterize each cluster. To address this problem, we try to remove objects that do not contribute to the clustering under consideration. Thus, we define them as "irrelevant" objects. To distinguish relevant and irrelevant objects, we introduce a new hidden variable $r_i \in \{0, 1\}$. If object i is (not) relevant, then $r_i = 1$ (0).

In the case of one-domain IRM ($D \times D \to \{0, 1\}$), our SIRM is described as follows:

$$\phi | a, b \sim \mathrm{Beta}(a, b), \tag{10}$$

$$\theta_{k,l} | c_{k,l}, d_{k,l} \sim \mathrm{Beta}(c_{k,l}, d_{k,l}), \tag{11}$$

$$\lambda_i | e, f \sim \mathrm{Beta}(e, f), \tag{12}$$

$$r_i | \lambda_i \sim \mathrm{Bernoulli}(\lambda_i), \tag{13}$$

$$z_i | r_i = 1, \alpha \sim \mathrm{CRP}(\alpha), \tag{14}$$

$$z_i | r_i = 0 \sim \mathbb{I}(z_i = 0), \tag{15}$$

$$x_{i,j} | \boldsymbol{Z}, \boldsymbol{R}, \{\theta\}, \phi \sim \mathrm{Bernoulli}\left(\theta_{z_i,z_j}^{r_i r_j} \phi^{1 - r_i r_j}\right). \tag{16}$$

Equation 10 defines the distribution of a relation strength for irrelevant data entries, and Eq. (11) defines a relation strength for relevant data entries. Based on that assumption, the relevant parameters are independently sampled for each (k, l) cluster pair. λ_i in Eq. (12) denotes the probability of a relevancy flag variable r_i being 1. As explained, $r_i = \{0, 1\}$ in Eq. (13) indicates whether the object i is relevant or not.

The relevancy variables $\boldsymbol{R} = \{r_i\}_{i=1,\dots,N}$ affect the remaining generative process. If $r_i = 1$, then z_i is chosen based on the CRP as in Eq. (14). Otherwise ($r_i = 0$), its cluster assignment is set to $z_i = 0$ with a probability 1 as in Eq. (15). $\mathbb{I}(\cdot)$ denotes that the predicate always holds with a probability 1. The cluster 0 is an "irrelevant" cluster that *is not* related to the CRP in Eq. (14). Finally, the observed relation $x_{i,j}$ is conditioned by $\boldsymbol{Z}$ and $\boldsymbol{R}$. Equation 16 is slightly tricky: if both items i and j are assumed as relevant objects, i.e. $r_i = r_j = 1$, then "relevant" relation strength θ is used as a parameter of a Bernoulli trial. Otherwise, "irrelevant" relation strength ϕ is employed. These two equations indicate that the irrelevant object does not affect the clustering by the CRP in terms of the CRP prior (Eq. (14)) nor the likelihood (Eq. (16)). Thus, the SIRM focuses on relevant objects and is able to effectively reconstruct the hidden relation structure among these objects. A graphical model of one-domain SIRM is illustrated in Fig. 2(B).

In the case of cross-domain relational data ($D_1 \times D_2 \rightarrow \{0, 1\}$), we need to augment the "two-domain" IRM. Its extension is easy: we just double the variables of "one-domain" SIRM. The generative model for the two-domain SIRM is described as follows:

$$\phi | a, b \sim \mathrm{Beta}\left(a, b\right), \tag{17}$$

$$\theta_{k,l} | c_{k,l}, d_{k,l} \sim \mathrm{Beta}\left(c_{k,l}, d_{k,l}\right), \tag{18}$$

$$\lambda_{1,i} | e_1, f_1 \sim \mathrm{Beta}\left(e_1, f_1\right), \tag{19}$$

$$\lambda_{2,j} | e_2, f_2 \sim \mathrm{Beta}\left(e_2, f_2\right), \tag{20}$$

$$r_{1,i} | \lambda_{1,i} \sim \mathrm{Bernoulli}\left(\lambda_{1,i}\right), \tag{21}$$

$$r_{2,j} | \lambda_{2,j} \sim \mathrm{Bernoulli}\left(\lambda_{2,j}\right), \tag{22}$$

$$z_{1,i} | r_{1,i} = 1, \alpha_1 \sim \mathrm{CRP}\left(\alpha_1\right), \tag{23}$$

$$z_{1,i} | r_{1,i} = 0 \sim \mathbb{I}\left(z_{1,i} = 0\right), \tag{24}$$

$$z_{2,j} | r_{2,j} = 1, \alpha_2 \sim \mathrm{CRP}\left(\alpha_2\right), \tag{25}$$

$$z_{2,j} | r_{2,j} = 0 \sim \mathbb{I}\left(z_{2,j} = 0\right), \tag{26}$$

$$x_{i,j} | \boldsymbol{Z}_1, \boldsymbol{Z}_2, \boldsymbol{R}_1, \boldsymbol{R}_2, \{\theta\}, \phi \sim \mathrm{Bernoulli}\left(\theta_{z_{1,i}, z_{2,j}}^{r_{1,i} r_{2,j}} \phi^{1 - r_{1,i} r_{2,j}}\right). \tag{27}$$

Equation 19, Eq. (21), Eq. (23), and Eq. (24) define parameters and hidden variables for the first domain D_1. Equation 20, Eq. (22), Eq. (25), and Eq. (26) define parameters and hidden variables for the second domain D_2.

We explain the virtues of SIRM for highly noisy and sparse relational data using Fig. 1, comparing panels (B) and (C). Panel (C) presents a typical clustering result of SIRM. SIRM automatically excludes irrelevant data entries to the lower right region thanks to relevance variables r_i. Because of Eq. (14) and Eq. (15), irrelevant objects have no impact on the clustering of the detailed structure of relevant data entries in the upper left region.

SIRM also solves the problem of minute clusters. One reason for generating so many minute clusters in IRM is that IRM partitions every object to one of K clusters. Therefore, a noisy random object tends to fit as an independent cluster. However, in SIRM only relevant objects are partitioned by CRP. Thus, most of the clusters in the color-lines-surrounded upper left region in Fig. 1 (C) have some meaning, including minute clusters. On the contrary, noisy minute clusters will be merged in the irrelevant region.

3.1 Related Works

The proposed SIRM is most related to the feature (variable) selection models like [Hoff, 2005], but is also connected to sparse latent variable models and bi-clustering techniques. Carvalho et al. applied a latent factor model for regression tasks of cancer characteristics using high-dimensional gene expression feature vectors [Carvalho et al., 2008]. In their work, sparsity priors were placed on loading matrices of the regression to pick up a limited number of effective elements. They also studied the use of Dirichlet Process to discover the unknown number of latent factors. Miller et al. [Miller et al., 2009] incorporated binary variable selection variables in the context of relational data analysis. In their model, each object is characterized by binary variables that select appropriate latent features to describe the object. Their model extends the number of latent features to infinite (so as the binary variables), and is formulated using the Indian Buffet Process [Griffiths and Ghahramani, 2011]. One major difference between the model of [Miller et al., 2009] and our proposed model is the usage of binary variables. The former employs binary variables to select *latent* features, while the latter, our model, employs binary variables to select relevant and irrelevant *observed* data entries: i.e. objects.

In the context of bi-clustering of relational data, Sutskever et al. proposed a bi-clustering model called Bayesian Clustered Tensor Factorization (BCTF) [Sutskever et al., 2010] . BCTF is a richer bi-clustering model that the relations itself can be decomposed into several clusters while our model (and the original IRM) considers only one type of relations. Instead, our model introduces the binary variables to cope with the sparsity problem mentioned above while BCTF does not directly solve this problem. In [Bordes et al., 2011], the authors proposed a novel technique to embed objects into a low-dimensional vector space using object-object multi-type relations in knowledge bases such as WordNet. Since the goals of [Bordes et al., 2011] and our work is different, soft clustering of objects like vector embedding could be another possible application of our "subset" model.

4 Inference

Since SIRM is a rather simple probabilistic model, a variety of inference procedures are applicable for solving SIRM. In this paper, we briefly explain the inference algorithm of one-domain SIRM by Gibbs sampling. We can marginalize out all parameters ϕ, θ, λ thanks to the conjugacy. Also we fit hyperparameters α, a, b, c, d, e, f by posterior sampling assuming Gamma priors.

For detailed derivations and the inference algorithms for the two-domain SIRM, please consult the supplemental material.

4.1 Sampling Hidden Variables

We found that simultaneous sampling of r_i and z_i leads to a simpler inference for SIRM than deriving a solution for

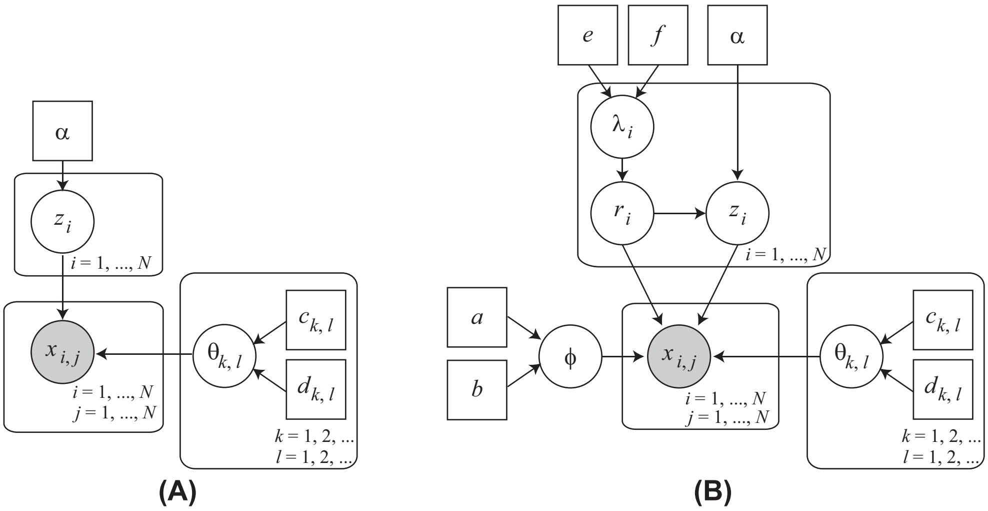

Figure 2: Graphical models. Circle node denotes probabilistic variables, rectangle nodes denote constants, and shaded nodes represent observations. (A): Original one-domain IRM. (B): Proposed one-domain SIRM.

each variable independently.

Let us denote the current number of realized clusters by K. Regarding the sampling of the ith object, we divide the observations $\boldsymbol{X}$ into two parts: data entries that relate to the object i $\boldsymbol{X}^{+i} = \{x_{i,\cdot}, x_{\cdot,i}\}$, and those that do not $\boldsymbol{X}^{\setminus i} = \{\boldsymbol{X} \setminus \boldsymbol{X}^{+i}\}$. Also we define the following quantities:

$$n_{k,l} = \sum_i \sum_j r_i z_{i,k} r_j z_{j,l} x_{i,j}, \tag{28}$$

$$\bar{n}_{k,l} = \sum_i \sum_j r_i z_{i,k} r_j z_{j,l} \left(1 - x_{i,j}\right), \tag{29}$$

$$m_k = \sum_i r_i z_{i,k}, \tag{30}$$

$$q = \sum_i \sum_j \left(1 - r_i r_j\right) x_{i,j}, \tag{31}$$

$$\bar{q} = \sum_i \sum_j \left(1 - r_i r_j\right)\left(1 - x_{i,j}\right). \tag{32}$$

The superscript of $\setminus i$ denotes the above statistics computed on $\boldsymbol{X}^{\setminus i}$. Also, the superscript $+i0$, $+ik$ denotes that the same statistics are computed on $\boldsymbol{X}^{+i}$ assuming $r_i = 0$ or $\{r_i = 1, z_i = k\}$, respectively.

We formulate the Gibbs posterior of $\{r_i, z_i\}$ as follows:

$$\begin{aligned} p\left(z_i = k, r_i | \boldsymbol{X}, \boldsymbol{Z}^{\setminus i}, \boldsymbol{R}^{\setminus i}\right) &\propto p\left(z_i = k, r_i | \boldsymbol{Z}^{\setminus i}, \boldsymbol{R}^{\setminus i}\right) \\ &\times p\left(\boldsymbol{X}^{+i} | z_i = k, r_i, \boldsymbol{X}^{\setminus i}, \boldsymbol{Z}^{\setminus i}, \boldsymbol{R}^{\setminus i}\right) \end{aligned} \tag{33}$$

The first term of the right hand of Eq. (33) is straightforward. Multiplying Eq. (13) and Eq. (14) and marginalizing λ_i out using the conjugacy, we obtain the following:

$$\begin{aligned} & p\left(z_i = k, r_i | \boldsymbol{Z}^{\setminus i}, \boldsymbol{R}^{\setminus i}\right) \\ & \propto \begin{cases} f + \sum_{i' \neq i} (1 - r_{i'}) & r_i = 0, z_i = 0, \\ (e + \sum_{i' \neq i} r_{i'}) \frac{m_k^{\setminus i}}{\alpha + \sum_k m_k^{\setminus i}} & r_i = 1, z_i = k \in \{1, 2, \ldots, K\}, \\ (e + \sum_{i' \neq i} r_{i'}) \frac{\alpha}{\alpha + \sum_k m_k^{\setminus i}} & r_i = 1, z_i = K + 1. \end{cases} \end{aligned} \tag{34}$$

The second term of the right hand of Eq. (33) requires some computations. This is a likelihood term; thus we separately present the final results for $r_i = 0$ and $r_i = 1$.

$$\begin{aligned} & p\left(\boldsymbol{X}^{+i} | z_i = 0, r_i = 0, \boldsymbol{X}^{\setminus i}, \boldsymbol{Z}^{\setminus i}, \boldsymbol{R}^{\setminus i}\right) \\ & \qquad = \frac{B\left(a + q^{\setminus i} + q^{+i0}, b + \bar{q}^{\setminus i} + \bar{q}^{+i0}\right)}{B\left(a + q^{\setminus i}, b + \bar{q}^{\setminus i}\right)} \end{aligned} \tag{35}$$

$$\begin{aligned} & p\left(\boldsymbol{X}^{+i} | z_i = k, r_i = 1, \boldsymbol{X}^{\setminus i}, \boldsymbol{Z}^{\setminus i}, \boldsymbol{R}^{\setminus i}\right) \\ & = \frac{B\left(a + q^{\setminus i} + q^{+i1k}, b + \bar{q}^{\setminus i} + \bar{q}^{+i1k}\right)}{B\left(a + q^{\setminus i}, b + \bar{q}^{\setminus i}\right)} \\ & \times \frac{B\left(c_{k,k} + n_{k,k}^{\setminus i} + n_{k,k}^{+i1k}, d_{k,k} + \bar{n}_{k,k}^{\setminus i} + \bar{n}_{k,k}^{+i1k}\right)}{B\left(c_{k,k} + n_{k,k}^{\setminus i}, d_{k,k} + \bar{n}_{k,k}^{\setminus i}\right)} \\ & \times \prod_{l \neq k} \frac{B\left(c_{k,l} + n_{k,l}^{\setminus i} + n_{k,l}^{+i1k}, d_{k,l} + \bar{n}_{k,l}^{\setminus i} + \bar{n}_{k,l}^{+i1k}\right)}{B\left(c_{k,l} + n_{k,l}^{\setminus i}, d_{k,l} + \bar{n}_{k,l}^{\setminus i}\right)} \\ & \times \prod_{l \neq k} \frac{B\left(c_{l,k} + n_{l,k}^{\setminus i} + n_{l,k}^{+i1k}, d_{l,k} + \bar{n}_{l,k}^{\setminus i} + \bar{n}_{l,k}^{+i1k}\right)}{B\left(c_{l,k} + n_{l,k}^{\setminus i}, d_{l,k} + \bar{n}_{l,k}^{\setminus i}\right)} \end{aligned} \tag{36}$$

4.2 Posteriors of parameters

Though all parameters are marginalized out during the inference, the posteriors of parameters may be useful in many cases. Here we present these posteriors for that purpose.

$$p(\phi|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{R}) = \text{Beta}(a + q, b + \bar{q}) \tag{37}$$

$$p(\theta_{k,l}|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{R}) = \text{Beta}(c_{k,l} + n_{k,l}, d_{k,l} + \bar{n}_{k,l}) \tag{38}$$

$$p(\lambda_i|\boldsymbol{R}) = \text{Beta}(e + r_i, f + (1 - r_i)) \tag{39}$$

5 Experiments

5.1 Data

We evaluated the performance of our methods with original IRMs using synthetic data and real data.

For synthetic data, we assume a sparse data scenario: i.e. the irrelevant objects have very few relation observations, and relevant objects have some dense relationships among them. We have prepared two datasets for one-domain models and two-domain models, respectively. For one-domain models, $N = 500$ or $N = 1,000$ while the number of relevant objects is fixed at 50. The ground truth number of hidden clusters among relevant objects is fixed at $K = 3$ for each data. For two-domain models, $\{N_1, N_2\} = \{400, 500\}$ or $\{800, 900\}$. The number of relevant objects is again fixed at 40 for domain 1, and 50 for domain 2. The ground truth number of hidden clusters among relevant objects is fixed at $K_1 = 4$ and $K_2 = 5$ for each data.

We choose the Enron e-mail dataset [Klimat and Yang, 2004] as an example of a real-world one-domain dataset, and it is used in many studies [Tang et al., 2008, Fu et al., 2009, Ishiguro et al., 2010]. We extracted monthly transactions of e-mails sent in 2001. The dataset contained $N = 151$ company members of Enron. $x_{i,j} = 1(0)$ if there is (not) an e-mail sent from a member i to a member j. Out of twelve months, we selected the transactions of August and October because these two periods were milestones of the Enron scandal: the CEO of Enron resigned in August, and the accounting scandal was first reported in October.

As real-world cross domain relational data, we collected log data of an online cartoon distribution service for mobile phones in Japan. With this service, users pay monthly to read cartoons on their phones. Thus, some users purchased an item more than once to read it over the course of a month. Some cartoons have several volumes, and we regarded a cartoon that had several volumes as one item. The first domain index i corresponds to a user, and the second domain index j corresponds to a cartoon item. We selected $N_1 = 1000$ users' records randomly. The number of cartoon items is $N_2 = 316$. We prepared two datasets (Cartoon 1, Cartoon 2), each of which randomly and independently subsamples users.

Table 1: Test data log likelihood per test data entry on one-domain data. Averages of 20 runs are presented, parenthesized numbers indicate standard deviations. Larger values are better.

Dataset	IRM	SIRM
synth (small)	-0.140 (0.007)	**-0.098** (0.001)
synth (large)	-0.143 (0.082)	**-0.097** (0.000)
Enron (Aug.)	-0.128 (0.012)	**-0.112** (0.006)
Enron (Oct.)	-0.173 (0.006)	**-0.150** (0.004)

Table 2: Test data log likelihood per test data entry on cross-domain data. Averages of 20 runs are presented, parenthesized numbers indicate standard deviations. Larger values are better.

Dataset	IRM	SIRM
synth (small)	-0.094 (0.007)	**-0.060** (0.000)
synth (large)	-0.103 (0.006)	**-0.062** (0.000)
Cartoon 1	-0.130 (0.001)	**-0.122** (0.012)
Cartoon 2	-0.133 (0.013)	**-0.127** (0.007)

Please note that the sizes of these datasets are larger compared to the original IRM paper [Kemp et al., 2006]. This is because our goal is to build a model that is capable of effectively analyzing larger and noisy relational data.

5.2 Quantitative Results

For quantitative comparisons, we compute test data log likelihood per test data entry. For each experimental run, we randomly pick test data entries from the matrix $\boldsymbol{X}$ (approximately 1% of the entire $\boldsymbol{X}$ entries), and hide them during the training period. After the Gibbs inference of the model is completed, we compute the log likelihoods of these test data entries. To align the different number of test data entries, we divide the log likelihoods by the number of test data entries at that run. We tested several initial hyperparameter settings, and report the best result for each model and the dataset.

Table 1 summarizes the test data log likelihoods on one-domain models. Also, we show the test data log likelihoods on two-domain models in Table 2. As evident from the table, the proposed SIRM is better than the original IRM for all datasets.

5.3 Qualitative Results on One-domain Data

Next, we qualitatively examine the clustering results of one-domain real data from Enron. The raw data and clustering results of the August data (Enron (Aug.)) are presented in Fig. 1. As evident from the figure, SIRM nicely excludes irrelevant objects (company members) and finds some detailed clusters within relevant objects.

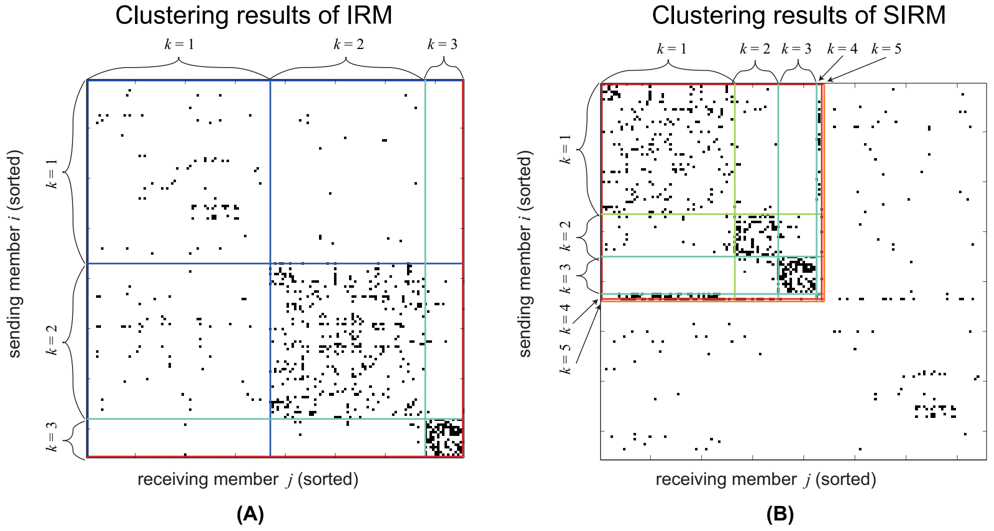

Figure 3: Typical clustering results of Enron (Oct.) dataset (best viewed in color). (A) Clustering result by IRM. (B) Clustering result by SIRM.

In this section, we closely look at the results on the October data. The clustering results of IRA and SIRM on the October data (Enron (Oct.)) are presented in Fig. 3(A) and Fig. 3(B), respectively. Comparing these two figures, the most remarkable difference is the number of irrelevant objects in SIRM. Out of $N = 151$ objects, 64 objects are assumed irrelevant ($r_i = 0$). Most of these irrelevant objects are merged in the first cluster in IRM, resulting in a very sparse and non-informative cluster. On the other hand, clusters in SIRM are much tighter and show strong (dense) connections among the clusters.

The second cluster ($k = 2$) in SIRM is a set of VIP members. The objects assigned to this cluster include presidents of Enron companies such as Enron online and Enron Global Markets, a governmental relation executive, a manager of chief risk management officers, an in-house lawyer, a head manager of risk management, several vice presidents and the founder of Enron. We thought that these VIP members must have had to keep in touch during this month to deal with the accounting scandal news. We note that those members are also clustered in the second cluster of the IRM result (Fig. 3(A)). However, in the IRM result, it is difficult to derive this VIP interpretation because other members are also collected in the same cluster. This comparison indicates that relevant variables r_i effectively exclude irrelevant objects, and "concentrate" the meaningful memberships.

One interesting point is that the third ($k = 3$) cluster by IRM is almost the same with the third cluster by SIRM. This cluster consists of employees and managers related to pipeline and regulatory business. Clustered company members include a manager and a vice president of regulatory affairs, a director of pipeline business, and a president of Enron Gas Pipeline.

As discussed, minute clusters in SIRM are meaningful since irrelevant objects are excluded already. In Fig. 3(B), the fourth and fifth clusters are the ones. Both clusters have only one object as a cluster member. However these objects show very active relations among other clusters. because these two objects correspond to a manager and the COO. As expected, the IRM could not find these clusters.

5.4 Qualitative Results on Two-domain Data

Next, we qualitatively examine the result of two-domain real data from Cartoon purchase data. Clustering results are almost the same between two datasets. The clustering results of IRM and SIRM on the Cartoon data are presented in Fig. 4(A) and Fig. 4(B), respectively.

Figure 4(A) shows a typical example of using IRM on real-world dataset. Because of noisy inputs and many objects, the first domain D_1 (users, the vertical axis) and the second domain D_2 (cartoon items, the horizontal axis) have many minute clusters that prevent an effective interpretation of the clustering results. On the other hand, SIRM effectively excludes irrelevant cartoon items, and results in a relatively moderate numbers of clusters for both domains (Fig. 4(A)).

Since we have no information about the users (the first domain), we focus on the clusters of the second domains in the SIRM result (Fig. 4(B)). The second cluster ($l = 2$) of the second domain in SIRM is a cluster of major cartoons for youngsters. Genres of cartoons in this cluster is mixed: romantic comedies for girls and ladies (e,g, "Sensual Phrase"), action and heroic stories (e.g, "Devilman", "BASARA"), situation comedies (e.g. "Crayon Shin-chan") and others. However, cartoon items in this cluster have several common points to be major such as:

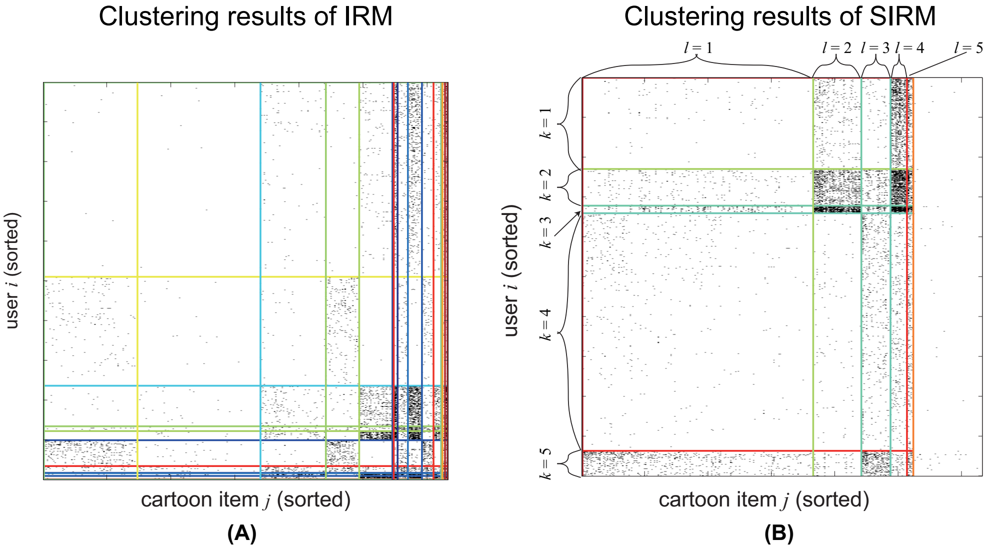

Figure 4: Typical clustering results of Cartoon dataset (best viewed in color). (A) Clustering result by IRM. (B) Clustering result by SIRM.

1. Published in the major cartoon magazines, mainly targeted for teenage boys and girls.
2. Animated as TV programs or OVAs. Some cartoons are even filmized.
3. Relatively recent cartoons are assembled. Older cartoons are famous among recent young readers because they are reprinted over years.

The third cluster ($l = 3$) of the second domain in SIRM are a set of cartoon items mainly targeted at older (20-40 year) male audiences. This cluster includes hard cyberpunks (e.g. “Appleseed”), stories about outlaws and gangsters (e.g. “Ore-no-Sora”, “Lupin III”), and stories about the historical heroes (of Japan and China).

The fourth cluster ($l = 4$) of the second domain in SIRM would be collected through a different perspective. This cluster consists of a mixture of different kinds of cartoons such that some oldies (e.g. “Cyborg 009”, “Dokonjo Gaeru”) and recent romance stories that were extraordinary popular among the middle-aged (“Winter Sonata”). Our guess is that this cluster is a set of cartoons popular for male audiences of specific ages (around 30 to 50). The oldies were popular when the audiences were very young, while the recent stories started the boom when the audiences got matured. Assuming from this observation, we can infer that the fifth cluster ($k = 5$) of the first domain would be the users with middle-aged males.

These qualitative results indicate that SIRM is useful for noisy and sparse real-world cross-domain relational data in the sense that it can extract interpretable and interesting relations.

6 Conclusion

In this paper, we have addressed the problem of noisiness and sparseness of the relational data. We have introduced a notion of the relevancy of objects in relational data, and proposed an extension of the IRM incorporating the relevance variables. The proposed model partitions relevant rows and columns of a given relational data matrix, while automatically excludes irrelevant entries. Thus the IRM analysis on the core relations is not annoyed by noise observations. Through experiments, we confirmed that the proposed SIRM is superior in test data log likelihoods. We also observed that the SIRM successfully extracted hidden clusters of core relations from real-world datasets.

As future work, one promising way is to derive a dynamic extension of SIRM similar to a dynamic version of IRM [Ishiguro et al., 2010]. Because the relationships among objects are inherently time-varying, the relevancy of an object in the relations is also time-varying. It is also important to verify the performance of SIRM against higher-order tensor relations such as three-place relations.

The proposed model is a bold simplification of the actual data generation process. For example, the proposed model does not care *why* an item is assumed relevant or irrelevant in purchase data. However, revealing such reasons (e.g. shops to check items, or budget constraints of users) is useful for further clustering. This simple example indicates that there are still many open problems concerning the relevance clustering models.

Acknowledgements

This work is partially supported by FIRST program of JSPS.

References

D. Blackwell and J. B. MacQueen. Ferguson distributions via Polya urn schemes. *The Annals of Statistics*, 1(2): 353–355, 1973.

A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. In *Proc. AAAI*, 2011.

C. M. Carvalho, J. Chang, J. E. Lucas, J. R. Nevins, Q. Wang, and M. West. High-dimensional sparse factor modeling: Applications in gene expression genomics. *J. Am. Stat. Assoc.*, 103(484):1438–1456, 2008.

A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.

E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. *PNAS*, 101(Suppl 1):5220–5227, 2004.

W. Fu, L. Song, and E. P. Xing. Dynamic mixed membership blockmodel for evolving networks. In *Proc. ICML*, 2009.

T. L. Griffiths and Z. Ghahramani. The Indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224, 2011.

Y. Guan, J. Dy, and M. Jordan. A unified probabilistic model for global and local unsupervised feature selection. In *Proc. ICML*, 2011.

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

P. D. Hoff. Model-based subspace clustering. *Bayesian Analysis*, 1(2):321–344, 2006.

K. Ishiguro, T. Iwata, N. Ueda, and J. Tenenbaum. Dynamic infinite relational model for time-varying relational data analysis. In *Proc. NIPS*, 2010.

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*, 2006.

B. Klimat and Y. Yang. The enron corpus: A new dataset for email classification research. In *Proc. ECML*, 2004.

D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proc. Int. Conf. on Information and Knowledge Management*, pages 556–559, 2003.

K. T. Miller, T. L. Griffiths, and M. I. Jordan. Nonparametric latent feature models for link prediction. In *Proc. NIPS*, 2009.

K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *J. Am. Stat. Assoc.*, 96(455):1077–1087, 2001.

J. Sethuraman. A constructive definition of dirichlet process. *Statistica Sinica*, 4:639–650, 1994.

I. Sutskever, R. Salakhutdinov, and J. B. Tenenbaum. Modelling relational data using Bayesian clustered tensor factorization. In *Proc. NIPS*, 2010.

L. Tang, H. Liu, J. Zhang, and Z. Nazeri. Community evolution in dynamic multi-mode networks. In *Proc. SIGKDD*, pages 677–685, 2008.

S. Zhu, K. Yu, and Y. Gong. Stochastic relational models for large-scale dyadic data using mcmc. In *Proc. NIPS*, 2009.