Stochastic Blockmodel with Cluster Overlap, Relevance Selection, and Similarity-Based Smoothing

Joyce Jiyoung Whang*, Piyush Rai†, and Inderjit S. Dhillon*
*Department of Computer Science
The University of Texas at Austin, Austin, TX, USA
{joyce,inderjit}@cs.utexas.edu
†Department of Electrical & Computer Engineering
Duke University, Durham, NC, USA
piyush.rai@duke.edu

***Abstract*—Stochastic blockmodels provide a rich, probabilistic framework for modeling relational data by expressing the objects being modeled in terms of a latent vector representation. This representation can be a latent indicator vector denoting the cluster membership (hard clustering), a vector of cluster membership probabilities (soft clustering), or more generally a real-valued vector (latent space representation). Recently, a new class of *overlapping* stochastic blockmodels has been proposed where the idea is to allow the objects to have hard memberships in multiple clusters (in form of a latent binary vector). This aspect captures the properties of many real-world networks in domains such as biology and social networks where objects can simultaneously have memberships in multiple clusters owing to the multiple *roles* they may have. In this paper, we improve upon this model in three key ways: (1) we extend the overlapping stochastic blockmodel to the bipartite graph case which enables us to simultaneously learn the overlapping clustering of two different sets of objects in the graph; the unipartite graph is just a special case of our model, (2) we allow objects (in either set) to not have membership in any cluster by using a *relevant object selection* mechanism, and (3) we make use of additionally available object *features* (or a *kernel matrix* of pairwise object similarities) to further improve the overlapping clustering performance. We do this by *explicitly* encouraging similar objects to have *similar cluster membership vectors*. Moreover, using nonparametric Bayesian prior distributions on the key model parameters, we side-step the model selection issues such as selecting the number of clusters *a priori*. Our model is quite general and can be applied for both overlapping clustering and link prediction tasks in unipartite and bipartite networks (directed/undirected), or for *overlapping* co-clustering of general binary-valued data. Experiments on synthetic and real-world datasets from biology and social networks demonstrate that our model outperforms several state-of-the-art methods.**

***Keywords*-stochastic blockmodel; overlapping clustering; link prediction; relevance selection; nonparametric Bayesian**

I. Introduction

Modeling relationships (e.g., pairwise interactions) among objects is becoming ever increasingly commonplace in various domains [10]. In biology, known gene-gene interactions may be specified as a graph in form of an adjacency matrix, or drug-protein associations may be specified as a bipartite graph. In social network analysis, a friendship network may be specified as an adjacency matrix. In recommender systems, a user-item ratings information may be given in form of dyads. In all these and other related problems dealing with relational data, an important goal is to learn latent low-dimensional representations of the objects and use these representations in downstream tasks such as clustering or prediction; for example, clustering users in a social network into communities [28], predicting values of unobserved links in a network [19], and so on.

The stochastic blockmodel (SB) is a generative model of a network, and relational data in general. This model and its variations have been applied to various problems in the statistical analysis of network data [10]. The SB models the link probability of a pair of objects as a stochastic function of their low-dimensional representations. Different types of latent representations lead to specific types of stochastic blockmodels. For example, the latent class model [23], [16] assumes that each object exclusively belongs to one cluster and the link probability between a pair of node depends on their cluster memberships (usually, a *pair* of objects belonging to clusters l and m will have a link generation probability w_{lm}). The latent class model was later generalized to the mixed-membership stochastic blockmodel [4]. In the mixed membership stochastic blockmodel (MMSB), each node has a distribution over clusters in form of a probability vector. The MMSB considers only soft-assignments of objects to multiple clusters. Both SB and MMSB have been applied to the problem of clustering objects given network data and also to the task of link prediction using the learned cluster memberships of the objects [10].

In many real-world datasets, however, it is natural to think of objects having hard memberships in multiple clusters. For example, users in a social network may belong to multiple communities; a protein may have multiple functions; or a gene may be regulated by multiple transcription factors. The assumption of each object being assigned to a single cluster [23], [16], [32], [21], or having *soft* memberships in multiple clusters [4], would be rather restrictive in such

scenarios. Motivated by this, there has been a considerable interest in learning models that allow objects to have *hard* memberships in multiple clusters. A recent survey about overlapping clustering methods in the context of network data can be found in [31].

On the other hand, in many real-world networks, some objects may be irrelevant and therefore would not be expected to have membership in any cluster. Moreover, in addition to the graph over objects, often we may have access to side information (e.g., in form of a similarity matrix between objects). It would be desirable to be able to use this side information to regularize the clustering assignments: if two objects are similar (as measured by the similarity matrix) then their cluster assignment vectors should also be similar. Finally, most existing methods for overlapping clustering only consider "unipartite" graphs and cannot be applied to bipartite graphs [7] where we want to simultaneously cluster two different sets of objects where clusters in each set could overlap.

Motivated by these desiderata, we propose a new stochastic blockmodel for overlapping clustering and link prediction. In particular, (1) our model has a noise robustness property where noise means presence of irrelevant objects in the network: it can *simultaneously* perform relevant object selection and overlapping clustering of relevant objects, (2) our model can make use of side information in form of a kernel/similarity matrix between the objects (either directly given or computed using *features* of the objects) to encourage similar objects to have similar cluster membership vectors, and (3) our model works for both unipartite (directed/undirected) and bipartite graphs; in the latter case, we simultaneously perform overlapping clustering for both sets of objects. Moreover, taking a nonparametric Bayesian approach [8], our model also circumvents the crucial issue of selecting the number of clusters. In addition, our model is not limited to modeling relational data; the bipartite version can be applied to general binary-valued data such as *overlapping* co-clustering of binary-valued data. Finally, although in this exposition we assume that the graph is binary, the model can easily be extend to weighted graphs [3] where edges can take non-binary values. We name our model ROCS to denote **R**elevance-based **O**verlapping **C**lustering with **S**imilarity-based-smoothing.

The rest of the paper is organized as follows: In Section II, we introduce some notation. In Section III, we briefly describe the Indian Buffet Process [12], a nonparametric Bayesian prior distribution, which we will use as one of the building blocks of our model. In Section IV, we first describe our basic model, and then describe how to extend it to handle relevant object selection, and to incorporate the similarity (kernel) matrix over the objects. We explain the inference method in Section V, and present experimental results on overlapping clustering and link prediction tasks in Section VI. Finally, we briefly review related work in Section VII, and state the conclusions and possible extensions of our model in Section VIII.

II. Notation

Bipartite case: In the bipartite graph case, we are given two sets of objects and a graph in form of an adjacency matrix such that an edge only exists between objects in different sets. We assume that the first set $\mathcal{A}$ has N objects, and the second set $\mathcal{B}$ has M objects. The "adjacency" matrix is denoted by $\boldsymbol{A} \in \{0,1\}^{N \times M}$. In addition, we are given two similarity matrices $\boldsymbol{S}^A \in [0,1]^{N \times N}$ and $\boldsymbol{S}^B \in [0,1]^{M \times M}$ defined over the set of objects in $\mathcal{A}$ and $\mathcal{B}$, respectively. We will denote the overlapping cluster assignments of objects in set $\mathcal{A}$ by $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N]^\top \in \{0,1\}^{N \times K}$ and in set $\mathcal{B}$ by $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M]^\top \in \{0,1\}^{M \times L}$. Here K and L refer to the number of clusters in set $\mathcal{A}$ and $\mathcal{B}$, respectively. An entry $u_{nk} = 1$ (resp. $v_{ml} = 1$) denotes that object n in set $\mathcal{A}$ (resp. m in set $\mathcal{B}$) belongs to cluster k (resp. cluster l). In our model, we do not have to assume that K and L are specified *a priori* but these will be inferred from data via nonparametric Bayesian modeling [8]. Further, we use $\boldsymbol{R}^A \in \{0,1\}^{N \times 1}$ and $\boldsymbol{R}^B \in \{0,1\}^{M \times 1}$ which we call a relevance vector for objects in set $\mathcal{A}$ and $\mathcal{B}$ respectively. An entry $R^A_n = 1$ (resp. $R^B_m = 1$) indicates that object n in set $\mathcal{A}$ (resp. object m in set $\mathcal{B}$) is relevant, and irrelevant otherwise. We will learn the relevance vectors as part of our model.

Unipartite case: Our notations in the unipartite graph case would be similar to the bipartite graph case except that we will drop the superscripts identifying the sets. In particular, we are given a set of N objects and a graph (directed/undirected) in form of their adjacency matrix $\boldsymbol{A} \in \{0,1\}^{N \times N}$. The similarity matrix will be denoted by $\boldsymbol{S} \in [0,1]^{N \times N}$, the overlapping cluster assignments by $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N]^\top \in \{0,1\}^{N \times K}$, and the relevance vector by $\boldsymbol{R} \in \{0,1\}^{N \times 1}$. As in the bipartite graph case, we will learn the number of clusters K and the relevance vector from the data.

III. Background

Our stochastic blockmodel is based on clustering the objects such that each object could potentially belong to multiple clusters. Given a set of N objects and K clusters, an overlapping clustering of these objects can be represented by a binary matrix $\boldsymbol{U} \in \{0,1\}^{N \times K}$. An entry $u_{nk} = 1$ means that object n belongs to cluster k. Each row in $\boldsymbol{U}$ can have multiple 1s in the overlapping clustering setting. A crucial issue is choosing the number of clusters which is rarely known *a priori*. This translates to choosing the number of columns in $\boldsymbol{U}$. The Indian Buffet Process [12] provides a nonparametric Bayesian prior distribution on sparse binary matrices, such as $\boldsymbol{U}$, such that the number of columns in $\boldsymbol{U}$ need not be specified beforehand but can instead be learned from data.

The Indian Buffet Process (IBP) can be most easily understood using a culinary metaphor. In this metaphor, customers correspond to the N rows of $\boldsymbol{U}$ and dishes correspond to the K columns of $\boldsymbol{U}$. Customers enter one-by-one into an Indian Buffet that serves a potentially infinite number of dishes. The first customer selects $Poisson(\alpha)$ dishes to begin with, where α is a hyperparameter. Each subsequent customer (say n-th customer) selects an already selected dish k with a probability proportional to how many previous customers have chosen this dish. This probability is given by m_k/n where m_k is the number of previous customers who chose dish k. The n-th customer thereafter selects $Poisson(\alpha/n)$ new dishes. This process results in a binary matrix $\boldsymbol{U}$ of customer-dish assignments where $u_{nk} = 1$ means that customer n selected dish k. Note that since a customer could potentially select multiple dishes, there can be multiple 1s in each row of $\boldsymbol{U}$.

Denoting the number of *new* dishes chosen by the n-th customer by ${K_1}^{(n)}$, the resulting probability distribution of $\boldsymbol{U}$ has the following form:

$$P(\boldsymbol{U}) = \frac{\alpha^{K_+}}{\prod_{n=1}^{N} {K_1}^{(n)}!} \exp\{-\alpha H_N\} \prod_{k=1}^{K_+} \frac{(N-m_k)!(m_k-1)!}{N!}$$

where K_+ denotes the number of nonzero columns in $\boldsymbol{U}$ and H_N denotes the N-th Harmonic number ($H_N = \sum_{j=1}^{N} \frac{1}{j}$). Since the number of overall dishes (which is also equal to the number of columns in $\boldsymbol{U}$) was not specified *a priori*, the IBP becomes a natural choice of a prior distribution for binary matrices like $\boldsymbol{U}$ with a fixed number of rows and an *a priori* unknown number of columns. In the stochastic blockmodel we present in Section IV, we will be using the IBP as a prior distribution on the object-cluster assignment matrix. This will allow us to simultaneously infer the number of clusters from the data and the cluster assignments for each object.

IV. The Model

We now describe our proposed stochastic blockmodel for doing overlapping clustering in bipartite and unipartite graphs. We first describe our basic model without the relevant object selection mechanism or exploiting the kernel based similarity information. We will discuss these extensions after describing the basic model.

Recall that, in the bipartite graph case, we are given an $N \times M$ binary adjacency matrix $\boldsymbol{A}$. Our model has the following form:

$$\begin{aligned} \boldsymbol{U} &\sim \mathcal{IBP}(\alpha_u) \\ \boldsymbol{V} &\sim \mathcal{IBP}(\alpha_v) \\ \boldsymbol{W} &\sim \mathcal{N}or(0, \sigma_w^2) \\ \boldsymbol{A} &\sim \mathcal{B}er(\sigma(\boldsymbol{U}\boldsymbol{W}\boldsymbol{V}^\top)) \end{aligned}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$, $\mathcal{B}er(p)$ is the Bernoulli distribution with parameter p, $\mathcal{IBP}(\alpha)$ is the IBP prior distribution with hyperparameter α, $\mathcal{N}or(0, \sigma^2)$ is the Gaussian distribution with hyperparameter σ, and $\boldsymbol{U} \in \{0,1\}^{N\times K}$ and $\boldsymbol{V} \in \{0,1\}^{M\times L}$ are the cluster assignment matrices for the objects in set $\mathcal{A}$ and $\mathcal{B}$, respectively. Here, $\boldsymbol{W}$ is a real-valued matrix of size $K \times L$ such that the entry w_{kl} controls the probability of a link between an object in set $\mathcal{A}$ and an object in set $\mathcal{B}$ when the former has a membership in cluster k and the latter has a membership in cluster l. In particular, note that we can write the probability of A_{nm} being 1 as follows:

$$P(A_{nm} = 1) = \sigma(\boldsymbol{u}_n \boldsymbol{W} \boldsymbol{v}_m^\top) = \sigma(\sum_{k,l} u_{nk} W_{kl} v_{ml})$$

Therefore, this model considers and combines all pairwise cluster interactions (suitably weighted by $\boldsymbol{W}$) to compute the link probabilities.

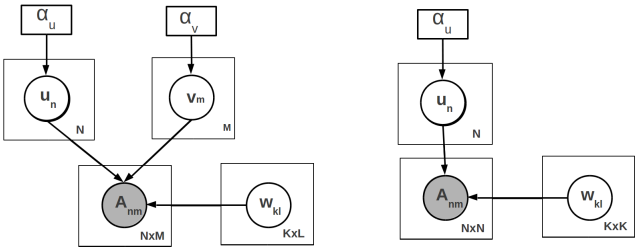

Figure 1. Our basic model. **Left:** The bipartite case. **Right:** The unipartite case. Hyperparameter σ_w is not shown for the sake of brevity. Note that K and L are actually unbounded due to nonparametric Bayesian modeling, and will be learned from data.

Figure 1 represents the graphical models for both bipartite and unipartite case. In the unipartite (directed/undirected) case where we only have a single set of objects and we want to model their pairwise interactions given by the adjacency matrix $\boldsymbol{A} \in \{0,1\}^{N\times N}$ (can be symmetric/asymmetric), our basic model reduces to the following form:

$$\begin{aligned} \boldsymbol{U} &\sim \mathcal{IBP}(\alpha_u) \\ \boldsymbol{W} &\sim \mathcal{N}or(0, \sigma_w^2) \\ \boldsymbol{A} &\sim \mathcal{B}er(\sigma(\boldsymbol{U}\boldsymbol{W}\boldsymbol{U}^\top)) \end{aligned}$$

This special case of unipartite graph was considered previously in the nonparametric Bayesian Latent Feature Relational Model (LFRM) proposed in [20]. The LFRM however cannot be applied for the bipartite case unlike our model, and does not have properties of relevant object selection or exploiting pairwise similarities between objects which we will now describe. We use the LFRM as one of the baselines in our experiments.

A. Relevance Selection Mechanism

In a real-world interaction network, there may be some objects that are irrelevant (or noisy) and should not belong to any cluster. For example, in a social network, a user might be a spammer who indiscriminately subscribes to

many random users, thereby forming many spurious links in the adjacency matrix. Without safeguarding the stochastic blockmodel against such irrelevant objects, it can lead to bad parameter estimates (e.g., overestimating the number of clusters, or yielding false positives in the link prediction tasks, etc.). Our experiments in Section VI corroborate this. To deal with such irrelevant objects in a principled way, we propose using a relevant object selection mechanism.

Our relevant object selection mechanism is based on maintaining two random binary vectors $\boldsymbol{R}^A \in \{0,1\}^{N\times 1}$ and $\boldsymbol{R}^B \in \{0,1\}^{M\times 1}$ for sets $\mathcal{A}$ and $\mathcal{B}$ respectively. Here, we will only discuss the bipartite case (for the unipartite case, there will be only a single random vector $\boldsymbol{R} \in \{0,1\}^{N\times 1}$). Our proposal is inspired by subset feature selection methods used in (non-overlapping) clustering [13] and factor analysis [27].

In the relevance selection variant of our model, we will assume a background noise link probability $\phi \sim \mathcal{B}et(a,b)$, where $\mathcal{B}et(a,b)$ denotes the Beta distribution with parameters a and b. If one or both objects $n \in \mathcal{A}$ and $m \in \mathcal{B}$ are irrelevant ($R^A_n = 0$ and/or $R^B_m = 0$), we assume that A_{nm} is drawn from a Bernoulli distribution with parameter ϕ. If both n and m are relevant ($R^A_n = 1$ and $R^B_m = 1$), then A_{nm} is drawn from a Bernoulli distribution with parameter $p = \sigma(\boldsymbol{u}_n \boldsymbol{W} \boldsymbol{v}_m^\top)$. Here, we will slightly abuse notation and use $\boldsymbol{u}_n \sim \mathcal{IBP}(\alpha_u)$ (resp. $\boldsymbol{v}_m \sim \mathcal{IBP}(\alpha_v)$) to denote that each row of $\boldsymbol{U}$ (resp. $\boldsymbol{V}$) is drawn from the IBP. The full generative model is given below:

$$
\begin{aligned}
\phi &\sim \mathcal{B}et(a,b) \\
\rho^A_n &\sim \mathcal{B}et(c,d), \quad \rho^B_m \sim \mathcal{B}et(e,f) \\
R^A_n &\sim \mathcal{B}er(\rho^A_n), \quad R^B_m \sim \mathcal{B}er(\rho^B_m) \\
\boldsymbol{u}_n &\sim \mathcal{IBP}(\alpha_u) \quad \text{if } R^A_n = 1\text{; zeros otherwise} \\
\boldsymbol{v}_m &\sim \mathcal{IBP}(\alpha_v) \quad \text{if } R^B_m = 1\text{, zeros otherwise} \\
p &= \sigma(\boldsymbol{u}_n \boldsymbol{W} \boldsymbol{v}_m^\top) \\
A_{nm} &\sim \mathcal{B}er(p^{R^A_n R^B_m} \phi^{1-R^A_n R^B_m})
\end{aligned}
$$

In the above generative model, it can be easily seen that if $R^A_n = 0$ or $R^B_m = 0$ (i.e., one or both of n and m are irrelevant) then the link A_{nm} is drawn from $\mathcal{B}er(\phi)$ (i.e., the background noise model). On the other hand, if both R^A_n and R^B_m are 1 (i.e., both n and m are relevant) then the link A_{nm} is drawn from $\mathcal{B}er(p) = \mathcal{B}er(\sigma(\boldsymbol{u}_n \boldsymbol{W} \boldsymbol{v}_m^\top))$ using their cluster assignment vectors $\boldsymbol{u}_n$ and $\boldsymbol{v}_m$.

B. Exploiting Pairwise Similarities

The model described so far only makes use of the adjacency graph. We now describe how our model can exploit additionally available pairwise object similarities given in the form of kernel matrices $\boldsymbol{S}^A$ and $\boldsymbol{S}^B$. The goal is to encourage two objects having a high similarity to have similar cluster membership vectors. In other words, if two objects n and n' in set $\mathcal{A}$ have a high pairwise similarity (i.e., a high value $S^A_{nn'}$) then the cluster membership vectors $\boldsymbol{u}_n$ and $\boldsymbol{u}_{n'}$ should also be similar. Likewise, if two objects m and m' in set $\mathcal{B}$ have a high pairwise similarity then their cluster membership vectors $\boldsymbol{v}_m$ and $\boldsymbol{v}_{m'}$ should be similar (see Figure 2 for an illustration).

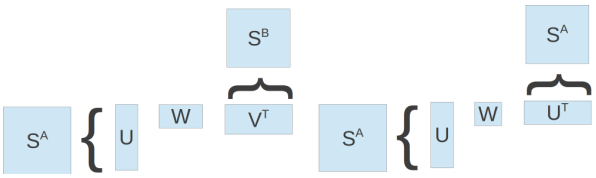

Figure 2. The similarity based smoothing of cluster membership vectors. Note that row n in $\boldsymbol{U}$ denotes the cluster membership vector of object n in set $\mathcal{A}$. Likewise, row m in $\boldsymbol{V}$ denotes the cluster membership vector of object m in set $\mathcal{B}$. **Left:** For the bipartite case where the adjacency matrix is decomposed as $\boldsymbol{UWV}^\top$, using the similarity matrix $\boldsymbol{S}^A$ (resp. $\boldsymbol{S}^B$) encourages rows of $\boldsymbol{U}$ (resp. rows of $\boldsymbol{V}$) to be correlated depending on how similar the corresponding objects are. **Right:** The same idea for the unipartite graph case where the adjacency matrix is assumed to be decomposed as $\boldsymbol{UWU}^\top$.

The IBP based model for overlapping clustering does not take the pairwise similarity information into account. Recall that in the standard IBP culinary analogy, the customer n chooses an existing dish with a probability proportional to *how many* other customers have chosen that dish, regardless of the similarity of this customer with those customers. Also, the number of new dishes selected by customer n is given by $Poisson(\alpha/n)$ which again does not depend on how similar/dissimilar this customer is w.r.t. the other customers.

Intuitively, we would like to have a scheme that encourages a customer to select a dish if the customer has a high similarity with all other customers who chose that dish. Also, when it comes to selecting the new dishes, we would like the number of new dishes to be large if the customer has low similarity with the other customers (so it is more desirable to choose its own set of new dishes). In the context of overlapping clustering, it would mean that we want an object n to get membership in cluster k if object n has a high similarity with the other objects which belong to cluster k. Furthermore, we would want the number of new clusters for object n to be influenced by how similar/dissimilar it is w.r.t. the other objects. A high aggregate dissimilarity w.r.t. all other objects would encourage the number of new clusters for this object to be large.

To accomplish this, we modify the sampling scheme in the IBP based generative model to exploit the pairwise similarity information. In particular, we modify the IBP model in both sampling steps (selecting existing dishes and selecting the number of new dishes). Here, we switch the terminology and will talk directly in terms of objects (for customers) and clusters (for dishes). The generative model for object-cluster assignments will be as follows:

1) The probability that object n (assuming it belongs to

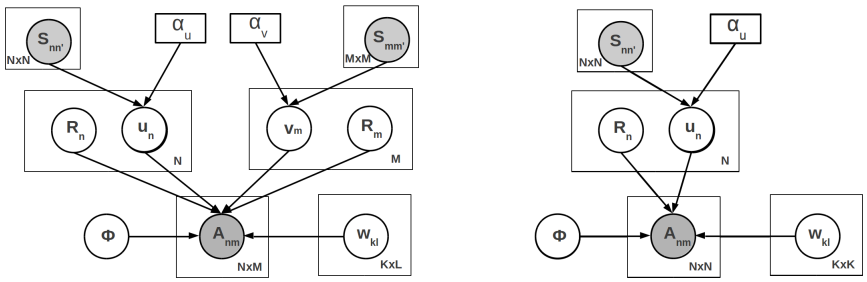

Figure 3. The full model ROCS with relevant object selection mechanism and exploiting pairwise similarities. Some of the low-level parameters and hyperparameters are not shown for the sake of brevity. **Left:** The full model for the bipartite case. **Right:** The full model for the unipartite case.

set $\mathcal{A}$) gets membership in cluster k will be proportional to $\frac{\sum_{n' \neq n} S^A_{nn'} u_{n'k}}{\sum_{n'=1}^{n} S^A_{nn'}}$. Intuitively, it means that we do not simply count *how many* of the other objects belong to cluster k but rather use a "weighted count" using the pairwise similarity scores. One can think of $\sum_{n'=1}^{n} S^A_{nn'}$ as the "effective" total number of objects, and $\sum_{n' \neq n} S^A_{nn'} u_{n'k}$ as the effective number of objects (other than n) that belong to cluster k.

2) The number of new clusters for object n is given by $Poisson(\alpha / \sum_{n'=1}^{n} S^A_{nn'})$. Intuitively, it means that if the object n has low similarities with the previous objects, the model encourages it to get memberships in its own new clusters, rather than sharing existing clusters with them.

Note that in the absence of pairwise similarity information, we can assume that $\boldsymbol{S}^A$ is a matrix of all 1s (all pairs are equivalent in terms of similarity). In this case, in step 1, the probability of object n getting membership in cluster k turns out to be $\sum_{n' \neq n} u_{n'k}/n = m_k/n$, which is identical to the standard IBP. Also, in step 2, the number of new clusters object n gets memberships into is given by $Poisson(\alpha/n)$ which is again the same as in standard IBP. We refer to the similarity information augmented variant of the IBP as $Sim\mathcal{IBP}(\alpha_u, \boldsymbol{S}^A)$. We have a similar prior distribution for the cluster memberships $\boldsymbol{V}$ for objects in set $\mathcal{B}$ and denote it by $Sim\mathcal{IBP}(\alpha_v, \boldsymbol{S}^B)$.

We would like to also note here that, even in the absence of additionally provided pairwise similarity matrix, one could use the Jaccard index [34] (percentage of common neighbors in the adjacency graph) as the entries in the similarity matrix. The Jaccard index is defined as follows:

$$S_{ij} = \frac{|\mathcal{N}(i) \cap \mathcal{N}(j)|}{|\mathcal{N}(i) \cup \mathcal{N}(j)|}$$

where $\mathcal{N}(i)$ denotes the set of neighbors of object i.

The full generative model with relevant object selection and pairwise similarity information is as follows:

$$\begin{aligned}
\phi &\sim \mathcal{Bet}(a, b) \\
\rho^A_n &\sim \mathcal{Bet}(c, d), \quad \rho^B_m \sim \mathcal{Bet}(e, f) \\
R^A_n &\sim \mathcal{Ber}(\rho^A_n), \quad R^B_m \sim \mathcal{Ber}(\rho^B_m) \\
\boldsymbol{u}_n &\sim Sim\mathcal{IBP}(\alpha_u, \boldsymbol{S}^A) \quad \text{if } R^A_n = 1\text{; zeros otherwise} \\
\boldsymbol{v}_m &\sim Sim\mathcal{IBP}(\alpha_v, \boldsymbol{S}^B) \quad \text{if } R^B_m = 1\text{; zeros otherwise} \\
p &= \sigma(\boldsymbol{u}_n \boldsymbol{W} \boldsymbol{v}_m^\top) \\
A_{nm} &\sim \mathcal{Ber}(p^{R^A_n R^B_m} \phi^{1 - R^A_n R^B_m})
\end{aligned}$$

In the case of a unipartite graph, we have a single similarity matrix $\boldsymbol{S}$ and the cluster membership vectors $\boldsymbol{u}_n$ are drawn from $Sim\mathcal{IBP}(\alpha_u, \boldsymbol{S})$. The graphical model in plate notation is shown in Figure 3. We call our model ROCS (abbreviated for **R**elevance-based **O**verlapping **C**lustering with **S**imilarity-based-smoothing).

Our idea of using pairwise similarities to regularize the latent representations (cluster membership vectors) of objects is similar in spirit to recent work on kernelized probabilistic matrix factorization [36], [11]. In kernelized probabilistic matrix factorization (KPMF) [36], the idea is to incorporate the side information through kernel matrix into the matrix factorization process such that the rows in the latent factor matrix (each row represents the latent factor of an object) are no longer independent. In particular, the KPMF model introduces dependence among the rows of the factor matrix by drawing them from a Gaussian Process (GP) prior using the provided kernel matrix as the GP covariance matrix, instead of a Gaussian prior with identity covariance matrix. The KPMF, however, assumes real-valued latent representations unlike our case where the latent representations are in form of binary cluster membership vectors. Moreover, KMPF does not perform relevant object selection and also needs the number of factors to be specified *a priori*. We use the KMPF model as one of the baselines in our experiments.

Finally, some other recently proposed variants of the IBP can also take into account object-specific features [35] or directly available pairwise similarities/distances between objects [9]. For example, one could replace the IBP by a recently proposed variation of the IBP called the Distance

Dependent Latent Feature Model [9]. This variation tries to accomplish a similar effect as our proposed modification of the IBP in this paper, albeit in a slightly different manner. The Distance Dependent variant in [9] is based on checking, for each dish, whether customer n can be "reached" to the "owner" of that dish. If it does then customer n inherits that dish. We leave the evaluation of these variants, in the context of stochastic blockmodels, to future work.

V. Inference

Exact inference in the Indian Buffet Process based models is intractable [12] and therefore we use approximate inference using MCMC (Gibbs sampling with some Metropolis-Hastings steps). In this section, we provide the sampling equations for cluster membership matrix $\boldsymbol{U}$ (sampling $\boldsymbol{V}$ is similar), the matrix $\boldsymbol{W}$ which controls the link probabilities, and the relevance vectors $\boldsymbol{R}^A$ (sampling $\boldsymbol{R}^B$ is similar). As a shorthand, we will denote by Θ the set of all the random variables $\{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{R}^A, \boldsymbol{R}^B, \phi\}$ and the hyperparameters in our model. We collapsed some of the hyperparameters for efficient Gibbs sampling.

Sampling $\boldsymbol{U}$: For each cluster k, with all other random variables fixed, the posterior probability that object n gets membership in this cluster is:

$$P(u_{nk} = 1|\boldsymbol{A}, \boldsymbol{S}^A) \propto P(u_{nk} = 1|\alpha_u, \boldsymbol{S}^A) P(\boldsymbol{A}|\Theta, u_{nk} = 1)$$

where $P(u_{nk} = 1|\alpha_u, \boldsymbol{S}^A) = \frac{\sum_{n' \neq n} S^A_{nn'} u_{n'k}}{\sum_{n' \neq n} S^A_{nn'}}$ is the prior probability of object n getting membership in an existing cluster k as per the (modified) IBP prior distribution. The likelihood term is given by $P(\boldsymbol{A}|\Theta) = \prod_{nm} P(A_{nm}|\Theta) = \prod_{nm} \mathcal{B}er(A_{nm}|p^{R^A_n R^B_m} \phi^{1 - R^A_n R^B_m})$.

The posterior probability that object n does not get membership in an existing cluster k is:

$$P(u_{nk} = 0|\boldsymbol{A}, \boldsymbol{S}^A) \propto P(u_{nk} = 0|\alpha_u, \boldsymbol{S}^A) P(\boldsymbol{A}|\Theta, u_{nk} = 0)$$

where $P(u_{nk} = 0|\alpha_u, \boldsymbol{S}^A) = \frac{\sum^n_{n' \neq n} S^A_{nn'} - \sum^n_{n' \neq n} S^A_{nn'} u_{n'k}}{\sum^n_{n' \neq n} S^A_{nn'}}$.

We sample the number of new clusters that object n gets memberships into using a Metropolis-Hastings procedure [27]. For this, we draw a number k_{new} of new clusters where $k_{new} = Poisson(\alpha / \sum_{n'} S^A_{nn'})$ and accept/reject this number based on the acceptance probability.

Sampling $\boldsymbol{W}$: Since the Gaussian prior distribution on $\boldsymbol{W}$ is not conjugate to the likelihood, sampling $\boldsymbol{W}$ directly from its posterior distribution is not possible. Therefore, we use Metropolis-Hastings sampling to sample each entry of $\boldsymbol{W}$ by proposing its new value from a Gaussian centered around the old value, and accept/reject it based on the acceptance probabilities.

Sampling Object Relevance Vector: To sample the object relevance vector $\boldsymbol{R}^A$, we integrate out the Beta distributed parameters ρ^A_n and get the following sampling distribution for the relevance variable for object n:

$$\begin{aligned} P(R^A_n = 1|\boldsymbol{A}) &\propto (c + \sum_{n' \neq n} R^A_n) P(\boldsymbol{A}|\Theta, R^A_n = 1) \\ P(R^A_n = 0|\boldsymbol{A}) &\propto (d + N - \sum_{n' \neq n} R^A_n) P(\boldsymbol{A}|\Theta, R^A_n = 0) \end{aligned}$$

Sampling the Background Noise Probability ϕ: The background noise probability has a Beta posterior distribution of the following form:

$$\begin{aligned} P(\phi|\boldsymbol{A}, \boldsymbol{R}^A, \boldsymbol{R}^B) &= \mathcal{B}et(a + t_{on}, b + t_{off}) \\ t_{on} &= \sum_{nm} (1 - R^A_n R^B_m) A_{nm} \\ t_{off} &= \sum_{nm} (1 - R^A_n R^B_m)(1 - A_{nm}) \end{aligned}$$

The variables in the posterior distribution can be interpreted as follows: t_{on} is a count of how many entries of A_{nm} in $\boldsymbol{A}$ are 1 given that one or both of objects n and m are irrelevant. Likewise, t_{off} is a count of how many entries in $\boldsymbol{A}$ are 0 for which either one or both the associated objects are irrelevant.

Initialization Strategies: The problem of overlapping clustering has a solution space that is combinatorial in size: for N objects and K clusters, the search-space is of size 2^{NK} and consequentially the posterior distribution of the cluster assignment matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ can be highly multimodal. The problem becomes even more compounded for the case of nonparametric Bayesian priors for which the number of parameters (K in this case) can grow adaptively as the inference procedure is running. Although more sophisticated MCMC inference methods such as split-merge sampling could be used [21], even a sensible initialization of the basic Gibbs sampling based inference procedure can often yield equally good results. In the IBP based stochastic blockmodels, two initialization schemes have been suggested in prior work: (1) running a disjoint clustering model on the data and use this as an initialization point [20], and (2) sequential initialization [25] where we add one object at a time and, given the (overlapping) cluster assignments of the previous objects, sample the cluster assignments for this new object. We use scheme (2) in our experiments, which is slower than (1), but tends to work better empirically.

Prediction: Our generative model can naturally deal with missing data and, apart from doing overlapping clustering, one application of our proposed model is link prediction where we want to predict the missing entries in $\boldsymbol{A}$ based on the observed entries. Given T samples of $\Theta = \{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{R}^A, \boldsymbol{R}^B\}$ and other random variables/hyperparameters from the MCMC run, and training data in form of the observed entries $\boldsymbol{A}_{train}$, we predict the missing entries in $\boldsymbol{A}$ by averaging the predictions made

using each MCMC sample:

$$P(A_{nm} = 1|\boldsymbol{A}_{train}) = \frac{1}{T}\sum_{i=1}^{T} P(A_{nm} = 1|\Theta^{(i)})$$

where $P(A_{nm} = 1|\Theta) = \mathcal{B}er(p^{R_n^A R_m^B}\phi^{1-R_n^A R_m^B})$, $p = \sigma(\boldsymbol{u}_n \boldsymbol{W} \boldsymbol{v}_m^\top)$, and ϕ is the estimated background noise probability. Using the model therefore also gives a measure of confidence (in terms of link probabilities) in the predictions.

VI. Experiments

We apply our ROCS model on a synthetic and several real-world datasets (both bipartite and unipartite networks) from social networks and biology domains, and compare against a number of state-of-the-art methods.

In our experiments, we are interested in assessing how well our model is able to infer the correct number of clusters, can identify relevant/irrelevant objects, can make use of the pairwise similarity information (when available) to improve overlapping clustering, and how well it can predict the missing links in the network. As done in other recent works on stochastic blockmodels [20], [21], for the link prediction task, we hide 50% of the entries in the adjacency matrix and predict the rest 50% using the model. We use the 0-1 test error and Area under the ROC Curve (AUC) as our performance metrics (the networks used are highly sparse). For datasets with known number of clusters, we also report the discovered number of clusters by our method. For the baselines that require this number to be specified, we provide either the ground truth number of clusters (when known), or report the results using the best choice of this number for that baseline.

We repeat each experiment 10 times, each time using a different random mask for missing links, and report the averaged results. We first provide a description of the baselines and then present the experimental results on the various datasets.

A. Baselines

We compare our model with a number of state-of-the-art methods for overlapping clustering and link prediction. We list our baselines below:

- **Overlapping Clustering using Nonnegative Matrix Factorization (OCNMF) [26]:** This method assumes that the adjacency matrix admits a nonnegative factorization into two low-rank matrices. We use the Bayesian NMF method in [26] which also learns the number of clusters using a shrinkage mechanism. This method however cannot do relevant object selection or exploit pairwise object similarities.
- **Kernelized Probabilistic Matrix Factorization (KPMF) [36]:** We modified the original model in [36], which is applicable only for real-valued data, to deal with binary adjacency matrix. Although this method can make use of side information in the form of similarity matrices over the rows and columns, the learned factor matrices are real-valued so the method can be applied for link prediction but not for overlapping clustering. Moreover, the rank of the factor matrices needs to be specified under the KPMF model.
- **Bayesian Community Detection (BCD) [21]:** This is a state-of-the-art disjoint clustering based stochastic blockmodel. The model is an extension of the Infinite Relational Model (IRM) [16] for stochastic blockmodeling that assumes non-overlapping clustering but learns the number of clusters from data using a nonparametric Bayesian prior distribution (the Dirichlet Process mixture model [29]). The method however cannot do relevant object selection or exploit pairwise object similarities.
- **Latent Feature Relational Model (LFRM) [20]:** This is a state-of-the-art overlapping clustering method. It learns the number of clusters from data (using the IBP prior on the object-cluster assignment matrix) but does not perform relevant object selection and does not make use of pairwise object similarities. Moreover, LFRM is applicable only for unipartite graphs [20]. For experiments on bipartite graphs, we replace this baseline with our own model without pairwise similarities and relevance selection which is equivalent to the bipartite version of LFRM (we will still refer it by LFRM).

Of these baselines, the first two (OCNMF and KPMF) compute a point estimate of the model parameters whereas the latter two (BCD and LFRM) produce samples from the posterior distribution of the model parameters. So, for BCD and LFRM, test error and AUC scores are computed by averaging over the posterior samples.

We would like to note here that the mixed-membership stochastic blockmodel [4] (MMSB) is another relevant baseline. However, we do not compare with MMSB because LFRM (which we use as one of the baselines) has been shown in the recent prior work [20], [25] to outperform MMSB on the kind of problems and datasets we are considering in this paper.

B. Datasets and Experimental Results

Synthetic Data: Using a synthetic data set that includes a significant fraction of irrelevant objects, we first show that our model can successfully identify irrelevant objects and therefore results in more accurate overlapping clustering. For this data, we do not have pairwise object similarities and only want to focus on identifying the relevant/irrelevant objects.

We generated the synthetic dataset as a unipartite network with three overlapping clusters (shown in Figure 4). In this dataset, there are 50 objects of which the first 30 objects are relevant (these objects belong to at least one cluster) and

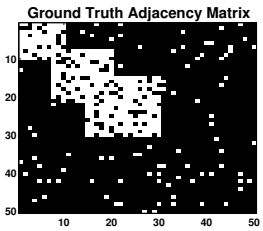

Figure 4. The synthetic data of 30 relevant and 20 irrelevant objects with three overlapping clusters.

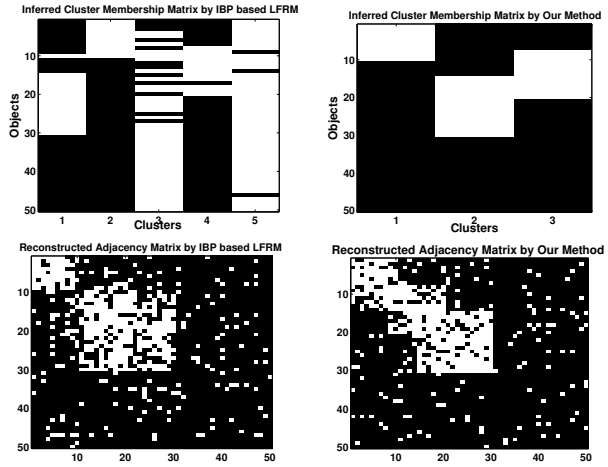

Figure 5. Synthetic data experiment: Inferred overlapping clustering (Top row) and reconstructed adjacency matrix by LFRM and ROCS in the presence of irrelevant objects (Bottom row). The ground truth number of clusters is 3 in this dataset (Figure 4 shows the true adjacency matrix). LFRM not only overestimates the number of clusters, but also assigns irrelevant objects to some cluster(s). On the other hand, our model correctly infers the number of clusters as well as the irrelevant objects.

the remaining 20 objects are irrelevant (the links involving one or both of such irrelevant objects are drawn from a background noise distribution). To achieve this effect, we created three dense blocks in the adjacency matrix such that they overlap and span the first 30 objects. The ones in the remaining part of our 50×50 adjacency matrix represent noisy links (we sample each of them using a background noise probability of 0.05). Table I shows the link prediction results. In terms of both 0-1 test error and AUC score, ROCS is better than the other methods.

Table I
SYNTHETIC DATA

Method	0-1 Test Error (%)	AUC
OCNMF	44.82 ($\pm$12.59)	0.7164 ($\pm$0.1987)
KPMF	39.70 ($\pm$1.78)	0.6042 ($\pm$0.0517)
BCD	20.05 ($\pm$1.49)	0.8504 ($\pm$0.0197)
LFRM	9.59 ($\pm$0.36)	0.8619 ($\pm$0.0374)
ROCS	**9.05 ($\pm$0.42)**	**0.8787 ($\pm$ 0.0303)**

The overlapping clustering results of LFRM and ROCS are shown in Figure 5. As the figure shows, ROCS identifies the correct number of clusters even in the presence of noisy objects. Moreover, it perfectly identifies which objects are irrelevant, leading to an improved overlapping clustering (Figure 5, top-right). On the other hand, the LFRM overestimates the number of clusters, and wrongly infers the irrelevant objects as relevant ones (it assigns them to one or more clusters). Although not shown here, BCD also assigns all the irrelevant objects to some cluster(s). Note that for the overlapping clustering task, we cannot directly compare the results with OCNMF because it produces soft-partitioning solutions. Also note that KPMF is not applicable for overlapping clustering.

Table II
FACEBOOK DATA

Method	0-1 Test Error (%)	AUC
OCNMF	36.58 ($\pm$19.74)	0.7215 ($\pm$0.1666)
KPMF	35.76 ($\pm$2.76)	0.7013 ($\pm$0.0174)
BCD	13.59 ($\pm$0.31)	0.9187 ($\pm$0.0242)
LFRM	12.38 ($\pm$2.82)	0.9156 ($\pm$0.0134)
ROCS	**11.96 ($\pm$1.44)**	**0.9388 ($\pm$ 0.0156)**

Facebook Data: This is a real-world dataset extracted from the Facebook online social network. Specifically, this dataset is an ego-network in Facebook from [18]. Given a user (also called ego), an ego-network is constructed by extracting the subgraph which is induced by the ego and the ego's friends. The resulting network represents user-user interactions around the ego node. There are 228 nodes and the known number of communities is 14 in this ego-network. Also, in Facebook, a user can provide profile information (e.g., age, gender, education information, etc.). By selecting some informative attributes in this profile information, we create a feature vector for each user. We select 92 features, and based on the feature vectors, we create a user-user similarity matrix by computing the similarity between each pair of the users using the Gaussian kernel.

The results are shown in Table II. As we can see, our model yields better results (in predicting the missing links as measured by the 0-1 error and AUC scores) than the other baselines. Although BCD did roughly comparably to our method in terms of predicting the missing entries, it overestimated the number of clusters (20-22 across multiple runs), whereas the number of clusters discovered by both LFRM and our model was close to the ground truth (13-15 across multiple runs).

Table III
DRUG-PROTEIN INTERACTION DATA

Method	0-1 Test Error (%)	AUC
KPMF	16.65 ($\pm$ 0.36)	0.8734 ($\pm$ 0.0133)
LFRM	2.75 ($\pm$ 0.04)	0.9032 ($\pm$ 0.0156)
ROCS	**2.31 ($\pm$ 0.06)**	**0.9276 ($\pm$ 0.0142)**

Drug-Protein Interaction Data: Our next experiment is on a drug-protein interaction biological network from [1].

The drug-protein interaction network is a bipartite graph which represents interactions between 200 drug molecules and 150 target proteins. In addition to the interaction network, we also have a drug-drug similarity matrix and a protein-protein similarity matrix which are constructed based on chemical structure similarity and amino acid sequence similarity, respectively. The results are shown in Table III. We excluded the OCNMF and BCD baselines from the comparison because they are not applicable for bipartite graphs. Since the IBP based LFRM model proposed in [20] is also not applicable for bipartite graphs, we use our model without similarity information and refer this variant of our model as LFRM for this dataset. As we can see from the table, our method which makes use of the protein-protein and drug-drug similarities leads to much better predictions than both the other baselines: LFRM which uses IBP to learn the number of clusters but ignores the similarity information, and KPMF which takes into account the similarity information but does not assume overlapping clustering (it is a latent factor model and needs the number of latent factors to be specified as well; for KPMF we used the number of latent factors that gave the best results).

Lazega Lawyers Data: The Lazega lawyers dataset [17] is a directed, unipartite graph representing the social network between 71 partners and associates in some New England law firms. In addition, each entity in the network is described by features such as gender, office-location, age, years employed, etc. We did some preprocessing of the features (binarized the features such as the age and years employed) and then constructed a kernel matrix of pairwise similarities. Each entry of the kernel matrix measures what fraction of the features match between two partners. The results on this dataset are shown in Table IV. As we can see, even a weak similarity information can yield reasonable improvements in the prediction accuracy when compared to the best performing baseline of LFRM.

Table IV
LAZEGA-LAWYERS DATA (FRIENDSHIP NETWORK)

Method	0-1 Test Error (%)	AUC
OCNMF	35.36 (±20.71)	0.6388 (±0.1527)
KPMF	34.69 (±1.13)	0.7203 (±0.0229)
BCD	16.58 (±0.56)	0.7876 (±0.0168)
LFRM	14.05 (± 2.04)	0.8025 (± 0.0205)
ROCS	**12.98 (± 0.32)**	**0.8248 (± 0.01642)**

VII. RELATED WORK

For doing overlapping clustering in the framework of stochastic blockmodels, the most similar in spirit to our work is the nonparametric Bayesian Latent Feature Relational Model [20] which was also one of our baselines. However, as discussed earlier in the paper, LFRM does not have a mechanism to deal with noisy, irrelevant objects and can not exploit pairwise similarity information that may be available for many real-world datasets. Moreover, the LFRM is defined only for unipartite graphs so it can not be applied to bipartite graph clustering. Other extensions of LFRM include a recently proposed two-level model [25] which assumes that each cluster further consists of subclusters, and a model that uses noisy-OR likelihood model instead of the Bernoulli likelihood model [22] for links.

The idea of automatically selecting relevant objects in a stochastic blockmodel was also considered recently in [15]. However, it is different from our model in two ways: (1) it considers only disjoint clustering whereas we consider the overlapping clustering, and (2) it cannot exploit pairwise similarities between objects.

Non-Bayesian methods for overlapping clustering include clique percolation [24], line graph partitioning [2], ego network extraction [5], low-rank non-negative matrix factorization based modeling [33], and seed set expansion [30]. The clique percolation method assumes that a graph consists of overlapping sets of cliques, and considers adjacent cliques as overlapping clusters in the graph. In [2], a line graph is constructed from the original graph by converting edges into nodes. Then deriving clustering on the line graph yields overlapping clustering of the original graph. Coscia et al. [5] developed a local-first method which first extracts and computes clustering of ego networks of each node and then merges the local communities into a global collection. Low-rank methods also can be used to detect overlapping clusters. For example, Yang et al. [33] presented a model-based community detection algorithm using a non-negative matrix factorization. Recently, Whang et al. [30] have proposed an efficient overlapping community detection algorithm using seed set expansion. They presented effective seed finding methods and produced a set of small conductance clusters by expanding the seed sets. Compared with these non-Bayesian methods, our stochastic blockmodel allows much more flexibility including detecting an appropriate number of clusters, incorporating node attributes, and overlapping clustering of bipartite graphs. Furthermore, our model can also be applied to link prediction tasks.

Although a number of models exist for disjoint clustering of bipartite graphs [7], the problem of doing *overlapping* clustering for bipartite networks has been relatively less studied so far. In the specific context of co-clustering of general data, there have been some prior works such as [37], [6]. However, these models either do not handle relevant object selection, do not exploit pairwise object similarities, and need the number of clusters to be specified *a priori*.

VIII. DISCUSSION AND CONCLUSION

In this paper, we have presented a flexible model for modeling relational data such that each object can potentially belong to multiple clusters, irrelevant objects can be dealt with in a principled manner, and pairwise similarity

between objects can be exploited to regularize the cluster memberships of objects.

Our model can be easily extended to model multi-relational data where instead of a single relationship network, we are given multiple relations such as friendships, work-relation, advice-relations, etc [17]. This can be accomplished by having multiple cluster interaction matrices $\boldsymbol{W}$, one per relation. With this modification, the model likelihood (in our basic model without pairwise similarities) will be a product over all the M relations $\prod_{i=1}^{M} P(\boldsymbol{A}^{(i)}|\boldsymbol{U}\boldsymbol{W}^{(i)}\boldsymbol{V}^T)$, where $\{\boldsymbol{A}^{(i)}\}_{i=1}^{M}$ are the M observed relation networks. The other components of our model will remain the same.

An important avenue of future work will be making the model more scalable. In this paper, we used MCMC sampling for doing inference in our model. However, it is possible to employ a different inference machine instead of MCMC (for example, variational inference methods [14]), which will make our method faster. This would enable our model to scale up to reasonably large networks.

ACKNOWLEDGMENT

This research was supported by NSF grant CCF-1117055. PR acknowledges the Sheldon Ekland-Olson Postdoctoral Fellowship from UT Austin; most of PR's contribution to this work happened during this postdoctoral appointment.

REFERENCES

[1] Drug-target Interaction Network Inference Engine based on Supervised Analysis. http://www.genome.jp/tools/dinies/help.html.

[2] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466:761–764, August 2010.

[3] C. Aicher, A. Z. Jacobs, and A. Clauset. Adapting the stochastic block model to edge-weighted networks. *arXiv preprint arXiv:1305.5782*, 2013.

[4] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *JMLR*, 2008.

[5] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi. Demon: a local-first discovery method for overlapping communities. In *KDD*, pages 615–623, 2012.

[6] M. Deodhar, G. Gupta, J. Ghosh, H. Cho, and I. Dhillon. A scalable framework for discovering coherent co-clusters in noisy data. In *ICML*, 2009.

[7] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, 2001.

[8] S. J. Gershman and D. M. Blei. A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology*, 2012.

[9] S. J. Gershman, P. I. Frazier, and D. M. Blei. Distance dependent infinite latent feature models. *arXiv preprint arXiv:1110.5454*, 2011.

[10] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Foundations and Trends® in Machine Learning*, 2010.

[11] M. Gönen, S. A. Khan, and S. Kaski. Kernelized Bayesian matrix factorization. ICML, 2013.

[12] T. L. Griffiths and Z. Ghahramani. The Indian buffet process: An introduction and review. *JMLR*, 2011.

[13] P. D. Hoff. Subset clustering of binary sequences, with an application to genomic abnormality data. *Biometrics*, 2005.

[14] M. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *arXiv preprint arXiv:1206.7051*, 2012.

[15] K. Ishiguro, N. Ueda, and H. Sawada. Subset infinite relational models. AISTATS, 2012.

[16] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, 2006.

[17] E. Lazega. *The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership*. Oxford University Press on Demand, 2001.

[18] J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *NIPS*, 2012.

[19] A. K. Menon and C. Elkan. Link prediction via matrix factorization. In *ECML PKDD*. 2011.

[20] K. Miller, T. Griffiths, and M. Jordan. Nonparametric latent feature models for link prediction. *NIPS*, 2009.

[21] M. Mørup and M. N. Schmidt. Bayesian community detection. *Neural Computation*, 24(9):2434–2456, 2012.

[22] M. Mørup, M. N. Schmidt, and L. K. Hansen. Infinite multiple membership relational modeling for complex networks. In *Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on*, pages 1–6. IEEE, 2011.

[23] K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *JASA*, 2001.

[24] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, June 2005.

[25] K. Palla, D. Knowles, and Z. Ghahramani. An infinite latent attribute model for network data. In *ICML*, 2012.

[26] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon. Overlapping community detection using Bayesian non-negative matrix factorization. *Physical Review E*, 2011.

[27] P. Rai and H. Daumé III. The infinite hierarchical factor regression model. NIPS, 2008.

[28] M. Salter-Townshend, A. White, I. Gollini, and T. B. Murphy. Review of statistical network analysis: models, algorithms, and software. *Statistical Analysis and Data Mining*, 2012.

[29] Y. W. Teh. Dirichlet process. *Encyclopedia of machine learning*, pages 280–287, 2010.

[30] J. J. Whang, D. F. Gleich, and I. S. Dhillon. Overlapping community detection using seed set expansion. In *CIKM*, 2013.

[31] J. Xie, S. Kelley, and B. K. Szymanski. Overlapping community detection in networks: the state of the art and comparative study. *ACM Computing Surveys*, 45(4), 2013.

[32] Z. Xu, V. Tresp, K. Yu, and H.-P. Kriegel. Learning infinite hidden relational models. In *UAI*, 2006.

[33] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *WSDM*, 2013.

[34] Y. Zhao, E. Levina, and J. Zhu. Link prediction for partially observed networks. *arXiv preprint arXiv:1301.7047*, 2013.

[35] M. Zhou, H. Yang, G. Sapiro, D. Dunson, and L. Carin. Dependent hierarchical beta process for image interpolation and denoising. In *AISTATS*, 2011.

[36] T. Zhou, H. Shan, A. Banerjee, and G. Sapiro. Kernelized probabilistic matrix factorization: Exploiting graphs and side information. In *SDM*, 2012.

[37] H. Zhu, G. Mateos, G. B. Giannakis, N. D. Sidiropoulos, and A. Banerjee. Sparsity-cognizant overlapping co-clustering for behavior inference in social networks. In *ICASSP*, 2010.