Nonparametric Bayesian Co-clustering Ensembles

Pu Wang* Kathryn B. Laskey† Carlotta Domeniconi* Michael I. Jordan‡

Abstract

A nonparametric Bayesian approach to co-clustering ensembles is presented. Similar to clustering ensembles, co-clustering ensembles combine various base co-clustering results to obtain a more robust consensus co-clustering. To avoid pre-specifying the number of co-clusters, we specify independent Dirichlet process priors for the row and column clusters. Thus, the numbers of row- and column-clusters are unbounded *a priori*; the actual numbers of clusters can be learned *a posteriori* from observations. Next, to model non-independence of row- and column-clusters, we employ a Mondrian Process as a prior distribution over partitions of the data matrix. As a result, the co-clusters are not restricted to a regular grid partition, but form nested partitions with varying resolutions. The empirical evaluation demonstrates the effectiveness of nonparametric Bayesian co-clustering ensembles and their advantages over traditional co-clustering methods.

1 Introduction.

Ensemble methods have been a major success story in machine learning and data mining, particularly in classification and regression problems. Recent work has also focused on clustering, where ensembles can yield robust consensus clusterings [19, 20, 7, 6, 11]. In this paper we contribute to this line of research by studying the application of ensembles to *co-clustering*, the problem of simultaneously clustering the rows and columns of a data matrix into row- and column-clusters to achieve homogeneity in the blocks in the induced partition of the data matrix.

Our approach to co-clustering ensembles is a nonparametric Bayesian approach based on the Dirichlet process and the Mondrian process. While nonparametric Bayesian methods have previously been used in co-clustering [12], to allow the number of row clusters and column clusters to be random and inferred from the data, our work makes use of nonparametric Bayesian ideas to model co-clustering *ensembles*. In particular, we develop a model-based approach to ensembles that explicitly models the way in which multiple co-clusterings differ from each other and from a consensus co-clustering.

One way in which multiple co-clusterings can arise is via different local optima of a single base co-clustering method. Rather than selecting one of these optima, our approach explicitly recognizes the possibility that these local optima may contribute distinct, complementary perspectives on the co-clustering problem, in which case all optima should contribute to the formation of a consensus co-clustering. It is worth noting that this issue arises in many problems in which there is combinatorial structure, and our model-based approach to ensembles may have applications beyond co-clustering.

Most co-clustering algorithms [4, 17, 18, 21] assume that row- and column-clusters are variation independent; i.e., individual co-clusters are obtained as the product of row- and column-clusters. This partitions the data matrix into a regular grid. This assumption of variation independence is inappropriate in situations exhibiting context-specific independence (for example, one cannot represent the situation in which, for some rows, a given set of columns is partitioned into several clusters, whereas for other rows, the columns form a single undifferentiated cluster). Recent work has explored a nonparametric prior known as the *Mondrian processes* that relaxes this assumption [15]. A sample drawn from a two-dimensional Mondrian process is a random partition over a matrix that is not constrained to be a regular grid. In this paper we explore ensemble versions of both kinds of base co-clustering method. Specifically we develop (1) a Dirichlet process-based co-clustering ensemble model (DPCCE), which assumes independent Dirichlet process mixture priors for rows and columns; and (2) a Mondrian process-based co-clustering ensemble model (MPCCE) that places a Mondrian process prior over the matrix partitions. For both the DPCCE and the MPCEE, the number of blocks is not fixed *a priori*, but is open-ended and inferred from the data.

This paper is organized as follows. We review related work in Section 2 and introduce some necessary background in Section 3. We then propose two new nonparametric Bayesian co-clustering ensemble models

*Department of Computer Science, George Mason University, {pwang7, cdomenic}@gmu.edu

†Department of System Engineering and Operation Research, George Mason University, klaskey@gmu.edu

‡Computer Science Division and Department of Statistics, University of California, Berkeley, jordan@eecs.berkeley.edu

in Sections 4 and 5. Experimental results are presented in Section 6, followed by our conclusions in Section 7.

2 Related Work.

Co-clustering is an active area of research. Dhillon et al. [4] introduced an information-theoretic co-clustering approach based on hard partitions. Shafiei et al. [17] proposed a soft-partition co-clustering method called "Latent Dirichlet Co-clustering." This model, however, does not cluster rows and columns simultaneously. A Bayesian Co-Clustering (BCC) model has been proposed in [18]. BCC 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) [3] model, BCC assumes symmetric Dirichlet priors for the data distributions given the row- and column-clusters. Shan and Banerjee [18] proposed a variational Bayesian algorithm to perform inference. In [21] the authors developed a collapsed Gibbs sampling and a collapsed variational Bayesian algorithm to perform inference.

While clustering ensembles have been explored by researchers to provide robust solutions to the problem of clustering [19, 20, 7, 6, 11], co-clustering ensembles have received little attention. An exception is the projective clustering method in [10], where the authors formulate co-clustering ensembles as an optimization problem which involves both data and feature clustering.

3 Background.

3.1 Dirichlet Process. The Dirichlet process (DP) [5] is an infinite-dimensional generalization of the Dirichlet distribution. Formally, let S be a set, G_0 a measure on S, and α_0 a positive real number. The random probability distribution G on S is distributed as a DP with concentration parameter α_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 [5]. 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 [2]:

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

The DP is often used as a nonparametric prior in Bayesian mixture models [1]. Assume the data are generated from the following generative procedure:

$$\begin{aligned} G &\sim \text{Dir}(\alpha_0, G_0) \\ \theta_{1:N} &\sim G \\ x_{1:N} &\sim \prod_{n=1}^{N} F(\cdot|\theta_n), \end{aligned}$$

where the $F(\cdot|\theta_n)$ are probability distributions known as mixture components. There typically 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 [13]. Therefore, DPM is known as an "infinite" mixture model.

The DP can be generated via the stick-breaking construction [16]. Stick-breaking draws two infinite sequences of independent random variables, $v_k \sim Beta(1, \alpha_0)$ and $\theta^*_k \sim G_0$ for $k = \{1, 2, \cdots\}$. Let G be defined as:

$$\pi_k = v_k \prod_{j=1}^{k-1}(1 - v_j) \tag{3.1}$$

$$G = \sum_{k=1}^{\infty} \pi_k \delta(\theta^*_k) \tag{3.2}$$

where $\vec{\pi} = \langle \pi_k | k = 1, 2, \cdots \rangle$ are mixing proportions and $\delta(\theta)$ is the distribution that samples the value θ 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 DPM model using the stick-breaking construction is:

1. Draw $v_k \sim \text{Beta}(1, \alpha_0)$, $k = \{1, 2, \cdots\}$ and calculate $\vec{\pi}$ as in Eq (3.1).
2. Draw $\theta^*_k \sim G_0$, $k = \{1, 2, \cdots\}$
3. For each data point $n = \{1, 2, \cdots, N\}$:
   - Draw $z_n \sim \text{Discrete}(\vec{\pi})$
   - Draw $x_n \sim F(\cdot|\theta^*_{z_n})$

The most popular inference method for DPM is MCMC [13]. Here we briefly introduce Gibbs sampling for DPM when $F(\cdot|\theta^*_{z_n})$ and G_0 are conjugate. Conditional on observations $\{x_n\}_{n\in\{1,\cdots,N\}}$ sampled from G and values $\{z_n\}_{n\in\{1,\cdots,N\}}$ for the indicator variables, the posterior density function for the parameter θ^*_k for the k^{th} cluster is also a member of the conjugate family:

$$p(\theta^*_k|\{x_n,z_n\}_{n\in\{1,\cdots,N\}}) = g(\theta^*_k|\zeta^*_k) = \frac{\prod_{n=1}^N f(x_n|\theta^*_k)^{1_{[z_n=k]}} g(\theta^*_k|\zeta_0)}{\int \prod_{n=1}^N f(x_n|\theta^*_k)^{1_{[z_n=k]}} g(\theta^*_k|\zeta_0) d\theta^*_k} \tag{3.3}$$

where $1_{[.]}$ is the indicator function, $f(x|\theta)$ is the density (or mass) function for $F(\cdot|\theta)$, $g(\theta|\zeta_0)$ is the density function for G_0, and $g(\theta^*_k|\zeta^*_k)$ is the posterior density function, with parameter ζ^*_k obtained using the conjugate updating rule. Conditional on the next indicator variable z_{N-1}, the predictive distribution for the next data point is given by:

$$p(x_N|\{x_n,z_n\}_{n\in\{1,\cdots,N\}}, z_{N+1}=k) \int f(x_N|\theta^*_k) g(\theta^*_k|\zeta^*_k) d\theta^*_k, \tag{3.4}$$

can also be obtained in closed form. Having integrated out the parameters, it is necessary to Gibbs sample only the indicator variables. The conditional probability for sampling the indicator variable for the i^{th} data point is given as follows. For populated clusters $k \in \{z_n\}_{n\in\{1,\cdots,i-1,i+1,\cdots,N\}}$,

$$p(z_i=k|x_i,\{x_n,z_n\}_{n\in\{1,\cdots,i-1,i+1,\cdots,N\}}) \propto \frac{n_k^{\neg i}}{N-1+\alpha_0} \int f(x_i|\theta^*_k) g(\theta^*_k|\zeta^{*\neg i}_k) d\theta^*_k. \tag{3.5}$$

Here, $n_k^{\neg i}$ is the number of data points other than x_i assigned to the k^{th} cluster, and $g(\theta^*_k|\zeta^{*\neg i}_k)$ is the posterior density for the k^{th} cluster parameter given all observations except x_i. If $z_i \notin \{z_n\}_{n\in\{1,\cdots,i-1,i+1,\cdots,N\}}$ is a singleton cluster and $k = z_i$, or if $z_i \in \{z_n\}_{n\in\{1,\cdots,i-1,i+1,\cdots,N\}}$ is not a singleton cluster and $k = N+1$, the predictive probability is:

$$p(z_i=k|x_i,\{x_n,z_n\}_{n\in\{1,\cdots,i-1,i+1,\cdots,N\}}) \propto \frac{\alpha_0}{N-1+\alpha_0} \int f(x_i|\theta^*_k) g(\theta^*_k|\zeta_0) d\theta^*_k. \tag{3.6}$$

Eq (3.5) is the probability of assigning x_i to the k^{th} existing cluster, while Eq (3.6) is the probability of assigning x_i to its own singleton cluster.

Additional details on DPM inference can be found in [13, 14].

3.2 Mondrian Process. A Mondrian process $\mathcal{M} \sim MP(\lambda,(a,A),(b,B))$ on a 2-dimensional rectangle $(a,A)\times(b,B)$ generates random partitions of a rectangle as follows [15]: The parameter λ, called the *budget*, controls the overall number of cuts in the partition. At each stage, a random cost E is drawn and compared to the budget. If E exceeds the budget, the process halts with no cuts; otherwise, a cut is made at random, the cost is subtracted from the budget, and the process recurses on the two sub-rectangles, each being drawn independently from its own MP distribution.

The cost E of cutting the rectangle $(a,A)\times(b,B)$ is distributed exponentially with mean equal to $1/(A-a+B-b)$, the inverse of the combined length of the sides. That is, for fixed λ, a longer perimeter tends to result in a lower cost. The parameter λ can be viewed as a rate of cut generation per unit length of perimeter. If a cut is made, it has horizontal or vertical direction with probability proportional to the lengths of the respective sides, and its placement is uniformly distributed along the chosen side. After a cut is made, a new budget $\lambda' = \lambda - E$ is calculated, and the sub-rectangles are independently partitioned according to a Mondrian process with rate λ'. That is, if the cut splits the horizontal side into (a,x) and (x,A), then the two sub-rectangle processes are $\mathcal{M}_< \sim MP(\lambda',(a,x),(b,B))$ and $\mathcal{M}_> \sim MP(\lambda',(x,A),(b,B))$, respectively. Conversely, for a vertical cut into (b,x) and (x,B), the sub-rectangle processes are $\mathcal{M}_< \sim MP(\lambda',(a,A),(b,x))$ and $\mathcal{M}_> \sim MP(\lambda',(a,A),(x,B))$.

The one-dimensional Mondrian process reduces to a Poisson process. The MP shares with the Poisson process the self-consistency property that its restriction to a subspace is a Mondrian process with the same rate parameter as the original Mondrian process. As with the Poisson process, one can define a non-homogeneous MP by sampling the cuts non-uniformly according to a measure defined along the sides of the rectangle [15]. Here, we consider only the homogeneous MP.

Algorithm 1 samples a Mondrian process $\mathcal{M}$ with rate λ on a 2-dimensional space $(a,A)\times(b,B)$. Additional details on the Mondrian Process can be found in [15].

4 Dirichlet Process-based Co-clustering Ensembles.

4.1 DPCCE Generative Model. Following general practice in the clustering ensemble literature, [19], the DPCCE model does not specify a probabilistic model for the original $R\times C$ data matrix $\tilde{X}$, but rather models the output of M base co-clusterings $\langle\varphi_m|m\in\{1,2,\cdots,M\}\rangle$. The base co-cluster φ_m partitions the rows and columns of the data matrix into I_m row clus-

Algorithm 1 Mondrian $\mathcal{M} \sim MP(\lambda, (a, A), (b, B))$

```
let λ' ← λ − E where E ~ Exp(A − a + B − b)
if λ' < 0 then
  return M ← {(a, A) × (b, B)}
end if
draw ρ ~ Bernoulli((A−a)/(A−a+B−b))
if ρ = 1 then
  draw x ~ Uniform(a, A)
  let M1 ← MP(λ', (a, x), (b, B))
  let M2 ← MP(λ', (x, A), (b, B))
  return M ← M1 ∪ M2
else
  draw x ~ Uniform(b, B)
  let M1 ← MP(λ', (a, A), (b, x))
  let M2 ← MP(λ', (a, A), (x, B))
  return M ← M1 ∪ M2
end if
```

ters and J_m column clusters. We assume that rows and columns are clustered independently by the base clusterings, resulting in a grid-style partition. That is, all entries in a given row (column) are assigned to the same row (column) cluster. The base co-clusterings are organized into a $R \times C \times M$ array $\vec{Y}$, where the entries $y_{rcm} = \langle y^R_{rm}, y^C_{cm} \rangle$ denote the row- and column-cluster ID's assigned by φ_m. The indices y^R_{rm} and y^C_{cm} range from 1 to I_m and J_m, respectively.

According to the DPCCE model, the observations $\vec{Y}$ are generated from independent row and column Dirichlet process mixture models with pseudo-counts α^R and α^C, and row and column base measures G^R_m and G^C_m, respectively. Figure 1 depicts the DPCCE model. A stick-breaking process is used to generate the row and column Dirichlet processes. The mixing proportions $\vec{\pi}^R$ and $\vec{\pi}^C$ are generated as in Eq (3.1), and the consensus cluster indicator variables z^R_r and z^C_c are drawn according to these mixing proportions. The unique row and column parameters $\vec{\theta}^{*R}_{lm}$ and $\vec{\theta}^{*C}_{km}$ for each consensus row-cluster l and column-cluster k are generated as independent draws from symmetric T-dimensional Dirichlet distributions G^R_m and G^C_m with pseudo-counts β^R_m and β^C_m, respectively. We assume $I_m, J_m \leq T$; as T grows without bound with fixed total pseudo-count, G^R_m and G^C_m become Dirichlet process distributions. The row-cluster ID's y^R_{rm} are independent draws from a T-dimensional discrete distribution with parameter $\vec{\theta}^{*R}_{lm}$, where $l = z^R_r$ is the row-cluster indicator for row r. Similarly, the column-cluster ID's y^C_{cm} are independent draws from a T-dimensional discrete distribution with parameter $\vec{\theta}^{*C}_{km}$, where $k = z^C_c$ is the column-cluster indicator for row r.

Formally, the generative process for DPCCE is:

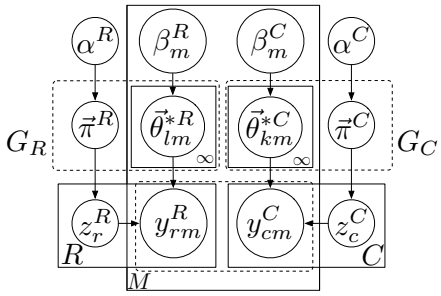

Figure 1: The DPCCE models.

- Draw $v^R_l \sim \text{Beta}(1, \alpha^R)$, for $l = 1, 2, \cdots, \infty$
- Set mixture weights for consensus row-clusters $\pi^R_l = v^R_l \prod_{t=1}^{l-1}(1 - v^R_t)$, for $l = 1, 2, \cdots, \infty$
- Draw $v^C_k \sim \text{Beta}(1, \alpha^C)$, for $k = 1, 2, \cdots, \infty$
- Set mixture weights for consensus column-clusters $\pi^C_k = v^C_k \prod_{t=1}^{k-1}(1 - v^C_t)$, for $k = 1, 2, \cdots, \infty$
- Draw parameters for consensus row-clusters $\vec{\theta}^{*R}_l \sim \text{Dir}(\beta^R)$, for $l = 1, 2, \cdots, \infty$
- Draw parameters for consensus column-clusters $\vec{\theta}^{*C}_k \sim \text{Dir}(\beta^C)$, for $k = 1, 2, \cdots, \infty$
- For each row r:
  - Draw consensus row-cluster $z^R_r \sim \text{Discrete}(\vec{\pi}^R)$
  - For each base co-clustering φ_m:
    - Generate $y^R_{rm} \sim \text{Discrete}(\vec{\theta}^{*R}_{lm})$, where $l = z^R_r$
- For each column c:
  - Draw consensus column-cluster $z^C_c \sim \text{Discrete}(\vec{\pi}^C)$
  - For each base co-clustering φ_m:
    - Generate $y^C_{cm} \sim \text{Discrete}(\vec{\theta}^{*C}_{km})$, where $k = z^C_c$

4.2 DPCCE Inference. We use the collapsed Gibbs sampling method discussed in Sec. 3.1 for DPCCE inference. As all model parameters are marginalized out, we sample only z^R_r and z^C_c. We assume infinite T, so that G^R_m and G^C_m become Dirichlet process distributions.

The conditional distribution for sampling z^R_r given $\vec{Y}$ and all other indicator variables $\vec{z}^{R\neg r}$ is:

$$p(z^R_r = l | \vec{Y}, \vec{z}^{R\neg r}, \gamma^R) \propto \frac{\mathcal{N}^{R\neg r}_l}{R - 1 + \alpha^R} \prod_{m=1}^{M} \mathcal{N}^{R\neg r}_{y^R_{rm}} \tag{4.7}$$

when the cluster index l appears among the indices in $\vec{z}^{R\neg r}$, and

$$p(z_r^R = l|\vec{Y}, \vec{z}^{R\neg r}, \gamma^R) \propto \frac{\alpha^R}{R-1+\alpha^R} \prod_{m=1}^{M} \mathcal{N}_{y_{rm}^R}^{R^{\neg r}} \tag{4.8}$$

when the cluster index l does not appear among the indices in $\vec{z}^{R\neg r}$. Here, $\mathcal{N}_l^{R^{\neg r}}$ is the number of rows assigned to the l^{th} consensus row-cluster excluding the r^{th} row, and $\mathcal{N}_{y_{rm}^R}^{R^{\neg r}}$ is the number rows assigned to the same row-cluster as the r^{th} row by φ_m excluding the r^{th} row.

Similarly, the conditional distribution for sampling z_c^C given $\vec{Y}$ and all other indicator variables $\vec{z}^{C\neg c}$ is:

$$p(z_c^C = k|\vec{Y}, \vec{z}^{C\neg c}, \gamma^C) \propto \frac{\mathcal{N}_k^{C^{\neg c}}}{C-1+\alpha^C} \prod_{m=1}^{M} \mathcal{N}_{y_{cm}^C}^{C^{\neg c}} \tag{4.9}$$

when the cluster index k appears among the indices in $\vec{z}^{C\neg c}$, and

$$p(z_c^C = k|\vec{Y}, \vec{z}^{C\neg c}, \gamma^C) \propto \frac{\alpha^C}{C-1+\alpha^C} \prod_{m=1}^{M} \mathcal{N}_{y_{cm}^C}^{C^{\neg c}} \tag{4.10}$$

when the cluster index k does not appear among the indices in $\vec{z}^{C\neg c}$. Here, $\mathcal{N}_k^{C^{\neg c}}$ is the number of columns assigned to the k^{th} consensus column-cluster excluding the c^{th} column, and $\mathcal{N}_{y_{cm}^C}^{C^{\neg c}}$ is the number columns assigned to the same column-cluster as the c^{th} column by φ_m excluding the c^{th} column.

Table 1 summarizes notation used throughout the paper.

5 Mondrian Process-based Co-clustering Ensembles.

5.1 MPCCE Generative Model. The Mondrian Process-based Co-clustering Ensemble (MPCCE) model generalizes the grid-style partitions of the DPCCE to allow different resolutions in different parts of the data matrix. The non-regular partitions generated by the MP provide increased flexibility and parsimony.

A sample drawn from a two-dimensional Mondrian Process partitions a rectangle using axis-aligned cuts, as illustrated in Figure 2 (left). If we overlay this partition on a data matrix, we can identify each block with a co-cluster consisting of entries falling inside the block. The model replaces the independent row clusters and column clusters of the DPCCE model with a set of co-clusters. It is more natural to deal with these co-clusters directly, rather than with row- and column-clusters separately. To achieve the same level of resolution with a grid-style partition would require a much less parsimonious model, as shown in Figure 2 (right).

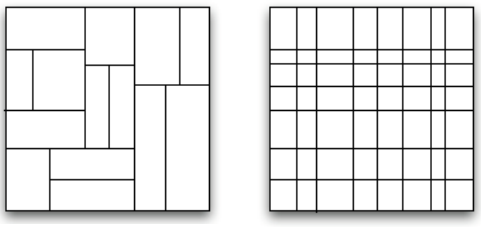

Figure 2: Unpermuted Synthetic Data Matrix Sampled from Mondrian Process (left) and Corresponding Grid (right)

The MPCCE generative process, depicted in Figure 3, puts a two-dimensional MP prior on partitions of the data matrix. Following [15], we treat a MP prior as generating a partition $\mathcal{M}$ over the unit square $[0,1] \times [0,1]$. Rows and columns of the data matrix are mapped to vertical and horizontal coordinates of the unit square through latent variables ξ_r and η_c. The latent variables $\vec{\xi} = \langle \xi_r | r \in \{1, \cdots, R\} \rangle$ and $\vec{\eta} = \langle \eta_c | c \in \{1, \cdots, C\} \rangle$ act like permutations of the rows and columns of the data matrix. The partition $\mathcal{M}$ and the latent variables $\vec{\xi}$ and $\vec{\eta}$ determine a partition over the original data matrix.

As with DPCCE and standard practice in the clustering ensemble literature and model the variables y_{rcm} that denote the co-cluster ID assigned to the entry in row r and column c by the m^{th} base clustering φ_m. The co-cluster ID y_{rcm} ranges from 1 to J_m, the number of co-clusters output by φ_m. We assume that y_{rcm} is sampled from a discrete distribution with parameter $\vec{\theta}_{mk}$, namely $p(y_{rcm} = j_m) = \theta_{mkj_m}$, where k is the block of $\mathcal{M}$ corresponding to row r and column c, and the parameter $\vec{\theta}_{mk}$ is sampled from a symmetric J_m-dimensional Dirichlet distribution.

Formally, the generative process for the base clusterings $\vec{Y}$ proceeds as follows:

- Draw a partition $\mathcal{M} \sim MP(\lambda, [0,1], [0,1])$; let K be the number of blocks in $\mathcal{M}$
- Draw block parameters $\vec{\theta}_{mk} \sim \text{Dir}(\beta_m)$, for $m = 1, 2, \cdots, M$ and $k = 1, 2, \cdots, K$
- Draw latent row coordinates $\xi_r \sim \text{Uniform}[0,1]$, for $r = 1, 2, \cdots, R$
- Draw latent column coordinates $\eta_c \sim \text{Uniform}[0,1]$, for $c = 1, 2, \cdots, C$
- For each row r and column c:

Table 1: Notation Description

Symbols	Description
R	number of rows in the data matrix $\vec{X}$
C	number of columns in the data matrix $\vec{X}$
M	number of base co-clusterings
φ_m	the m^{th} base co-clustering
	Notation for DPCCE
I_m	number of row-clusters in φ_m
J_m	number of column-clusters in φ_m
y^R_{rm}	the row-cluster assigned to the r^{th} row by φ_m, $y^R_{rm} \in \{1, \cdots, I_m\}$
y^C_{cm}	the column-cluster assigned to the c^{th} column by φ_m, $y^C_{cm} \in \{1, \cdots, J_m\}$
$\vec{Y}$	defined as $\langle y_{rcm} \vert r \in \{1, \cdots, R\}, c \in \{1, \cdots, C\}, m \in \{1, \cdots, M\}\rangle$
$\vec{\theta}^{R*}_{lm}$	the discrete distribution of observing the row-clusters of φ_m in the l^{th} consensus row-cluster
$\vec{\theta}^{C*}_{km}$	the discrete distribution of observing the column-clusters of φ_m in the k^{th} consensus column-cluster
$\vec{\theta}^{C*}_{l}$	defined as $\langle \theta^{R*}_{lm} \vert m \in \{1, 2, \cdots, M\}\rangle$
$\vec{\theta}^{C*}_{k}$	defined as $\langle \theta^{C*}_{km} \vert m \in \{1, 2, \cdots, M\}\rangle$
$\mathcal{N}^R_{i_m}$	the number of rows assigned to the ${i_m}^{th}$ row-cluster by φ_m
$\mathcal{N}^C_{j_m}$	the number of columns assigned to the ${j_m}^{th}$ column-cluster by φ_m
$\mathcal{N}^R_{l}$	the number of rows assigned to the l^{th} consensus row-cluster
$\mathcal{N}^C_{k}$	the number of columns assigned to the k^{th} consensus column-cluster
$\mathcal{N}^{R^{\neg r}}_{l}$	the number of rows assigned to the l^{th} consensus row-cluster excluding the r^{th} row
$\mathcal{N}^{C^{\neg c}}_{k}$	the number of columns assigned to the k^{th} consensus column-cluster excluding the c^{th} column
$\mathcal{N}^{R^{\neg r}}_{y^R_{r\cdot m}}$	the number rows assigned to the same row-cluster as the r^{th} row by φ_m excluding the r^{th} row
$\mathcal{N}^{C^{\neg c}}_{y^C_{\cdot cm}}$	the number of columns assigned to the same column-cluster of the c^{th} column by φ_m, excluding the c^{th} column
	Notation for MPCCE
J_m	number of co-clusters in φ_m
$\mathcal{M}$	a Mondrian sample, which is a Mondrian style partition over the unit square, and assume there are K blocks in $\mathcal{M}$
y_{rcm}	the co-cluster identity assigned to the entry (r, c) by the m^{th} base clustering φ_m, $y_{rcm} \in \{1, \cdots, K\}$
$\vec{Y}$	defined as $\langle y_{rcm} \vert r \in \{1, \cdots, R\}, c \in \{1, \cdots, C\}, m \in \{1, \cdots, M\}\rangle$
θ_{mkj_m}	the probability of assigning an entry in the k^{th} block of $\mathcal{M}$ by φ_m to its ${j_m}^{th}$ co-cluster
$\vec{\theta}_{mk}$	defined as $\langle \theta_{mkj_m} \vert j_m \in \{1, 2, \cdots, J_m\}\rangle$, which is drawn from a J_m-dimensional symmetric Dirichlet distribution with hyperparameter β_m
χ^R_h	the position of the h^{th} horizontal cut of the total L_R horizontal cuts in $\mathcal{M}$
χ^C_g	the position of the g^{th} vertical cut of the total L_C vertical cuts in $\mathcal{M}$
$\mathcal{N}_k$	the number of entries in the k^{th} block of $\mathcal{M}$
$\mathcal{N}^{y_{\cdot\cdot m}=j_m}_k$	the number of entries in both the k^{th} block of $\mathcal{M}$ and the ${j_m}^{th}$ co-cluster of φ_m
$\mathcal{N}^{\neg r}_k$	the number of entries in the k^{th} block of $\mathcal{M}$, excluding the entries in the r^{th} row
$\mathcal{N}^{\neg c}_k$	the number of entries in the k^{th} block of $\mathcal{M}$, excluding the entries in the c^{th} column
$\mathcal{N}^{\neg r}_{k,y_{\cdot\cdot m}=j_m}$	the number of entries in both the k^{th} block of $\mathcal{M}$ and the ${j_m}^{th}$ co-cluster of φ_m, excluding the entries in the r^{th} row
$\mathcal{N}^{\neg c}_{k,y_{\cdot\cdot m}=j_m}$	the number of entries in both the k^{th} block of $\mathcal{M}$ and the ${j_m}^{th}$ co-cluster of φ_m, excluding the entries in the c^{th} column

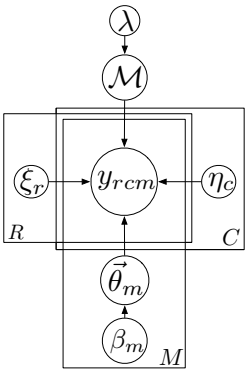

Figure 3: The Mondrian Process-based Co-Clustering Ensemble Model

- Let k be the block (co-cluster) of $\mathcal{M}$ to which (ξ_r, η_c) belongs
- For each base clustering φ_m, draw $y_{rcm} \sim \text{Discrete}(\vec{\theta}_{mk})$

5.2 MPCCE Inference. We perform Markov Chain Monte Carlo (MCMC) simulation on the posterior distribution over $\mathcal{M}$, $\vec{\xi}$, $\vec{\eta}$, and $\vec{\theta}$. The joint distribution of observed base co-clustering results $\vec{Y}$, hidden variable $\mathcal{M}$, $\vec{\xi}$ and $\vec{\eta}$, and model parameters $\vec{\theta}$ is:

$$p(\vec{Y}, \mathcal{M}, \vec{\xi}, \vec{\eta}, \vec{\theta}|\beta, \lambda) = p(\mathcal{M}|\lambda) \left(\prod_{r=1}^{R} p(\xi_r)\right) \left(\prod_{c=1}^{C} p(\eta_c)\right) \left(\prod_{k=1}^{K}\prod_{m=1}^{M} p(\vec{\theta}_{mk}|\beta)\right) \left(\prod_{r=1}^{R}\prod_{c=1}^{C}\prod_{m=1}^{M} p(y_{rcm}|\vec{\theta}, \mathcal{M}, \xi_r, \eta_c)\right). \tag{5.11}$$

We can integrate out the model parameter $\vec{\theta}$ because of conjugacy:

$$p(\vec{Y}, \mathcal{M}, \vec{\xi}, \vec{\eta}|\beta, \lambda) = p(\mathcal{M}|\lambda) \left(\prod_{r=1}^{R} p(\xi_r)\right) \left(\prod_{c=1}^{C} p(\eta_c)\right) \left(\prod_{k=1}^{K}\prod_{m=1}^{M} \frac{\Gamma(J_m\beta_m)}{\Gamma(J_m\beta_m + \mathcal{N}_k)} \times \prod_{j_m=1}^{J_m} \frac{\Gamma(\beta_m + \mathcal{N}_k^{y_{\cdot\cdot m}=j_m})}{\Gamma(\beta_m)}\right), \tag{5.12}$$

where $\mathcal{N}_k$ denotes the number of entries in the k^{th} block of $\mathcal{M}$, and $\mathcal{N}_k^{y_{\cdot\cdot m}=j_m}$ denotes the number of entries in both the k^{th} block of $\mathcal{M}$ and the ${j_m}^{th}$ co-cluster of φ_m.

We perform Gibbs sampling on the row and column coordinates $\vec{\xi}$ and $\vec{\eta}$. Since ξ_r and η_c have uniform prior distributions, their posterior distributions are piece-wise constant [15]. Define $\vec{\chi}^R = \langle \chi_h^R | h \in \{0, \cdots, L_R, L_R + 1\}\rangle$, where $\chi_0^R = 0$, $\chi_h^R < \chi_{h+1}^R$, $\chi_{L_R+1}^R = 1$. The value χ_h^R is the position of the h^{th} horizontal cut of the total L_R horizontal cuts in $\mathcal{M}$. The conditional probability that ξ_r falls in the interval (χ_h^R, χ_{h+1}^R) is:

$$p(\chi_h^R < \xi_r < \chi_{h+1}^R | \vec{X}, \mathcal{M}, \vec{\xi}^{\neg r}, \vec{\eta}, \beta, \lambda) \propto (\chi_{h+1}^R - \chi_h^R)\left(\prod_{k=1}^{K}\prod_{m=1}^{M} \frac{\Gamma(J_m\beta_m)}{\Gamma(J_m\beta_m + \mathcal{N}_k^{\neg r})} \times \prod_{j_m=1}^{J_m} \frac{\Gamma(\beta_m + \mathcal{N}_{k,y_{\cdot\cdot m}=j_m}^{\neg r})}{\Gamma(\beta_m)}\right). \tag{5.13}$$

Similarly, let $\vec{\chi}^C = \langle \chi_g^C | g \in \{0, \cdots, L_C, L_C + 1\}\rangle$, where $\chi_0^C = 0$, $\chi_g^C < \chi_{g+1}^C$, $\chi_{L_C+1}^C = 1$. The value χ_g^C is the position of the g^{th} vertical cut of the total L_C vertical cuts in $\mathcal{M}$. The conditional probability that η_c falls in the interval (χ_g^C, χ_{g+1}^C) is:

$$p(\chi_g^C < \eta_c < \chi_{g+1}^C | \vec{X}, \mathcal{M}, \vec{\xi}, \vec{\eta}^{\neg c}, \beta, \lambda) \propto (\chi_{g+1}^C - \chi_g^C)\left(\prod_{k=1}^{K}\prod_{m=1}^{M} \frac{\Gamma(J_m\beta_m)}{\Gamma(J_m\beta_m + \mathcal{N}_k^{\neg c})} \times \prod_{j_m=1}^{J_m} \frac{\Gamma(\beta_m + \mathcal{N}_{k,y_{\cdot\cdot m}=j_m}^{\neg c})}{\Gamma(\beta_m)}\right). \tag{5.14}$$

In these equations, the superscripts $\neg r$ and $\neg c$ mean that the r^{th} row and c^{th} column are excluded in the respective counts. Accordingly, we have:

$$\theta_{mkj_m} \propto \beta_m + \mathcal{N}_k^{y_{\cdot\cdot m}=j_m}. \tag{5.15}$$

Reversible jump MCMC (RJMCMC) [9] is used to sample from the posterior distribution $p(\mathcal{M}|\vec{Y}, \vec{\xi}, \vec{\eta}, \beta, \lambda)$. A state $\mathcal{M}$ consists of a tree of blocks and a vector $\vec{\zeta}$ of parameters. The parameters consist of a cost E_k and a location χ_k of the cut to each non-leaf block of the tree. The location χ_k ranges between zero and τ_k, where τ_k is half the length of the block perimeter. If χ_k is less than the width of the block, a vertical cut is made at position χ_k along the width; otherwise, a horizontal cut is made along the height of the block at position equal to χ_k minus the block width.

Each MCMC proposal either removes a pair of sibling leaf blocks or adds a cut to a leaf block. When a leaf block k is split into child blocks k' and k'', the parameter $\vec{\zeta}$ is extended to $\langle \vec{\zeta}, E_k, \chi_k \rangle$. When a split is removed, the associated cost E_k and location χ_k are removed from $\langle \vec{\zeta}, E_k, \chi_k \rangle$ to obtain $\vec{\zeta}$. RJMCMC

maintains reversibility of moves by adding auxiliary parameters so that moves occur between spaces of equal dimensions. When proposing to add a cut, we augment the current parameter $\vec{\zeta_t}$ and define a bijection between the augmented parameter $\langle\vec{\zeta_t}, u_1, u_2\rangle$ and the proposed parameter $\vec{\zeta}_{t+1} = \langle\vec{\zeta_t}, E_k, \chi_k\rangle$:

$$g^{add}_{t\to t+1}(\langle\vec{\zeta_t}, u_1, u_2\rangle) = \langle\vec{\zeta_t}, E_k, \chi_k\rangle. \tag{5.16}$$

Similarly, when proposing to remove a cut, we augment the proposed state $\vec{\zeta}_{t+1}$ and define a bijection between the current state $\vec{\zeta_t}$ and the augmented proposed state $\langle\vec{\zeta}_{t+1}, u_1, u_2\rangle$:

$$g^{remove}_{t\to t+1}(\vec{\zeta_t}) = g^{remove}_{t\to t+1}(\langle\vec{\zeta}_{t+1}, E_k, \chi_k\rangle) = \langle\vec{\zeta}_{t+1}, u_1, u_2\rangle. \tag{5.17}$$

The proposal distribution $Q(\mathcal{M}_{t+1}; \mathcal{M}_t)$ chooses with equal probability whether to add or remove a cut, and uses a uniform discrete distribution to sample the block at which to add or remove the cut. When a cut at block k is being added, $Q(\mathcal{M}_{t+1}; \mathcal{M}_t)$ proposes a location χ_k from a uniform distribution and a cost E_k from an exponential distribution with parameter τ_k. When a cut at block k is being removed, $Q(\mathcal{M}_{t+1}; \mathcal{M}_t)$ sets the new parameter $\vec{\zeta}_{t+1}$ deterministically by removing the cost E_k and location χ_k from the current state $\vec{\zeta_t}$, and the auxiliary parameters are then sampled from a distribution $q(u_1, u_2)$. The parameter u_1 is sampled from the same exponential distribution used to sample the cost of a new cut at k, and the parameter u_2 is sampled from the same uniform distribution used to sample the location of a new cut at k.

Following [9], the proposal to remove a cut is accepted if α drawn from $Uniform(0, 1)$ satisfies:

$$\alpha < \min\Bigg\{1, \frac{p(\mathcal{M}_{t+1}|\vec{Y}, \vec{\xi}, \vec{\eta}, \beta, \lambda)}{p(\mathcal{M}_t|\vec{Y}, \vec{\xi}, \vec{\eta}, \beta, \lambda)} \times \frac{Q(\mathcal{M}_t; \mathcal{M}_{t+1})}{Q(\mathcal{M}_{t+1}; \mathcal{M}_t)q(u_1, u_2)} \times \left|\frac{\partial\langle\vec{\zeta}_{t+1}, u_1, u_2\rangle}{\partial\vec{\zeta_t}}\right|\Bigg\}, \tag{5.18}$$

where $\left|\frac{\partial\langle\vec{\zeta}_{t+1}, u_1, u_2\rangle}{\partial\vec{\zeta_t}}\right|$ is the Jacobian of $g^{remove}_{t\to t+1}(\vec{\zeta_t})$. The acceptance probability for adding a cut is obtained in a similar manner. See [9] for details on RJMCMC.

To calculate the acceptance ratio in Equation (5.18), we need to calculate two ratios $\frac{Q(\mathcal{M}_t; \mathcal{M}_{t+1})}{Q(\mathcal{M}_{t+1}; \mathcal{M}_t)q(\vec{U}_{t+1})}$ and $\frac{p(\mathcal{M}_{t+1}|\vec{Y}, \vec{\xi}, \vec{\eta}, \beta, \lambda)}{p(\mathcal{M}_t|\vec{Y}, \vec{\xi}, \vec{\eta}, \beta, \lambda)}$. The first of these involves only the proposal distributions, and is straightforward to calculate. The second of these, the ratio of posterior probabilities of $\mathcal{M}_{t+1}$ and $\mathcal{M}_t$, is equal to the prior odds ratio times the likelihood ratio:

$$\frac{p(\mathcal{M}_{t+1}|\vec{Y}, \vec{\xi}, \vec{\eta}, \beta, \lambda)}{p(\mathcal{M}_t|\vec{Y}, \vec{\xi}, \vec{\eta}, \beta, \lambda)} = \frac{p(\mathcal{M}_{t+1}|\lambda)}{p(\mathcal{M}_t|\lambda)} \frac{\mathcal{L}(\mathcal{M}_{t+1})}{\mathcal{L}(\mathcal{M}_t)}, \tag{5.19}$$

where $\mathcal{L}(\mathcal{M}_{t+1})$ and $\mathcal{L}(\mathcal{M}_t)$ are the likelihood of $\mathcal{M}_{t+1}$ and $\mathcal{M}_t$, which are defined as:

$$\mathcal{L}(\mathcal{M}_{t+1}) = \Bigg(\prod_{k_{t+1}=1}^{K_{t+1}} \prod_{m=1}^{M} \frac{\Gamma(J_m\beta_m)}{\Gamma(J_m\beta_m + \mathcal{N}_{k_{t+1}})} \times \prod_{j_m=1}^{J_m} \frac{\Gamma(\beta_m + \mathcal{N}^{y_{\cdot\cdot m}=j_m}_{k_{t+1}})}{\Gamma(\beta_m)}\Bigg), \tag{5.20}$$

$$\mathcal{L}(\mathcal{M}_t) = \Bigg(\prod_{k_t=1}^{K_t} \prod_{m=1}^{M} \frac{\Gamma(J_m\beta_m)}{\Gamma(J_m\beta_m + \mathcal{N}_{k_t})} \times \prod_{j_m=1}^{J_m} \frac{\Gamma(\beta_m + \mathcal{N}^{x_{\cdot\cdot m}=j_m}_{k_t})}{\Gamma(\beta_m)}\Bigg). \tag{5.21}$$

For a proposal to remove a cut of block k into blocks k' and k'', the prior odds ratio is given by:

$$\frac{p(\mathcal{M}_{t+1}|\lambda)}{p(\mathcal{M}_t|\lambda)} = \frac{\omega_k}{p(\chi_k)p(E_k)\omega_{k'}\omega_{k''}}, \tag{5.22}$$

where ω_k is the probability that sampling terminates with no cut at block k; this happens when the cost E_k exceeds the budget λ_k. The cut cost E_k is generated from an exponential distribution with parameter τ_k. Thus, the probability of terminating with no split at block k is given by:

$$\omega_k = \int_{\lambda_k}^{+\infty} \tau_k \exp(-\tau_k e)de = \exp(-\tau_k\lambda_k). \tag{5.23}$$

Similarly, $\omega_{k'} = \exp(-\tau_{k'}\lambda_{k'})$ and $\omega_{k''} = \exp(-\tau_{k''}\lambda_{k''})$. Note that a block's budget is equal to its parent's budget minus the cost of cutting the parent. Thus, $\lambda'_k = \lambda''_k = \lambda_k - E_k$; and λ_k can be computed recursively from the budgets and cut costs of its ancestors.

A similar calculation gives the acceptance ratio for adding a random cut to $\mathcal{M}_t$ to generate $\mathcal{M}_{t+1}$. The inference procedure for MPCCE is given in Algorithm 2.

Algorithm 2 Inference for MPCCE

Input λ, β and $\vec{Y}$; randomly initialize $\vec{\xi}$ and $\vec{\eta}$
$t \leftarrow 0$
$\mathcal{M}_0$ has no cut
budget$\leftarrow \lambda$
repeat
  $t \leftarrow t+1$
  Propose $\mathcal{M}_{t+1}$ conditioned on $\mathcal{M}_t$ by either adding or removing a cut
  Accept or reject $\mathcal{M}_{t+1}$ according to Equation (5.18)
  if reject **then**
    $\mathcal{M}_{t+1} \leftarrow \mathcal{M}_t$
  else
    $\mathcal{M}_{t+1} \leftarrow \mathcal{M}_{t+1}$
  end if
  Gibbs sample $\vec{\xi}$ and $\vec{\eta}$ according to Equation (5.13) and (5.14)
until Stopping criteria met
Output the final $\mathcal{M}$, $\vec{\xi}$ and $\vec{\eta}$

6 Experiments.

6.1 Data. We conducted experiments on synthetic and real data. Following [15], we synthetically generated non grid-style clusters by sampling from a Mondrian process on the unit square. We then generated 250 row and 250 column coordinates from a uniform distribution, and set the data value to the cluster ID for the block at those coordinates. Finally, we permuted the rows and columns randomly to form the final data matrix. We also used two real datasets: (a) MovieLens[1] is a movie recommendation dataset containing 100,000 ratings in a sparse data matrix for 1682 movies rated by 943 users. (b) Jester[2] is a joke rating dataset. The original dataset contains 4.1 million continuous ratings of 100 jokes from 73,421 users. Following [18], we chose 1000 users who rated almost all jokes, discretized the ratings, and used this dense data matrix in our experiment. For both real datasets, we held out 25% of the data for testing.

6.2 Methodology. We compared DPCCE and MPCCE with other generative co-clustering approaches: Latent Dirichlet Co-clustering (LDCC) [18, 21], Dirichlet Process-based Co-clustering (DPCC) [12], and Mondrian Process-based Co-clustering (MPCC) [15]. LDCC requires specification of the numbers of row- and column-clusters. For the synthetic dataset, we varied the numbers of both row- and column-clusters from 5 to 10. For MovieLens, we set the number of user clusters to 20, the number of occupation categories, and the number of movie clusters to 19, the number of genres. For Jester, we used 5 joke clusters and 20 user clusters; this is the number of clusters given in the data description. The pseudo-counts of the DP priors for both rows and columns in DPCC and DPCCE are set to 20. We ran DPCC and MPCC five times with different random initializations, to generate five base co-clustering results. We then ran DPCCE and MPCCE based on the DPCC and MPCC results, respectively. We repeated DPCCE and MPCCE five times, each time with five different base co-clusterings. For MPCCE and MPCC we set the budget $\lambda = 1$, and let μ_d be Lebesgue measure. We ran DPCC and DPCCE for 3000 iterations, and MPCC and MPCCE for 1000 iterations.

We evaluated the models using perplexity: $\text{perp}(\vec{X}) = \exp(-(\log p(\vec{X}))/N)$, where N is the number of non-missing entries in $\vec{X}$. For the two real datasets, we report perplexity on both training and test sets; for the synthetic data, we report only training perplexity. If the chain mixes well and is run sufficiently long, each sample of five DPCC or MPCC results used to fit the DPCCE and MPCCE models can be viewed as a sample from the DPCC or MPCC posterior distribution, respectively. We therefore also evaluated a model averaging approach, in which we calculated the perplexity based on the average of the five DPCC or MPCC likelihood results.

6.3 Results. We present two main experimental comparisons: (a) perplexity comparisons on the synthetic data and the training sets for the real datasets; and (b) perplexity comparisons on the test sets for the real datasets.

6.3.1 Perplexity Comparison on Training Datasets. Figure 2 (left) shows the original non-grid style synthetic data matrix. After permuting its rows and columns, this matrix was input to the base co-clustering algorithms for DPCCE and MPCCE. Figure 2 (right) shows the corresponding grid-style partition of the original synthetic data matrix. Clearly, the grid-style partition of DPCCE over-segments the data, whereas the partition provided by MPCCE reflects the actual data distribution.

Table 2 shows the perplexity results for the training data. Each entry shows an average perplexity over five runs[3], with the standard deviation of the average shown in parentheses. The benefit of the non-grid parti-

[1] http://www.grouplens.org/node/73
[2] http://goldberg.berkeley.edu/jester-data/

[3] For DPCC and MPCC, the estimate for each run is the average of the results for the five base co-clusterings.

Table 2: Perplexity Comparison on Training Datasets

	Synthetic	MovieLens	Jester
LDCC	4.782 (0.025)	3.045 (0.026)	18.896 (0.072)
DPCC	3.723 (0.026)	2.797 (0.028)	15.984 (0.073)
Model Avg. of DPCC	3.687 (0.039)	2.312 (0.040)	14.223 (0.115)
DPCCE	3.573 (0.037)	2.130 (0.033)	13.677 (0.107)
MPCC	1.626 (0.023)	2.473 (0.043)	12.035 (0.088)
Model Avg. of MPCC	1.486 (0.046)	2.386 (0.051)	10.968 (0.142)
MPCCE	1.255 (0.038)	2.124 (0.037)	9.785 (0.122)

tion is demonstrated by the improvement of MPCC and MPCCE over LDCC, DPCC and DPCCE. The efficacy of the ensemble approach is demonstrated by the improvement of MPCCE and DPCCE over MPCC and DPCC, respectively. The model averaging estimates perform better than their respective non-ensemble counterparts, but not as well as the ensemble estimates. All nonparametric approaches perform better than LDCC. Note that for MovieLens, MPCCE performs only 2% better than DPCCE, a difference that cannot be distinguished from sampling noise. This may indicate that a grid structure of independent user and movie groups provides a good fit to the MovieLens data. For the Jester dataset, the perplexities are relatively high for all models. This is due to the large number of missing values in this dataset.

All experiments were run on a CentOS 5.5 server running Linux on a 4-core CPU with 4GB memory. The running time for 1000 iterations of MPCC was approximately 4 hours on MovieLens and 3 hours on Jester. For 1000 iterations of MPCCE, the running time was about 6 hours on MovieLens and 4 hours on Jester. For DPCC and DPCCE, 3000 iterations ran in under 3 hours.

Figure 4 plots the log-likelihoods on the Jester dataset for 5 MPCC runs and one MPCCE run initialized with iteration 1000 of the 5 MPCC runs. For comparison, we also continued the MPCC runs for another 1000 iterations. All chains appear to have reached different local optima. The local optimum for MPCCE has higher likelihood than all five MPCC local optima. The Potential Scale Reduction Factor MCMC diagnostic [8] for the 5 MPCC log-likelihood values plotted in Figure 4 is 3.0043, which is also indicative of non-convergence. The other MPCC and MPCCE runs followed the same pattern. These results suggest that the ensemble method finds superior local optima for samplers that mix poorly. Note running MPCCE for 1000 iterations requires less computation time than continuing the 5 MPCC runs for a second 1000 iterations, and results in a superior local optimum.

6.3.2 Perplexity Comparison on Test Datasets.

Predictive performance was evaluated by measuring perplexity on the test data for the two real datasets. Table 3 shows the prediction comparison results. Again, the results are reported as an average perplexity over multiple predictions, with the standard deviation of each average in parentheses.

Again, all nonparametric methods perform better than LDCC; clustering ensembles perform better than model averaging, which performs better than single-run methods; and the MP methods perform better than grid-style clustering. Statistical significance tests indicate that the improvement due to the ensemble method is much greater than expected from chance variation. Paired t-tests of the hypothesis that the mean perplexities are the same were significant at $p < 10^{-4}$ for MPCCE vs MPCC and for DPCC vs DPCCE, on both the MovieLens and Jester data sets. Although the differences remain smaller for MovieLens than for Jester, the improvement in both MovieLens and Jester due to the non-grid partitions of the MP exceeds sampling error. That co-clustering ensembles perform better than model averaging on both training and test sets for all data sets is consistent with the hypothesis that poor mixing of the MCMC algorithms for DPCC and MPCC kept the chains near local optima of the posterior distribution, and that the ensemble algorithms can combine information from multiple local optima to find a superior co-clustering.

7 Conclusion.

We have presented two nonparametric Bayesian co-clustering ensemble models, one based on Dirichlet Processes and the other based on Mondrian Processes. The latter relaxes the usual co-clustering assumption that row- and column-clusters are independent, providing a way to model context-specific independence of row- and column-clusters. The empirical evaluation demonstrated that nonparametric clustering ensemble methods can improve both fit and predictive performance over traditional co-clustering methods, and that the in-

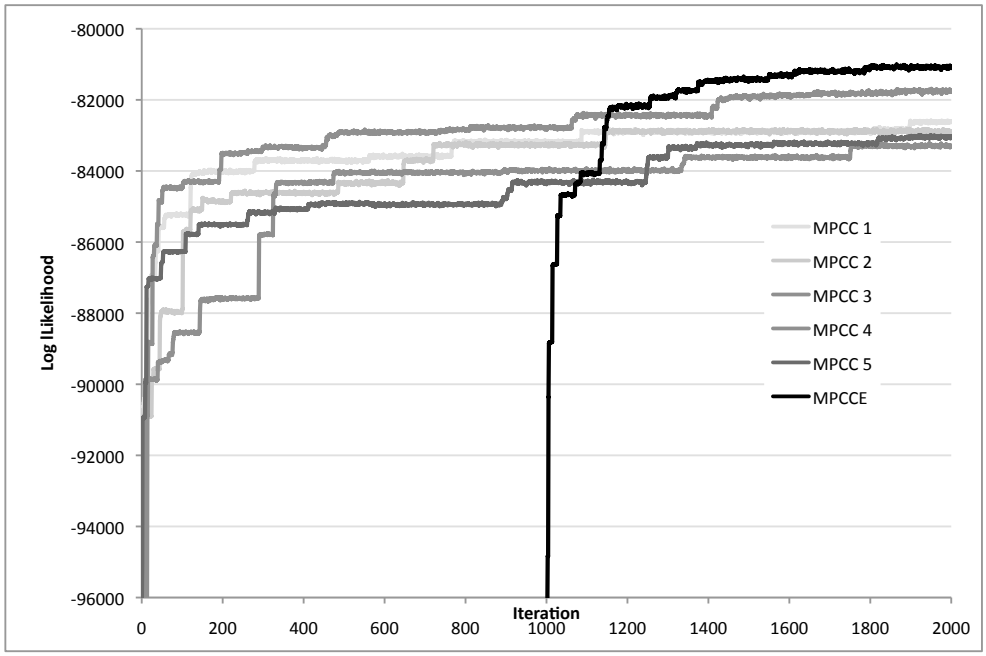

Figure 4: MPCC and MPCCE Likelihood Comparison

Table 3: Perplexity Comparison on Test Datasets

	MovieLens	Jester
LDCC	3.247 (0.052)	23.743 (0.236)
DPCC	2.908 (0.055)	20.174 (0.219)
Model Avg. of DPCC	2.838 (0.079)	19.165 (0.421)
DPCCE	2.707 (0.060)	18.092 (0.458)
MPCC	2.793 (0.067)	13.781 (0.263)
Model Avg. of MPCC	2.738 (0.089)	13.433 (0.379)
MPCCE	2.626 (0.084)	12.036 (0.438)

creased flexibility of the Mondrian process can improve both fit and predictive performance over independently clustering rows and columns. The ability of ensemble methods to incorporate complementary aspects of multiple local optima may have applications to other problems with combinatorial structure.

Acknowledgement

This work is in part supported by NSF CAREER Award IIS-0447814.

References

[1] C. E. Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

[2] D. Blackwell and J. B. Macqueen. Ferguson distributions via pólya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98, New York, NY, 2003. ACM.

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

[6] X. Fern and C. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *International Conference on Machine Learning*, pages 281–288, 2004.

[7] A. Fred and A. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005.

[8] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis, Second Edition (Chapman & Hall/CRC Texts in Statistical Science)*. Chapman and Hall/CRC, 2 edition, 2003.

[9] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[10] F. Gullo, C. Domeniconi, and A. Tagarelli. Projective clustering ensembles. In *IEEE International Conference on Data Mining*, pages 794–799, 2009.

[11] L. Kuncheva, S. Hadjitodorov, and L. Todorova. Experimental comparison of cluster ensemble methods. In *International Conference on Information Fusion*, pages 1–7, 2006.

[12] E. Meeds and S. Roweis. Nonparametric Bayesian biclustering. Technical Report UTML TR 2007-001, Department of Computer Science, University of Toronto, 2007.

[13] R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

[14] O. Papaspiliopoulos and G. O. Roberts. Retrospective markov chain monte carlo methods for dirichlet process hierarchical models. *Biometrika*, 95(1):169–186, March 2008.

[15] D. M. Roy and Y. W. Teh. The Mondrian process. In *Advances in Neural Information Processing Systems (NIPS)*, volume 21, 2008.

[16] J. Sethuraman. A constructive definition of dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[17] M. Shafiei and E. Milios. Latent Dirichlet co-clustering. In *IEEE International Conference on Data Mining*, pages 542–551, 2006.

[18] H. Shan and A. Banerjee. Bayesian co-clustering. In *IEEE International Conference on Data Mining (ICDM)*, 2008.

[19] A. Strehl and J. Ghosh. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2003.

[20] A. Topchy, A. Jain and W. Punch. A mixture model for clustering ensembles. In *SIAM International Conference on Data Mining*, pages 379–390, 2004.

[21] P. Wang, C. Domeniconi, and K. Laskey. Latent Dirichlet Bayesian co-clustering. In *Proceedings of the European Conference on Machine Learning*, volume 5782, pages 522–537. Springer Berlin Heidelberg, 2009.