# Feature Enriched Nonparametric Bayesian Co-clustering

Pu Wang, Carlotta Domeniconi, Huzefa Rangwala, and Kathryn B. Laskey

George Mason University
4400 University Ave., Fairfax, VA 22030 USA
{pwang7,carlotta,rangwala}@cs.gmu.edu
klaskey@gmu.edu

**Abstract.** Co-clustering has emerged as an important technique for mining relational data, especially when data are sparse and high-dimensional. Co-clustering simultaneously groups the different kinds of objects involved in a relation. Most co-clustering techniques typically only leverage the entries of the given contingency matrix to perform the two-way clustering. As a consequence, they cannot predict the interaction values for new objects. In many applications, though, additional features associated to the objects of interest are available. The *Infinite Hidden Relational Model* (IHRM) has been proposed to make use of these features. As such, IHRM has the capability to forecast relationships among previously unseen data. The work on IHRM lacks an evaluation of the improvement that can be achieved when leveraging features to make predictions for unseen objects. In this work, we fill this gap and re-interpret IHRM from a co-clustering point of view. We focus on the empirical evaluation of forecasting relationships between previously unseen objects by leveraging object features. The empirical evaluation demonstrates the effectiveness of the feature-enriched approach and identifies the conditions under which the use of features is most useful, i.e., with sparse data.

**Keywords:** Bayesian Nonparametrics; Dirichlet Processes; Co-clustering; Protein-molecule interaction data

## 1 Introduction

Co-clustering [11] has emerged as an important approach for mining relational data. Often, data can be organized in a matrix, where rows and columns present a symmetrical relation. Co-clustering simultaneously groups the different kinds of objects involved in a relation; for example, proteins and molecules indexing a contingency matrix that holds information about their interaction. Molecules are grouped based on their binding patterns to proteins; similarly, proteins are clustered based on the molecules they interact with. The two clustering processes are inter-dependent. Understanding these interactions provides insight into the underlying biological processes and is useful for designing therapeutic drugs.

Existing co-clustering techniques typically only leverage the entries of the given contingency matrix to perform the two-way clustering. As a consequence, they cannot predict the interaction values for new objects. This greatly limits the applicability of current co-clustering approaches.

In many applications additional features associated to the objects of interest are available, e.g., sequence information for proteins. Such features can be

leveraged to perform predictions on new data. The *Infinite Hidden Relational Model* (IHRM) [36] has been proposed to leverage features associated to the rows and columns of the contingency matrix to forecast relationships among previously unseen data. Although IHRM was originally introduced from a relational learning point of view, it is essentially a co-clustering model that overcomes the aforementioned limitations of existing co-clustering techniques. In particular, IHRM is a nonparametric Bayesian model, which learns the number of row and column clusters from the given samples. This is achieved by assuming Dirichlet Process priors to the rows and columns of the contingency matrix. As such, IHRM does not require the *a priori* specification of the numbers of row and column clusters in the data.

Existing Bayesian co-clustering models [30, 35, 19] are related to IHRM, but none makes use of features associated to the rows and columns of the contingency matrix. As a consequence, these methods can handle missing entries only for already observed rows and columns (e.g., for a protein and a molecule used during training, although not necessarily in combination). In particular, IHRM can be viewed as an extension to the nonparametric Bayesian co-clustering (NBCC) model [19]. IHRM adds to NBCC the ability to exploit features associated to rows and columns, thus enabling IHRM to predict entries for unseen rows and/or columns. The authors in [36] have applied IHRM to collaborative filtering [27]. Co-clustering techniques have also been applied to collaborative filtering [33, 15, 10], but again none of these involve features associated to rows or columns of the data matrix.

The work on IHRM [36] lacks an evaluation of the improvement that can be achieved when leveraging features to make predictions for unseen objects. In this work, we fill this gap and re-interpret IHRM from a co-clustering point of view. We call the resulting method Feature Enriched Dirichlet Process Co-clustering (FE-DPCC). We focus on the empirical evaluation of forecasting relationships between previously unseen objects by leveraging object features.

## 2 Related Work

Researchers have proposed several discriminative and generative co-clustering models, e.g. [7, 29]. Bayesian Co-clustering (BCC) [30] maintains separate Dirichlet priors for row- and column-cluster probabilities. To generate an entry in the data matrix, the model first generates the row and column clusters for the entry from their respective Dirichlet-multinomial distributions. The entry is then generated from a distribution specific to the row- and column-cluster. Like the original Latent Dirichlet Allocation (LDA) [5] model, BCC assumes symmetric Dirichlet priors for the data distributions given the row- and column-clusters. Shan and Banerjee [30] proposed a variational Bayesian algorithm to perform inference with the BCC model. In [35], the authors proposed a variation of BCC, and developed a collapsed Gibbs sampling and a collapsed variational algorithm to perform inference. All aforementioned co-clustering models are parametric, i.e., they need to have specified the number of row- and column-clusters.

A nonparametric Bayesian co-clustering (NBCC) approach has been proposed in [19]. NBCC assumes two independent Bayesian priors on rows and

columns. As such, NBCC does not require *a priori* the number of row- and column-clusters. NBCC assumes a Pitman-Yor Process [24] prior, which generalizes the Dirichlet Process. The feature-enriched method we introduce here is an extension of NBCC, where features associated to rows and columns are used. Such features enable our technique to predict entries for unseen rows/columns.

A related work is Bayesian matrix factorization. In [17], the authors alleviated overfitting in singular value decomposition (SVD) by specifying a prior distribution over parameters, and performing variational inference. In [26], the authors proposed a Bayesian probabilistic matrix factorization method, that assigns a prior distribution to the Gaussian parameters involved in the model. These Bayesian approaches to matrix factorization are parametric. Nonparametric Bayesian matrix factorization models include [8, 32, 25].

Our work is also related to collaborative filtering (CF) [27]. CF learns the relationships between users and items using only user preferences to items, and then recommends items to users based on the learned relationships. Various approaches have been proposed to discover underlying patterns in user consumption behaviors [6, 16, 1, 18, 17, 26, 31, 12, 14]. Co-clustering techniques have already been applied to CF [33, 15, 10]. None of these techniques involve features associated to rows or columns of the data matrix. On the contrary, content-based (CB) recommendation systems [3] predict user preferences to items using user and item features. In practice, CB methods are usually combined with CF. The approach we introduce in this paper is a Bayesian combination of CF and CB.

## 3 Background: Dirichlet Process

The Dirichlet process (DP) [9] is an infinite-dimensional generalization of the Dirichlet distribution. Formally, let $S$ be a set, $G_0$ a measure on $S$, and $\alpha_0$ a positive real number. The random probability distribution $G$ on $S$ is distributed as a DP with concentration parameter $\alpha_0$ (also called the pseudo-count) and base measure $G_0$ if, for any finite partition $\{B_k\}_{1\leq k\leq K}$ of $S$: $(G(B_1), G(B_2), \cdots, G(B_K)) \sim \text{Dir}(\alpha_0 G_0(B_1), \alpha_0 G_0(B_2), \cdots, \alpha_0 G_0(B_K))$.

Let $G$ be a sample drawn from a DP. Then with probability 1, $G$ is a discrete distribution [9]. Further, if the first $N-1$ draws from $G$ yield $K$ distinct values $\theta^*_{1:K}$ with multiplicities $n_{1:K}$, then the probability of the $N^{th}$ draw conditioned on the previous $N-1$ draws is given by the Pólya urn scheme [4]:

$$\theta_N = \begin{cases} \theta^*_k, & \text{with prob } \frac{n_k}{N-1+\alpha_0}, k \in \{1, \cdots, K\} \\ \theta^*_{K+1} \sim G_0, & \text{with prob } \frac{\alpha_0}{N-1+\alpha_0} \end{cases}$$

The DP is often used as a nonparametric prior in Bayesian mixture models [2]. Assume the data are generated from the following generative procedure: $G \sim \text{Dir}(\alpha_0, G_0); \theta_{1:N} \sim G; x_{1:N} \sim \prod_{n=1}^{N} F(\cdot|\theta_n)$, where the $F(\cdot|\theta_n)$ are probability distributions known as mixture components. Typically, there are duplicates among the $\theta_{1:N}$; thus, multiple data points are generated from the same mixture component. It is natural to define a cluster as those observations generated from a given mixture component. This model is known as the *Dirichlet process mixture* (DPM) model. Although any finite sample contains only finitely many clusters,

there is no bound on the number of clusters and any new data point has non-zero probability of being drawn from a new cluster [20]. Therefore, DPM is known as an "infinite" mixture model.

The DP can be generated via the stick-breaking construction [28]. Stick-breaking draws two infinite sequences of independent random variables, $v_k \sim \text{Beta}(1, \alpha_0)$ and $\theta_k^* \sim G_0$ for $k = \{1, 2, \cdots\}$. Let $G$ be defined as:

$$\pi_k = v_k \prod_{j=1}^{k-1}(1 - v_j) \qquad\qquad G = \sum_{k=1}^{\infty} \pi_k \delta(\theta_k^*) \tag{1}$$

where $\boldsymbol{\pi} = \langle \pi_k | k = 1, 2, \cdots \rangle$ are mixing proportions and $\delta(\theta)$ is the distribution that samples the value $\theta$ with probability 1. Then $G \sim \text{Dir}(\alpha_0, G_0)$. It is helpful to use an indicator variable $z_n$ to denote which mixture component is associated with $x_n$. The generative process for the DPM model using stick-breaking is as follows (additional details on the DPM model can be found in [20, 23]):

1. Draw $v_k \sim \text{Beta}(1, \alpha_0)$, $k = \{1, 2, \cdots\}$ and calculate $\boldsymbol{\pi}$ as in Eq (1).
2. Draw $\theta_k^* \sim G_0$, $k = \{1, 2, \cdots\}$
3. For each data point $n = \{1, 2, \cdots, N\}$:
   - Draw $z_n \sim \text{Discrete}(\boldsymbol{\pi})$; Draw $x_n \sim F(\cdot | \theta_{z_n}^*)$

## 4 Feature Enriched Dirichlet Process Co-clustering

The observed data $\boldsymbol{X}$ of FE-DPCC are composed of three parts: the observed row features $\boldsymbol{X}^R$, the observed column features $\boldsymbol{X}^C$, and the observed relational features $\boldsymbol{X}^E$ between rows and columns. If there are $R$ rows and $C$ columns, then $\boldsymbol{X}^R = \langle \boldsymbol{x}_r^R | r = \{1, \cdots, R\} \rangle$, $\boldsymbol{X}^C = \langle \boldsymbol{x}_c^C | c = \{1, \cdots, C\} \rangle$, and $\boldsymbol{X}^E = \langle x_{rc}^E | r = \{1, \cdots, R\}, c = \{1, \cdots, C\} \rangle$. $\boldsymbol{X}^E$ may have missing data, i.e., some entries may not be observed.

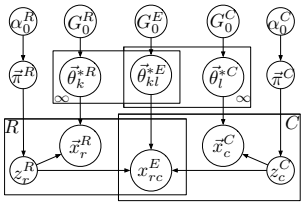


**Fig. 1:** FE-DPCC model

FE-DPCC is a generative model that assumes two independent DPM priors on rows and columns. We follow a stick-breaking representation to describe the FE-DPCC model. Specifically, assuming row and column DP priors $\text{Dir}(\alpha_0^R, G_0^R)$ and $\text{Dir}(\alpha_0^C, G_0^C)$, FE-DPCC draws row-cluster parameters $\boldsymbol{\theta}_k^{*R}$ from $G_0^R$, for $k = \{1, \cdots, \infty\}$, column-cluster parameters $\boldsymbol{\theta}_l^{*C}$ from $G_0^C$, for $l = \{1, \cdots, \infty\}$, and co-cluster parameters $\boldsymbol{\theta}_{kl}^{*E}$ from $G_0^E$, for each combination of $k$ and $l$[1]; then draws row mixture proportion $\boldsymbol{\pi}^R$ and column mixture proportion $\boldsymbol{\pi}^C$ as defined in Eq. 1. For each row $r$ and each column $c$, FE-DPCC draws the row-cluster indicator $z_r^R$ and column-cluster indicator $z_c^C$ according to $\boldsymbol{\pi}^R$ and $\boldsymbol{\pi}^C$, respectively. Further, FE-DPCC assumes the observed features of each row $r$ and each column $c$ are drawn from two parametric distributions $F(\cdot | \boldsymbol{\theta}_k^{*R})$ and $F(\cdot | \boldsymbol{\theta}_l^{*C})$, respectively, and each entry, $x_{rc}^E$, of the relational feature matrix is drawn from a parametric distribution $F(\cdot | \boldsymbol{\theta}_{kl}^{*E})$, where $z_r^R = k$ and $z_c^C = l$.

[1] Every co-cluster is indexed by a row-cluster ID and a column-cluster ID. Thus, we denote a co-cluster defined by the $k^{th}$ row-cluster and the $l^{th}$ column-cluster as $(k, l)$.

The generative process for FE-DPCC is as follows and the FE-DPCC model is illustrated in Figure 1.

1. Draw $v_k^R \sim \text{Beta}(1, \alpha_0^R)$, for $k = \{1, \cdots, \infty\}$ and calculate $\boldsymbol{\pi}^R$ as in Eq (1)
2. Draw $\boldsymbol{\theta}_k^{*R} \sim G_0^R$, for $k = \{1, \cdots, \infty\}$
3. Draw $v_l^C \sim \text{Beta}(1, \alpha_0^C)$, for $l = \{1, \cdots, \infty\}$ and calculate $\boldsymbol{\pi}^C$ as in Eq (1)
4. Draw $\boldsymbol{\theta}_l^{*C} \sim G_0^C$, for $l = \{1, \cdots, \infty\}$
5. Draw $\boldsymbol{\theta}_{kl}^{*E} \sim G_0^E$, for $k = \{1, \cdots, \infty\}$ and $l = \{1, \cdots, \infty\}$
6. For each row $r = \{1, \cdots, R\}$, draw $z_r^R \sim \text{Discrete}(\boldsymbol{\pi}^R)$, and draw $\boldsymbol{x}_r^R \sim F(\cdot|\boldsymbol{\theta}_{z_r^R}^{*R})$
7. For each column $c = \{1, \cdots, C\}$, draw $z_c^C \sim \text{Discrete}(\boldsymbol{\pi}^C)$, and draw $\boldsymbol{x}_c^C \sim F(\cdot|\boldsymbol{\theta}_{z_c^C}^{*C})$
8. For each entry $\boldsymbol{x}_{rc}^E$, draw $\boldsymbol{x}_{rc}^E \sim F(\cdot|\boldsymbol{\theta}_{z_r^R z_c^C}^{*E})$

### 4.1 Inference

The likelihood of the observed data is given by:

$$p(\boldsymbol{X}|\boldsymbol{Z}^R, \boldsymbol{Z}^C, \boldsymbol{\theta}^{*R}, \boldsymbol{\theta}^{*C}, \boldsymbol{\theta}^{*E}) = \left(\prod_{r=1}^{R} f(\boldsymbol{x}_r^R|\boldsymbol{\theta}_{z_r^R}^{*R})\right)\left(\prod_{c=1}^{C} f(\boldsymbol{x}_c^C|\boldsymbol{\theta}_{z_c^C}^{*C})\right)\left(\prod_{r=1}^{R}\prod_{c=1}^{C} f(x_{rc}^E|\boldsymbol{\theta}_{z_r^R z_c^C}^{*E})\right)$$

where $f(\cdot|\boldsymbol{\theta}_k^{*R})$, $f(\cdot|\boldsymbol{\theta}_l^{*C})$ and $f(\cdot|\boldsymbol{\theta}_{kl}^{*E})$ denote the probability density (or mass) functions of $F(\cdot|\boldsymbol{\theta}_k^{*R})$, $F(\cdot|\boldsymbol{\theta}_l^{*C})$ and $F(\cdot|\boldsymbol{\theta}_{kl}^{*E})$, respectively; $\boldsymbol{Z}^R = \langle z_r^R | r = \{1, \cdots, R\}\rangle$; $\boldsymbol{Z}^C = \langle z_c^C | c = \{1, \cdots, C\}\rangle$; $\boldsymbol{\theta}^{*R} = \langle \boldsymbol{\theta}_k^{*R} | k = \{1, \cdots, \infty\}\rangle$; $\boldsymbol{\theta}^{*C} = \langle \boldsymbol{\theta}_l^{*C} | l = \{1, \cdots, \infty\}\rangle$; and $\boldsymbol{\theta}^{*E} = \langle \boldsymbol{\theta}_{kl}^{*E} | k = \{1, \cdots, \infty\}, l = \{1, \cdots, \infty\}\rangle$.

The marginal likelihood obtained by integrating out the model parameters $\boldsymbol{\theta}^{*R}$, $\boldsymbol{\theta}^{*C}$, and $\boldsymbol{\theta}^{*E}$ is:

$$p(\boldsymbol{X}|\boldsymbol{Z}^R, \boldsymbol{Z}^C, G_0^R, G_0^C, G_0^E) = \left(\prod_{r=1}^{R} \int f(\boldsymbol{x}_r^R|\boldsymbol{\theta}_{z_r^R}^{*R}) g(\boldsymbol{\theta}_{z_r^R}^{*R}|\zeta^R) d\boldsymbol{\theta}_{z_r^R}^{*R}\right) \tag{2}$$

$$\left(\prod_{c=1}^{C} \int f(\boldsymbol{x}_c^C|\boldsymbol{\theta}_{z_c^C}^{*C}) g(\boldsymbol{\theta}_{z_c^C}^{*C}|\zeta^C) d\boldsymbol{\theta}_{z_c^C}^{*C}\right)\left(\prod_{r=1}^{R}\prod_{c=1}^{C} \int f(x_{rc}^E|\boldsymbol{\theta}_{z_r^R z_c^C}^{*E}) g(\boldsymbol{\theta}_{z_r^R z_c^C}^{*E}|\zeta^E) d\boldsymbol{\theta}_{z_r^R z_c^C}^{*E}\right)$$

where $g(\cdot|\zeta^R)$, $g(\cdot|\zeta^C)$ and $g(\cdot|\zeta^E)$ denote the probability density functions of $G_0^R$, $G_0^C$ and $G_0^E$, respectively. We assume $F(\cdot|\boldsymbol{\theta}_k^{*R})$ and $G_0^R$, $F(\cdot|\boldsymbol{\theta}_l^{*C})$ and $G_0^C$, and $F(\cdot|\boldsymbol{\theta}_{kl}^{*E})$ and $G_0^E$ are all pairwise conjugate. Thus, there is a closed form expression for the marginal likelihood (2). The conditional distribution for sampling the row-cluster indicator variable $z_r^R$ for the $r^{th}$ row $\boldsymbol{x}_r^R$ is as follows. For populated row-clusters $k \in \{Z_{r'}^R\}_{r'=\{1,\cdots,r-1,r+1,\cdots,R\}}$,

$$p(z_r^R = k|\boldsymbol{x}_r^R, \{x_{rc}^E\}_{c\in\{1,\cdots,C\}}, \boldsymbol{X}^{R\neg r}, \boldsymbol{X}^{E\neg r}, \boldsymbol{Z}^{R\neg r}) \propto \tag{3}$$

$$\frac{\mathcal{N}_k^{\neg r}}{R - 1 + \alpha_0^R}\left(\int f(\boldsymbol{x}_r^R|\boldsymbol{\theta}_k^{*R}) g(\boldsymbol{\theta}_k^{*R}|\zeta_k^{*R\neg r}) d\boldsymbol{\theta}_k^{*R}\right)\prod_{c=1}^{C}\left(\int f(x_{rc}^E|\boldsymbol{\theta}_{kz_c^C}^{*E}) g(\boldsymbol{\theta}_{kz_c^C}^{*E}|\zeta_{kz_c^C}^{*E\neg r}) d\boldsymbol{\theta}_{kz_c^C}^{*E}\right)$$

where $\neg r$ means excluding the $r^{th}$ row, $\mathcal{N}_k^{\neg r}$ is the number of rows assigned to the $k^{th}$ row-cluster excluding the $r^{th}$ row, $\zeta_k^{*R\neg r}$ is the hyperparameter of the posterior distribution of the $k^{th}$ row-cluster parameter $\boldsymbol{\theta}_k^{*R}$ given all rows assigned to the $k^{th}$ row-cluster excluding the $r^{th}$ row, and $\zeta_{kz_c^C}^{*E\neg r}$ is the hyperparameter of the posterior distribution of the co-cluster $(k, z_c^C)$ given all entries assigned to it excluding the entries in the $r^{th}$ row. When $k \notin \{z_{r'}^R\}_{r'=\{1,\cdots,r-1,r+1,\cdots,R\}}$, i.e., $z_r^R$ is being set to its own singleton row-cluster, the conditional distribution becomes:

$$p(z_r^R = k|\boldsymbol{x}_r^R, \{x_{rc}^E\}_{c\in\{1,\cdots,C\}}, \boldsymbol{X}^{R\neg r}, \boldsymbol{X}^{E\neg r}, \boldsymbol{Z}^{R\neg r}) \propto \tag{4}$$
$$\frac{\alpha_0^R}{R-1+\alpha_0^R}\left(\int f(\boldsymbol{x}_r^R|\boldsymbol{\theta}_k^{*R})g(\boldsymbol{\theta}_k^{*R}|\zeta^R)d\boldsymbol{\theta}_k^{*R}\right)\prod_{c=1}^{C}\left(\int f(x_{rc}^E|\boldsymbol{\theta}_{kz_c^C}^{*E})g(\boldsymbol{\theta}_{kz_c^C}^{*E}|\zeta_{kz_c^C}^{*E\neg r})d\boldsymbol{\theta}_{kz_c^C}^{*E}\right)$$

The conditional distribution for sampling the column-cluster indicator variable $z_c^C$ for the $c^{th}$ column $\boldsymbol{x}_c^C$ is obtained analogously. For populated column-clusters $l \in \{Z_{c'}^C\}_{c'=\{1,\cdots,c-1,c+1,\cdots,C\}}$,

$$p(z_c^C = l|\boldsymbol{x}_c^C, \{x_{rc}^E\}_{r\in\{1,\cdots,R\}}, \boldsymbol{X}^{C\neg c}, \boldsymbol{X}^{E\neg c}, \boldsymbol{Z}^{C\neg c}) \propto \tag{5}$$
$$\frac{\mathcal{N}_l^{\neg c}}{C-1+\alpha_0^C}\left(\int f(\boldsymbol{x}_c^C|\boldsymbol{\theta}_l^{*C})g(\boldsymbol{\theta}_l^{*C}|\zeta_l^{*C\neg c})d\boldsymbol{\theta}_l^{*C}\right)\prod_{r=1}^{R}\left(\int f(x_{rc}^E|\boldsymbol{\theta}_{z_r^Rl}^{*E})g(\boldsymbol{\theta}_{z_r^Rl}^{*E}|\zeta_{z_r^Rl}^{*E\neg c})d\boldsymbol{\theta}_{z_r^Rl}^{*E}\right)$$

where $\neg c$ means excluding the $c^{th}$ column, $\mathcal{N}_l^{\neg c}$ is the number of columns assigned to the $l^{th}$ column-cluster excluding the $c^{th}$ column, $\zeta_l^{*C\neg c}$ is the hyperparameter of the posterior distribution of the $l^{th}$ column-cluster parameter $\boldsymbol{\theta}_l^{*C}$ given all columns assigned to the $l^{th}$ column-cluster excluding the $c^{th}$ column, and $\zeta_{z_r^Rl}^{*E\neg c}$ is the hyperparameter of the posterior distribution of the co-cluster $(z_r^R, l)$ given all entries assigned to it excluding the entries in the $c^{th}$ column. If $z_c^C \notin \{z_{c'}^C\}_{c'=\{1,\cdots,c-1,c+1,\cdots,C\}}$, i.e., $z_c^C$ is being assigned to its own singleton column-cluster, the conditional distribution becomes:

$$p(z_c^C = l|\boldsymbol{x}_c^C, \{x_{rc}^E\}_{r\in\{1,\cdots,R\}}, \boldsymbol{X}^{C\neg c}, \boldsymbol{X}^{E\neg c}, \boldsymbol{Z}^{C\neg c}) \propto \tag{6}$$
$$\frac{\alpha_0^C}{C-1+\alpha_0^C}\left(\int f(\boldsymbol{x}_c^C|\boldsymbol{\theta}_l^{*C})g(\boldsymbol{\theta}_l^{*C}|\zeta^C)d\boldsymbol{\theta}_l^{*C}\right)\prod_{r=1}^{R}\left(\int f(x_{rc}^E|\boldsymbol{\theta}_{z_r^Rl}^{*E})g(\boldsymbol{\theta}_{z_r^Rl}^{*E}|\zeta_{z_r^Rl}^{*E\neg c})d\boldsymbol{\theta}_{z_r^Rl}^{*E}\right)$$

## 5 Experimental Evaluation

We conducted experiments on two rating datasets and two protein-molecule interaction datasets. MovieLens[2] is a movie recommendation dataset containing 100,000 ratings in a sparse data matrix for 1682 movies rated by 943 users. Jester[3] is a joke rating dataset. The original dataset contains 4.1 million continuous ratings of 140 jokes from 73,421 users. We chose a subset containing 100,000 ratings. Following [30], we uniformly discretized the ratings into 10 bins.

We also used two protein-molecule interaction datasets. The first dataset (MP1[4]) consists of G-protein coupled receptor (GPCR) proteins and their interaction with small molecules [13]. These interactions are the product of an experiment that determines whether a particular protein target is modulated by a molecule. MP1 had 4051 interactions between 166 proteins and 2687 molecules. The second dataset (MP2[5]) [21] differs from MP1 in that the protein targets belong to a more general class and are not restricted to GPCRs. The use of targets restricted to a specific group of proteins (GPCRs) is similar to a *chemogenomics*

[2] http://www.grouplens.org/node/73
[3] http://goldberg.berkeley.edu/jester-data/
[4] http://pharminfo.pharm.kyoto-u.ac.jp/services/glida/
[5] http://pubchem.ncbi.nlm.nih.gov/

**Table 2:** Average Test Perplexity

| | | MovieLens | Jester | MP1 | MP2 |
|---|---|---|---|---|---|
| DPCC | Row and Column Observed | 3.327 (0.020) | 17.111 (0.031) | 1.430 (0.011) | 1.484 (0.013) |
| | Row or Column Unseen | 4.427 (0.047) | 19.322 (0.025) | 8.845 (0.011) | 7.987 (0.011) |
| | Overall Perplexity | 4.424 (0.087) | 18.116 (0.035) | 8.843 (0.013) | 7.980 (0.021) |
| FE-DPCC | Row and Column Observed | 3.344 (0.021) | 17.125 (0.040) | 1.435 (0.024) | 1.489 (0.023) |
| | Row or Column Unseen | 3.892 (0.026) | 17.836 (0.053) | 1.453 (0.026) | 1.509 (0.024) |
| | Overall Perplexity | 3.889 (0.031) | 17.836 (0.062) | 1.450 (0.046) | 1.501 (0.045) |

approach where the assumption is that proteins in the same family have a similar activity/interaction profile. MP2 had 154 proteins, 2876 molecules and a total of 7146 positive interactions. Table 1 summarizes the dataset characteristics.

### 5.1 Experimental Methodology and Feature Information

**Table 1:** Training and Test Data

| | | MovieLens | Jester | MP1 | MP2 |
|---|---|---|---|---|---|
| Train | # Rows | 943 | 33459 | 1961 | 2674 |
| | # Columns | 1650 | 140 | 61 | 149 |
| | # Entries | 80000 | 80000 | 3000 | 5000 |
| | Density | 5.142% | 1.708% | 2.508% | 1.255% |
| Test | # Rows | 927 | 14523 | 856 | 1647 |
| | # Columns | 1407 | 139 | 68 | 145 |
| | # Entries | 20000 | 20000 | 1051 | 2146 |
| | Density | 1.533% | 0.991% | 1.806% | 0.899% |

We first compared FE-DPCC with a variant of NBCC, called *Dirichlet Process Co-clustering* (DPCC). DPCC restricts the Pitman-Yor priors of NBCC to the special case of independent Dirichlet Process priors on rows and columns, so as to compare with FE-DPCC fairly. So, the difference between FE-DPCC and DPCC is that FE-DPCC augments DPCC to exploit row and column features. We ran 1000 iterations of Gibbs sampling for both FE-DPCC and DPCC. We used perplexity as an evaluation metric to compare FE-DPCC with DPCC on all the test data. The perplexity of a dataset $D$ is defined as $perplexity(D) = \exp(-\mathcal{L}(D)/N)$, where $\mathcal{L}(D)$ is the log-likelihood of $D$, and $N$ is the number of data points in $D$. The higher the log-likelihood, the lower the perplexity, and the better a model fits the data.

The relational features in our data are discrete. We assume $f(\cdot|\boldsymbol{\theta}^{*E}_{kl})$ is a categorical distribution, $\text{Cat}(\cdot|\boldsymbol{\theta}^{*E}_{kl})$, and $g(\boldsymbol{\theta}^{*E}_{kl}|\zeta^E)$ is a Dirichlet distribution, $\text{Dir}(\boldsymbol{\theta}^{*E}_{kl}|\boldsymbol{\varphi})$, with $\zeta^E = \boldsymbol{\varphi}$. Because of conjugacy, we can marginalize out $\boldsymbol{\theta}^{*E}_{kl}$. Without loss of generality, we assume that $f(\cdot|\boldsymbol{\theta}^{*E}_{kl})$ is a $D$-dimensional categorical distribution with support $\{1, \cdots, D\}$, and we denote the Dirichlet hyperparameter as $\zeta^E = \boldsymbol{\varphi} = \langle \varphi_d | d = \{1, \cdots, D\} \rangle$. The predictive distribution of the co-cluster $(k, l)$ to observe a new entry $x^E_{r'c'} = d$, $d \in \{1, \cdots, D\}$, is:

$$p(x^E_{r'c'} = d|\zeta^{*E}_{kl}, z^R_{r'} = k, z^C_{c'} = l) = \int f(x^E_{r'c'} = d|\boldsymbol{\theta}^{*E}_{kl}) g(\boldsymbol{\theta}^{*E}_{kl}|\zeta^{*E}_{kl}) d\boldsymbol{\theta}^{*E}_{kl} =$$

$$\int \text{Cat}(x^E_{r'c'} = d|\boldsymbol{\theta}^{*E}_{kl}) \text{Dir}(\boldsymbol{\theta}^{*E}_{kl}|\boldsymbol{\varphi}^*_{kl}) d\boldsymbol{\theta}^{*E}_{kl} \propto \mathcal{N}^d_{(k,l)} + \varphi_d$$

where $\boldsymbol{\varphi}^*_{kl}$ is the posterior hyperparameter of the Dirichlet distribution of the co-cluster $(k, l)$, and $\mathcal{N}^d_{(k,l)}$ is the number of entries assigned to the co-cluster $(k, l)$ and is equal to $d$.

In MovieLens, users (rows) are represented with age, gender, and occupation, whereas the movies (columns) are associated with a 19-dimensional genre-representing binary vector. We assumed independence among the row features and the column features conditional on row- and column-clusters. We modeled age as drawn from a Poisson distribution, $\text{Poi}(\cdot|\lambda)$, with a conjugate

Gamma prior, $\text{Gamma}(\lambda|\varrho,\varsigma)$. We modeled gender as drawn from a Bernoulli distribution, $\text{Ber}(\cdot|\vartheta)$, with a conjugate Beta prior $\text{Beta}(\vartheta|\varkappa,\varpi)$. The occupation feature is categorical, modeled as $\text{Cat}(\cdot|\boldsymbol{\phi})$, with Dirichlet prior, $\text{Dir}(\boldsymbol{\phi}|\boldsymbol{\varphi})$. Thus, the row feature parameter is given by $\boldsymbol{\theta}_k^{*R} = \langle \lambda_k^*, \vartheta_k^*, \boldsymbol{\phi}_k^* \rangle$, and the row feature prior hyperparameter is $\zeta^R = \langle \varrho, \varsigma, \vartheta, \boldsymbol{\varphi} \rangle$. We denote the feature vector of a new user as $\boldsymbol{x}_{r'}^R = \langle a_{r'}, g_{r'}, o_{r'} \rangle$, where $a_{r'}$, $g_{r'}$, and $o_{r'}$ represent the age, gender and occupation, respectively. The predictive distribution of the $k^{th}$ row-cluster observing a new user, $\boldsymbol{x}_{r'}^R$, is:

$$p(\boldsymbol{x}_{r'}^R|\varrho_k^*,\varsigma_k^*,\varkappa_k^*,\varpi_k^*,\boldsymbol{\varphi}_k^*,z_{r'}^R=k) = \left(\int \text{Poi}(a_{r'}|\lambda_k^*)\text{Gamma}(\lambda_k^*|\varrho_k^*,\varsigma_k^*)d\lambda_k^*\right)$$
$$\left(\int \text{Ber}(g_{r'}|\vartheta_k^*)\text{Beta}(\vartheta_k^*|\varkappa_k^*,\varpi_k^*)d\vartheta_k^*\right)\left(\int \text{Cat}(o_{r'}|\boldsymbol{\phi}_k^*)\text{Dir}(\boldsymbol{\phi}_k^*|\boldsymbol{\varphi}_k^*)d\boldsymbol{\phi}_k^*\right) \quad (7)$$

where $\varrho_k^*$, $\varsigma_k^*$, $\varkappa_k^*$, $\varpi_k^*$, and $\boldsymbol{\varphi}_k^*$ are the posterior hyperparameters ($k$ indexes the row-clusters). Denote $\zeta_k^{*R} = \langle \varrho_k^*, \varsigma_k^*, \varkappa_k^*, \varpi_k^*, \boldsymbol{\varphi}_k^* \rangle$. We assume that features associated with movies are generated from a Multinomial distribution, $\text{Mul}(\cdot|\boldsymbol{\psi})$, with Dirichlet prior, $\text{Dir}(\boldsymbol{\psi}|\boldsymbol{\varphi})$. Accordingly, $\boldsymbol{\theta}_l^{*C} = \boldsymbol{\psi}_l^*$, and $\zeta^C = \boldsymbol{\varphi}$. The predictive distribution of the $l^{th}$ column-cluster observing a new movie, $\boldsymbol{x}_{c'}^C$, is: $p(\boldsymbol{x}_{c'}^C|\boldsymbol{\varphi}_l^*, z_{c'}^C = l) = \int \text{Mul}(\boldsymbol{x}_{c'}^C|\boldsymbol{\psi}_l^*)\text{Dir}(\boldsymbol{\psi}_l^*|\boldsymbol{\varphi}_l^*)d\boldsymbol{\psi}_l^*$, where $\zeta_l^{*C} = \boldsymbol{\varphi}_l^*$ is the posterior hyperparameter of the Dirichlet distribution ($l$ indexes the column-clusters).

In Jester, there are no features associated with the users (rows), thus row-clusters cannot predict an unseen user. We used a bag-of-word representation for joke features, and assumed each joke feature vector is generated from a Multinomial distribution, $\text{Mul}(\cdot|\boldsymbol{\psi})$, with a Dirichlet prior, $\text{Dir}(\boldsymbol{\psi}|\boldsymbol{\varphi})$. The predictive distribution of the $l^{th}$ column-cluster observing a new joke, $\boldsymbol{x}_{c'}^C$, is: $p(\boldsymbol{x}_{c'}^C|\boldsymbol{\varphi}_l^*, z_{c'}^C = l) = \int \text{Mul}(\boldsymbol{x}_{c'}^C|\boldsymbol{\psi}_l^*)\text{Dir}(\boldsymbol{\psi}_l^*|\boldsymbol{\varphi}_l^*)d\boldsymbol{\psi}_l^*$.

For MP1 and MP2, rows represent molecules and columns represent proteins. We extracted $k$-mer features from protein sequences. For MP1, we also used hierarchical features for proteins obtained from annotation databases. We used a graph-fragment-based feature representation that computes the frequency of different length cycles and paths for each molecule. These graph-fragment-based features were derived using AFGEN [34] (default parameters were used), known to capture structural aspects of molecules effectively. We assumed each protein was generated from a Multinomial distribution, $\text{Mul}(\cdot|\boldsymbol{\psi}^p)$, with a Dirichlet prior, $\text{Dir}(\boldsymbol{\psi}^p|\boldsymbol{\varphi}^p)$. We also assumed each molecule was generated from a Multinomial distribution, $\text{Mul}(\cdot|\boldsymbol{\psi}^m)$, with a Dirichlet prior, $\text{Dir}(\boldsymbol{\psi}^m|\boldsymbol{\varphi}^m)$. The predictive distribution of the $k^{th}$ row-cluster observing a new molecule, $\boldsymbol{x}_{r'}^R$, is: $p(\boldsymbol{x}_{r'}^R|\boldsymbol{\varphi}_k^{*m}, z_{r'}^R = k) = \int \text{Mul}(\boldsymbol{x}_{r'}^R|\boldsymbol{\psi}_k^{*m})\text{Dir}(\boldsymbol{\psi}_k^{*m}|\boldsymbol{\varphi}_k^{*m})d\boldsymbol{\psi}_k^{*m}$.

The predictive distribution of the $l^{th}$ column-cluster observing a new protein, $\boldsymbol{x}_{c'}^C$, is: $p(\boldsymbol{x}_{c'}^C|\boldsymbol{\varphi}_l^{*p}, z_{c'}^C = l) = \int \text{Mul}(\boldsymbol{x}_{c'}^C|\boldsymbol{\psi}_l^{*p})\text{Dir}(\boldsymbol{\psi}_l^{*p}|\boldsymbol{\varphi}_l^{*p})d\boldsymbol{\psi}_l^{*p}$.

### 5.2 Results

We performed a series of experiments to evaluate the performance of FE-DPCC across the four datasets. All experiments were repeated five times, and we

**Table 3:** Evaluation of feature enrichment on MP1.

| | *Perplexity* |
|---|---|
| Shuffle P | 3.034 (0.083) |
| Exchange M | 2.945 (0.083) |
| Exchange P | 2.932 (0.071) |
| Exchange M&P | 2.991 (0.095) |
| Use Only M | 7.235 (0.043) |
| Use Only P | 7.789 (0.045) |
| Use M and P | 1.450 (0.046) |

**Table 4:** Evaluation of protein features on MP1.

| | *Perplexity* |
|---|---|
| 2-mer | 1.471 (0.057) |
| 3-mer | 1.437 (0.044) |
| 4-mer | 1.441 (0.049) |
| 5-mer | 1.450 (0.046) |
| HF | 1.413 (0.010) |

**Table 5:** RMSE on Test Data.

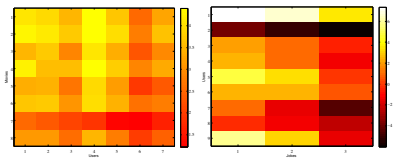
| | *FE-DPCC* | *Slope One* |
|---|---|---|
| Movie | 0.838 (0.031) | 0.924 (0.035) |
| Jester | 0.896 (0.062) | 0.961 (0.065) |

report the average (and standard deviation) perplexity across the five runs. The experiments were performed on an Intel four core, Linux server with 4GB memory. The average running time for FE-DPCC was 1, 3, 3.5 and 2.5 hours on the MovieLens, Jester, MP1 and MP2 datasets, respectively.

**Feature Enrichment Evaluation** Table 2 shows the average perplexity (and standard deviations) across five runs on the test data. To analyze the effect of new rows and columns on the prediction capabilities of the algorithms, we split each test set into subsets based on whether the subset contains new rows or columns w.r.t. the corresponding training data. Table 2 shows that the overall perplexity of FE-DPCC is lower than that of DPCC on all data, with an improvement of 12%, 1.5%, 84% and 81% for MovieLens, Jester, MP1 and MP2, respectively.

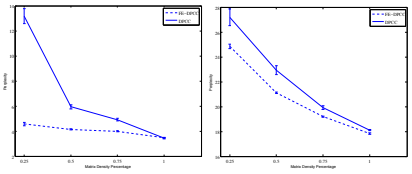
FE-DPCC is significantly better than DPCC on the portion of the test data that contains unseen rows and/or columns. These test sets consist of entries for rows and columns that are not included in the training set. The DPCC algorithm does not use features; as such it can predict entries for the new rows and columns using prior probabilities only. In contrast, the FE-DPCC algorithm leverages features along with prior probabilities; this enables our approach to predict values for the independent test entries more accurately. This ability is a major strength of our FE-DPCC algorithm. For the portion of the test data whose rows and columns are observed in the training as well, the perplexity values of FE-DPCC and DPCC are comparable. The standard deviations indicate that the algorithms are stable, yielding consistent results across different runs.

To accurately assess the performance of FE-DPCC, we performed a set of experiments that involved a perturbation of the protein and molecule features on MP1. Results are in Table 3. For these experiments, we used $k$-mer sequence features. First, we took the protein sequences (i.e., columns) and shuffled the ordering of the amino acids. This alters the ordering of the protein sequence but maintains the same composition (i.e., the shuffled sequences have the same number of characters or amino acids). We refer to this scheme as "Shuffle". It achieves an average perplexity of 3.034, versus the average perplexity of 1.450 achieved by FE-DPCC (with no shuffling of features). We also devised a scheme in which the row and/or column features are exchanged, e.g., the features of a particular molecule are exchanged with the features of another molecule. Such an exchange causes the inclusion of incorrect information within the FE-DPCC



(a) MovieLens (b) Jester

**Fig. 2:** Co-clusters Learned by FE-DPCC



(a) MovieLens (b) Jester

**Fig. 3:** Test Perplexity with Different Densities

algorithm. Our aim was to assess the strength of FE-DPCC when enriched with meaningful and correct features. We refer to this scheme as "Exchange." Table 3 shows the results of exchanging molecule features only (Exchange M), protein features only (Exchange P), and both (Exchange M and P). We noticed an average perplexity of 2.9 in each case. We also evaluated the FE-DPCC algorithm when only molecule or only protein features are used ("Use Only M" and "Use only P" in Table 3). The use of only one set of features prevents the co-clustering algorithm from making inferences on the unseen rows or columns in the test set.

For MP1 we performed additional experiments to evaluate the sequence features. The features are overlapping subsequences of a fixed length extracted from the protein sequences. We used $k$-mer lengths of 2, 3, 4 and 5, and observed that the average perplexity (Table 4) remained similar. As such, we used 5-mer features in all the experiments. We also compared the sequence features for the proteins to an alternate feature derived from a hierarchical biological annotation of the proteins. For MP1 the hierarchical features were extracted as done in the previous study [13, 22]. From Table 4 we observe that the hierarchical features (HF) achieved a slightly lower perplexity as compared to the 5-mer features. This is encouraging, as it suggests that sequence features perform similarly to manual annotation (hierarchy), that may not be easily available for all the proteins.

**Comparative Performance** We compared FE-DPCC with a well known collaborative filtering model, *Slope One* [16]. We used a Slope One implementation from the Apache Mahout machine learning library[6]. We used the root mean square error (RMSE) [6] to compare FE-DPCC and Slope One on MovieLens and Jester. Table 5 shows the RMSE values (and standard deviations) of FE-DPCC and Slope One across five runs on the test sets[7]. These results show that incorporating row and column features is beneficial for the prediction of relationships.

**Visualization of Co-clusters** In Figure 2 we illustrate the co-cluster structures learned by FE-DPCC on MovieLens and Jester. We calculate the mean entry value for each co-cluster, and plot the resulting mean values.

**Data Density** We varied the density of MovieLens and Jester to see how it affects the perplexity of FE-DPCC and DPCC. We varied the matrix density by randomly sampling 25%, 50% and 75% of the entries in the training data. The sampled matrices were then given as input to DPCC and FE-DPCC to train

[6] http://mahout.apache.org/

[7] No new rows or columns in the test sets w.r.t. the training sets.

a model and infer unknown entries on the test data. Figure 3 illustrates the results averaged across five iterations. As the sparsity of the relational matrix increases the test perplexity increases for both FE-DPCC and DPCC. But DPCC has far higher perplexity for a sparser matrix. As the matrix sparsity increases, the information within the relational matrix is lost and the FE-DPCC algorithm relies on the row and column features. Thus, for sparser matrices FE-DPCC shows far better results than DPCC. These experiments suggest the reason why we see a more dramatic difference between the two algorithms for MP1 and MP2, which are very sparse (see Table 1).

## 6 Conclusion

In this work, we focus on the empirical evaluation of FE-DPCC to predict relationships between previously unseen objects by using object features. We conducted experiments on a variety of relational data, including protein-molecule interaction data. The evaluation demonstrates the effectiveness of the feature-enriched approach and demonstrates that features are most useful when data are sparse.

## Acknowledgements

This work was in part supported by NSF III-0905117.

## References

1. D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, pages 19–28, 2009.
2. C. E. Antoniak. Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.
3. M. Balabanovic and Y. Shoham. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, 1997.
4. D. Blackwell and J. B. Macqueen. Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.
5. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, 2003.
6. Y.-H. Chen and E. I. George. A bayesian model for collaborative filtering. In *7th International Workshop on Artificial Intelligence and Statistics*, 1999.
7. I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, pages 89–98, 2003.
8. D. B. Dunson, Y. Xue, and L. Carin. The matrix stick-breaking process: Flexible Bayes meta-analysis. *Journal of the American Statistical Association*, 103(481):317–327, 2008.
9. T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.
10. T. George and S. Merugu. A scalable collaborative filtering framework based on co-clustering. In *Proceedings of the IEEE International Conference on Data Mining*, pages 625–628, 2005.
11. J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.
12. T. Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22:89–115, January 2004.

13. L. Jacob, B. Hoffmann, V. Stoven, and J.-P. Vert. Virtual screening of GPCRs: an in silico chemogenomics approach. *BMC Bioinformatics*, 9(1):363, 2008.
14. R. Jin, L. Si, and C. Zhai. A study of mixture models for collaborative filtering. *Journal of Information Retrieval*, 9:357–382, 2006.
15. M. Khoshneshin and W. N. Street. Incremental collaborative filtering via evolutionary co-clustering. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 325–328, New York, NY, USA, 2010. ACM.
16. D. Lemire and A. Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of the SIAM Data Mining (SDM)*, 2005.
17. Y. J. Lim and Y. W. Teh. Variational Bayesian Approach to Movie Rating Prediction. In *Proceedings of KDD Cup and Workshop*, 2007.
18. B. Marlin. Modeling user rating profiles for collaborative filtering. In *Advances in Neural Information Processing Systems (NIPS)*, volume 17, 2003.
19. E. Meeds and S. Roweis. Nonparametric Bayesian Biclustering. Technical Report UTML TR 2007-001, Department of Computer Science, University of Toronto, 2007.
20. R. M. Neal. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.
21. X. Ning, H. Rangwala, and G. Karypis. Multi-assay-based structure activity relationship models: Improving structure activity relationship models by incorporating activity information from related targets. *Journal of Chemical Information and Modeling*, 49(11):2444–2456, 2009. PMID: 19842624.
22. Y. Okuno, J. Yang, K. Taneishi, H. Yabuuchi, and G. Tsujimoto. GLIDA: GPCR-ligand database for chemical genomic drug discovery. *Nucleic Acids Research*, 34(suppl1):D673–D677, 2006.
23. O. Papaspiliopoulos and G. O. Roberts. Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika*, 95(1):169–186, March 2008.
24. J. Pitman and M. Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25(2):855–900, 1997.
25. I. Porteous, A. Asuncion, and M. Welling. Bayesian matrix factorization with side information and dirichlet process mixtures. In *AAAI*, 2010.
26. R. Salakhyuditnov and A. Mnih. Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo. In *International Conference on Machine Learning*, 2008.
27. J. B. Schafer, J. Konstan, and J. Riedi. Recommender systems in e-commerce. In *Proceedings of the ACM cConference on Electronic Commerce*, pages 158–166, 1999.
28. J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.
29. M. Shafiei and E. Milios. Latent Dirichlet co-clustering. In *IEEE International Conference on Data Mining*, pages 542–551, 2006.
30. H. Shan and A. Banerjee. Bayesian co-clustering. In *IEEE International Conference on Data Mining*, 2008.
31. H. Shan and A. Banerjee. Generalized probabilistic matrix factorizations for collaborative filtering. In *Proceedings of the IEEE International Conference on Data Mining*, pages 1025–1030, 2010.
32. I. Sutskever, R. Salakhutdinov, and J. Tenenbaum. Modelling relational data using Bayesian clustered tensor factorization. In *Advances in Neural Information Processing Systems 22*, pages 1821–1828, 2009.
33. P. Symeonidis, A. Nanopoulos, A. Papadopoulos, and Y. Manolopoulos. Nearest-Biclusters Collaborative Filtering. In *WEBKDD-06*, 2006.
34. N. Wale and G. Karypis. AFGEN. Technical report, Department of Computer Science & Enigneering, University of Minnesota, 2007. www.cs.umn.edu/∼karypis.
35. P. Wang, C. Domeniconi, and K. Laskey. Latent Dirichlet Bayesian co-clustering. In *Proceedings of the European Conference on Machine Learning*, pages 522–537, 2009.
36. Z. Xu, V. Tresp, K. Yu, and H. Kriegel. Infinite hidden relational models. In *Proceedings of the International Conference on Uncertainity in Artificial Intelligence*, 2006.