Conjoined Dirichlet Process

Michelle N. Ngo [*1] **Dustin S. Pluta** [*2] **Alexander N. Ngo** [3] **Babak Shahbaba** [1 2 4]

Abstract

Biclustering is a class of techniques that simultaneously clusters the rows and columns of a matrix to sort heterogeneous data into homogeneous blocks. Although many algorithms have been proposed to find biclusters, existing methods suffer from the pre-specification of the number of biclusters or place constraints on the model structure. To address these issues, we develop a novel, non-parametric probabilistic biclustering method based on Dirichlet processes to identify biclusters with strong co-occurrence in both rows and columns. The proposed method utilizes dual Dirichlet process mixture models to learn row and column clusters, with the number of resulting clusters determined by the data rather than pre-specified. Probabilistic biclusters are identified by modeling the mutual dependence between the row and column clusters. We apply our method to two different applications, text mining and gene expression analysis, and demonstrate that our method improves bicluster extraction in many settings compared to existing approaches.

1. Introduction

Biclustering, or co-clustering, is a technique used for sorting heterogeneous data into homogeneous blocks by allowing for simultaneous clustering of the rows and columns of a matrix. This technique has various important applications, including text mining and biological gene expression analysis. In text mining, biclustering text data from a document corpus allows for identification of document-word combinations with high co-occurrence. Extracted biclusters represent combinations of words and documents that form a (latent) topic. Biclustering has been particularly popular in the past several decades for gene expression microarray analyses. The method is used to group genes into similar conditions to study the functional roles of genes. More recently, biclustering is being used to analyze single cell RNA sequencing data. Here, the method is usually used to study cell proliferation by grouping cells into developmental stages and identifying the genetic drivers for each stage.

Current biclustering methods generally impose restrictive assumptions on the biclustering structure or data-generating mechanisms. However, in real-world applications, which are often exploratory, an appropriate model and bicluster structure can be difficult to specify. To address these limitations in current methods, we propose the Conjoined Dirichlet Process (CDP): a novel, non-parametric probabilistic biclustering method based on dual Dirichlet processes to identify biclusters with strong co-occurrences in both rows and columns. The name of the method derives from its usage of two conjoined DPMMs, akin to conjoined twins (see Figure 1). CDP provides the following advantages: 1) the number of biclusters is determined by the data and prior, and does not require selecting a number of clusters *á priori*, 2) fewer modeling assumptions compared to parametric alternatives, 3) estimated biclusters may overlap arbitrarily, and 4) efficient computational methods allow applications to high dimensional data, making applications to text and gene expression data practical.

The paper is organized as follows. In Section 2 we describe existing biclustering methods. In Section 3 we provide some background on Dirichlet process mixture models (DPMMs), particularly focusing on the parallel MCMC sampler for DPMMs. In Section 4 we discuss and provide details of our proposed biclustering method. In Section 5 we apply our method to simulated, text, and single cell RNA sequencing data sets, and present the results. Finally, in Section 6 we present our conclusion.

2. Previous Methods

Briefly, biclustering algorithms are based on four heuristics: greedy, divide-and-conquer, exhaustive enumeration, or distribution parameter identification (Padilha & Campello, 2017).

(Hartigan, 1972) proposed the first biclustering algorithm in 1972, but the technique was not popular until 2000 when

[*]Equal contribution [1]Center for Complex Biological Systems, University of California at Irvine [2]Department of Statistics, University of California at Irvine [3]Department of Computer Science, University of California at Santa Barbara [4]Department of Computer Science, University of California at Irvine . Correspondence to: Babak Shahbaba <babaks@uci.edu>.

Updated 07 February 2020

(Cheng & Church St, 2000) applied it to gene microarray data. Other popular gene microarray biclustering algorithms include (Kluger et al., 2003)'s spectral model and (Lazzeroni & Owen, 2002)'s plaid model.

While many biclustering algorithms have been developed for gene microarray analyses, one of the first applications for biclustering was text mining. Dhillon et al proposed two different biclustering algorithms for simultaneously partitioning documents and words: spectral co-clustering (Dhillon, 2001), and a co-clustering algorithm based on information theory (Dhillon et al., 2003). (Kluger et al., 2003)'s spectral model for gene microarray analyses is based on (Dhillon, 2001)'s spectral model.

More recently, biclustering has been applied to single cell RNA sequencing (scRNA seq) data. Biclustering methods specific to this application include BackSPIN (Zeisel et al., 2015) and QUBIC2 (Xie et al., 2019).

(Rugeles et al., 2017) developed Dual Topics for Bicluster (DT2B), a biclustering method based on a generalized latent Dirichlet allocation (LDA) model (Blei et al., 2003). Unlike the previous models, DT2B avoids the constraints of a model structure. However, the algorithm requires a discretized data set, pre-specification of the number of row and column clusters, and threshold values.

By using a Dirichlet process mixture model (DPMM) instead of a LDA model, we bypass the need to specify the number of biclusters, make strong modeling assumptions, and particular data format.

3. Background

3.1. Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) is a hierarchical Bayesian model used to infer latent features in collections of discrete data. Initially proposed to estimate and describe population structure from genotype data (Pritchard et al., 2000), it is also commonly used for the classification of documents based on word frequencies (Blei et al., 2003).

In the context of document classification, LDA posits that for a corpus of documents, the probability distribution of words for a given document is determined by a set of latent "topics" associated with that document. LDA infers these latent topics from observed word frequencies for each document to produce a clustering or classification of documents in the corpus.

3.2. Dirichlet Process Mixture Model

DPMMs remove the need to pre-specify the number of clusters by placing a Dirichlet process (DP) prior over the cluster parameters, and in this sense, allows "infinite" mixture models to incorporate automatic model selection.

The DPMM is intuitively an infinite dimensional generalization of a mixture of Dirichlet distributions. We begin by considering a Bayesian mixture model with K clusters and then extending $K \to \infty$:

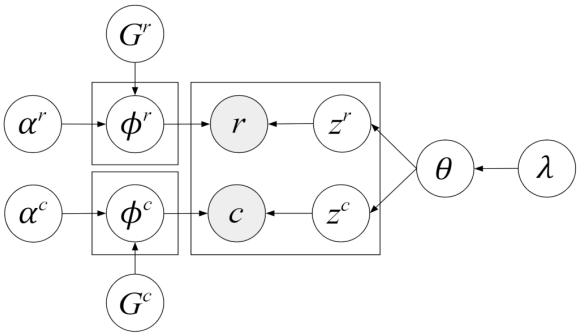
$$\begin{aligned} x_i|z_i, \theta_i &\sim F(\theta_{c_i}) \\ z_i|\boldsymbol{\pi} &\sim Discrete(p_1, \ldots, p_K) \\ \theta_c &\sim G_0 \\ \boldsymbol{\pi} &\sim Dirichlet\left(\frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right) \end{aligned}$$

Here $x_1, \ldots, x_n$ is the observed data and drawn from a mixture of distributions with the form $F(\theta)$, θ is the mixing distribution over G, and z is the cluster assignments for each observation (Neal, 2000). The prior for our mixing distribution is a Dirichlet process with concentration parameter α and base distribution G_0. For a more in-depth explanation, see (Sudderth & Freeman, 2006).

3.3. Parallel Sampling of DPMMs

DPMMs have been largely computationally heavy to implement. (Chang & Fisher III, 2013) parallelized the MCMC sampler for DPMMs by utilizing a restricted Gibbs sampler to fix the number of clusters before proposing splits and merges. Since the number of clusters are fixed, each of the Gibbs sampler steps can be done in parallel. Furthermore, to increase efficient cluster splits, they augment each cluster with two sub-clusters, labeled $\bar{z}_i \in \{l, r\}$ to denote whether each data point x_i is associated with the left or right sub-cluster. Additional auxiliary variables introduced are the sub-cluster weights $\bar{\pi}_k \in \{\bar{\pi}_{k,l}, \bar{\pi}_{k,r}\}$ and parameters $\bar{\theta}_k \in \{\bar{\theta}_{k,l}, \bar{\theta}_{k,r}\}$ of cluster k. The auxiliary variables for the sub-clusters are analogous in function to the variables for the regular clusters. In this augmented restricted Gibbs sampling algorithm, we now sample a regular cluster assignment and then a sub-cluster assignment for each data point. Splits and merges, to either split a cluster into its two sub-clusters or merge two sub-clusters into one new cluster, are proposed and accepted with probability $\min(1, H)$, where $H \in \{H_{split}, H_{merge}\}$ is the Hastings ratio for the respective action.

(Dinari et al., 2019) extended this implementation to enable parallelization on multiple multi-core machines instead of a single multi-core machine. The authors note that sampling cluster parameters θ_k is parallelizable over the clusters, sampling cluster assignments z_i is independently computed for each data point x_i, and proposing cluster splits is parallelizable. For computational efficiency, they rely on a distributed-memory model and utilize sufficient statistics to communicate between the cores as well as the between the machines. The sufficient statistic T for a multinomial cluster (e.g. for document classification or single cell RNA sequencing data analysis) is $T = \sum_{i=1}^{N} x_i \in \mathbb{N}_0^d$, where



Variable	Distribution	Description
r_i	$Cat_\infty(\phi_r)$	Observed row
c_i	$Cat_\infty(\phi_c)$	Observed column
(z_i^r, z_i^c)	$Cat_{\infty\times\infty}(\theta)$	Row and col. latent variables
θ	$Dirichlet(\lambda\|z^r, z^c)$	Joint latent variable distribution
ϕ^r	$DP(\alpha^r)$	Row latent variable distribution
ϕ^c	$DP(\alpha^c)$	Col latent variable distribution
α^r, α^c	–	DP hyperparameters
λ	–	Hyperparameter for θ

Figure 1. Plate diagram and description of included variables and parameters. Rows r and columns c are clustered separately through the DPMMs defined by ϕ^r and ϕ^c. After updating according to Algorithm 1, heavy biclusters can be extracted from ϕ^r, ϕ^c, and θ (the joint distribution of latent cluster assignments given by z^r, z^c).

d is the dimension of the data points x_i. The aggregation of the sufficient statistics for each cluster allows for the sampling of cluster parameters across multiple parallelized worker processes. Splits and merges are proposed similarly to (Chang & Fisher III, 2013) on the master process, with mappings of old cluster assignments to new assignments broadcasted to all worker processes to individually update its data points. Using this multi-machine, multi-core implementation considerably speeds up our model and allows us to handle high dimensional data.

4. Conjoined Dirichlet Process (CDP)

CDP is a probabilistic biclustering method that provides several important characteristics in the context of gene-cell count analysis. The estimated biclusters may overlap, posterior probabilities of each element belonging to a given bicluster can be calculated, and heavy biclusters (showing strong co-occurrence in rows and columns) are encouraged. By utilizing a pair of DPMMs for bicluster estimation, CDP eliminates the need to specify the number of row topics and columns topics *á priori*, which is particularly relevant for both gene expression and document analysis where the number of topics and biclusters is unknown or ill-defined.

4.1. Model Construction

CDP can be summarized in two steps:

1. Use DPMMs to learn row and column clusters.
2. Model the mutual dependence between the row and column clusters to extract biclusters with strong co-occurrence values in both rows and columns.

Given a $n_R \times n_C$ matrix where n_R is the number of rows and n_C is the number of columns, each matrix entry (r, c) represents the frequency of row r in column c. For text data, this corresponds to the frequency of word r in document c and for single cell RNA sequencing data, this corresponds to the gene expression of gene r in cell c.

Using a DPMM, we can sequentially cluster the rows and columns of the matrix to obtain row-cluster assignments z^r and column-cluster assignments z^c. Similar to DT2B (Rugeles et al., 2017), we now have two sets of latent variables (e.g. topics for text data) and use these sets to extract biclusters with strong co-occurrence values in rows and columns.

Figure 1 shows the graphical model for CDP, where row r and column c are the rows and columns of the data matrix. z^r and z^c are the vectors of row and column cluster indices (assignments) respectively. ϕ^r is the row per row latent variable distribution, ϕ^c is the column per column latent variable distribution, and θ is the joint latent variable distribution. These three variables maintain the counting over the relationships between the data, latent variables and their mutual dependence. For discrete data, the hyperparameters for CDP are γ, the concentration parameter for the DP; β, the prior for the DP measure; α^r, the hyperparameter for ϕ^r; α^c, the hyperparameter for ϕ^c; and λ, the hyperparameter for θ. Figure 2 shows an illustrative example of CDP.

Theorem 1 *If row assignments z^r are held fixed, then the CDP update step is equivalent to a latent Dirichlet allocation update on z^c. A similar result holds if z^c is held fixed for updating z^r.*

Proof: In evaluating Eq. 4 to update z^c, we can then treat ϕ^r as a constant, yielding

$$P(z_i^c = j | c_i = m, r_i = n, \mathbf{z}_{-i}^c) \tag{1}$$
$$\propto \phi_{mj}^c \theta_{jk} \tag{2}$$
$$\propto \frac{C_{mj} + \alpha^c}{\sum_{m'} C_{m'i} + n_C \alpha^c} (C_{ij} + \lambda). \tag{3}$$

Updating z^c according to this probability is equivalent to the update given by LDA (Blei et al., 2003).

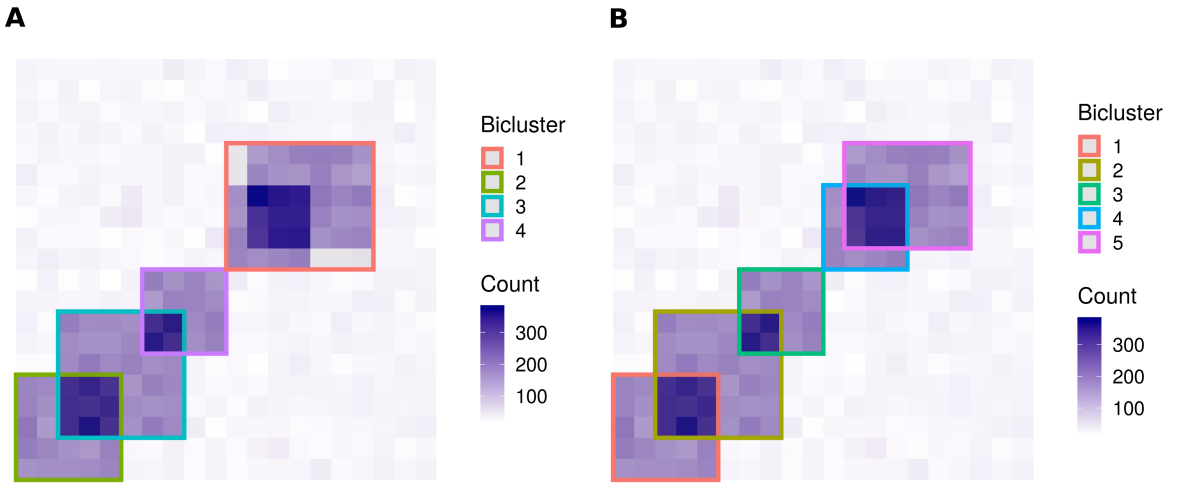


Figure 2. CDP is able to detect overlapping biclusters: (A) Heatmap of simulated count data and bicluster membership estimated by CDP. (B) True bicluster structure for simulated data.

4.2. Inference Process

Algorithm 1 shows the inference process for CDP, using the distributed MCMC inference algorithm outlined in (Dinari et al., 2019), which is based on the restricted Gibbs sampler method in (Chang & Fisher III, 2013). Due to the split and merge aspect and the high-dimensionality of our data, the posterior distribution of the assignment parameters may be multi-modal. For this reason, we update the assignment parameters for a specified number of iterations and take the MAP estimate of the maximum values of z^r and z^c.

The specifications for the hyperparameters of CDP (under the assumption that the base distribution of the DP is multinomial) are listed below:

$$\gamma^r, \gamma^c \in \mathbb{R}^{1\times 1} \qquad \alpha^r \in \mathbb{R}^{K_r\times 1}$$
$$\beta^r \in \mathbb{R}^{n_R\times 1} \qquad \alpha^c \in \mathbb{R}^{K_c\times 1}$$
$$\beta^c \in \mathbb{R}^{n_C\times 1} \qquad \lambda \in \mathbb{R}^{K_r\times K_c}$$

Given a collection of composites (e.g. documents, cells) C made up of parts (e.g. words, genes) R, we can write the probability of a composite c containing a part r as:

$$P(r,c) = \sum_{z_r}\sum_{z_c} P(r|\phi^r_{z_r}, \alpha^r)P(c|\phi^c_{z_c}, \alpha^c)P(z_r, z_c|\theta)$$

A major advantage of the CDP over DT2B (Rugeles et al., 2017) is that the CDP does not require thresholds to control the trade-off between quantity and quality of the biclusters. The hyperparameters in the DPMM step facilitate this trade-off automatically. Setting a large γ, the Dirichlet process concentration parameter, and for a multinomial base distribution, a large β (the Dirichlet distribution hyperparameter) will yield more clusters.

The probabilistic biclusters are given by the joint distribution of row and column latent variables, θ, which has dimension $K_r \times K_c$. K_r and K_c are the number of latent row and column variables respectively. As previously mentioned, from the DPMMs, we obtain the row-cluster assignments z^r and column-cluster assignments z^c. Calculating the mode of the posterior distributions of z^r and z^c yields the maximum a posteriori (MAP) estimate of the number of latent row and column variables, i.e. K_r and K_c.

We note that the dimensions of the row per row latent variable distribution, ϕ_r, and the column per column latent variable distribution, ϕ_c, are also given by the MAP. ϕ_r has dimensions $n_R \times K_r$, and ϕ_c has dimensions $n_C \times K_c$.

4.3. Bicluster Extraction

From the DPMM, we obtain latent variables z^r and z^c, which indicate the row and column cluster assignments respectively. To extract the biclusters from the data, we need to calculate three parameters: row per row latent variable distribution ϕ^r, column per column latent variable distribution ϕ^c, and joint distribution of row and column latent variables θ.

These three parameters are given by (Rugeles et al., 2017):

$$\phi^c_{mi} = \frac{C_{mi} + \alpha^c}{\sum_{m'} C_{m'i} + n_C\alpha^c} \tag{4}$$

$$\phi^r_{nj} = \frac{C_{nj} + \alpha^r}{\sum_{n'} C_{n'j} + n_R\alpha^r} \tag{5}$$

$$\theta \propto C_{ij} + \lambda \tag{6}$$

where C_{ab} is the number of instances a-th variable is assigned to b-th variable. For example, ϕ^r is the probability of the n-th row being assigned to j-th row latent variable.

Algorithm 1 Conjoined Dirichlet Process (CDP)

Input: Data $\boldsymbol{X}$, size $n_R \times n_C$
DP concentration parameters γ_R, γ_C
Dirichlet distribution hyperpriors β_R, β_C
Number of DPMM iterations $iter_R$, $iter_C$
Number of cluster reassignment iterations $iter_U$
for $i = 1$ **to** $iter_R$ **do**
  Run DPMM on $\boldsymbol{X}$
end for
for $i = 1$ **to** $iter_C$ **do**
  Run DPMM on $\boldsymbol{X}^T$
end for
Calculate $K_r = MAP(z^r)$ and $K_c = MAP(z^c)$
for $i = 1$ **to** $iter_U$ **do**
  Update z^r and z^c using the data as weights
end for
for $i = 1$ **to** n_R **do**
  for $j = 1$ **to** K_r **do**
    Calculate $\phi^r_{ij} = \frac{C_{ij}+\alpha^r}{\sum_{i'} C_{i'j}+n_R\alpha^r}$
  end for
end for
for $i = 1$ **to** n_C **do**
  for $j = 1$ **to** K_c **do**
    Calculate $\phi^c_{ij} = \frac{C_{ij}+\alpha^c}{\sum_{i'} C_{i'j}+n_C\alpha^c}$
  end for
end for
Calculate $\theta \propto C_{r,c} + \lambda, 1 \leq r \leq n_R, 1 \leq c \leq n_C$

Thus, C_{nj} is the number of times the n-th row is assigned to to j-th row latent variable. The joint distribution θ tracks the relationship between the current row and column latent variables to capture the mutual dependence between the two sets of latent variables (Rugeles et al., 2017).

First, we calculate ϕ^r and ϕ^c by using the aforementioned sets of latent variables z^r and z^c as the initial cluster assignments. These assignments are updated iteratively using the data as weights. Once the row and column assignments have been updated, we count the number of instances a row or column is assigned to a row or column latent variable.

To obtain the joint distribution of row and column latent variables θ, we need to calculate the frequency of each row and column latent variable pairing (i, j). The vector of frequencies for each row and column latent variable pairing is then transformed into a contingency table of size $K_r \times K_c$, i.e. the desired θ.

4.4. Implementation Overview

To obtain the row-cluster assignment z^r and column-cluster assignment z^c, we separately infer each parameter using (Dinari et al., 2019)'s implementation in Julia. We utilize a specific version of that package that outputs the cluster assignments z^r or z^c at each iteration rather than the final cluster assignments. While this requires more memory storage and run time, it allows CDP to have overlapping biclusters and more interpretable results depending on the application.

For N data observations, K clusters, and M machines with P cores, the total runtime complexity for the DPMM implementation is $\mathcal{O}(K) + \mathcal{O}(M + P) + \mathcal{O}(NK/(MP))$. For more details on the runtime complexity for the DPMM, see (Dinari et al., 2019).

CDP reassigns each observation iteratively in batches of size equal to either the row sums or column sums. These batches are parallelized to run on P processes (cores). Thus, updating the row and cluster assignments for J iterations takes $\mathcal{O}(NJ/P)$ time. Calculating ϕ for the rows and columns require the aforementioned assignment step. Once reassigned, CDP splits the N data points into vectors of length row sums (for ϕ^r) or column sums (for ϕ^c). These vectors are then tabulated over the number of latent variables K to determine the probability of each row or column being assigned to each row latent variable or column latent variable respectively. The runtime complexity for calculating ϕ excluding the cluster assignment update step is then $\mathcal{O}(NK)$ where K is equal to K_r when calculating ϕ^r and K_c when calculating ϕ^c. Calculating the joint distribution of both row and column latent variables θ requires looping over the assignments for one direction (e.g. row assignments) and matching the row and column indexes to the assignments in the other direction (e.g. column assignments). This operation requires $\mathcal{O}(N)$ time. CDP then tabulates the row and column assignment of each row and column pairing to obtain θ. Thus, the total runtime complexity for calculating θ is $\mathcal{O}(NK_rK_c)$ and $\mathcal{O}(NK_rK_c/P)$ if run in parallel.

As $N \gg K, P, M$ and J, CDP takes $\mathcal{O}(NJ)+\mathcal{O}(NK_rK_c)$ time. Experiments were conducted on an i5-7600K CPU.

5. Experimental Results

We compare CDP to DT2B (Rugeles et al., 2017) because this method also models the mutual dependency between two sets of latent variables. We also compare our algorithm to spectral biclustering (Kluger et al., 2003) since both try to extract high co-occurences. For completeness, Cheng and Church (Cheng & Church St, 2000) and the plaid (Lazzeroni & Owen, 2002) algorithms are also used for comparisons due to their common usage, and BiMax (Preli et al., 2006) which is known to serve as a reference method.

5.1. Data sets

5.1.1. Synthetic Data

Simulated count data were generated from a multinomial distribution defined by an $R \times C$ probability matrix θ (with

entries summing to 1), and by fixing the sum of entries in the resulting random matrix at some total count N. The total bicluster probability p of an element belonging to a bicluster was set to control the strength of biclusters and overall sparsity. Four different constructions of θ were chosen to evaluate performance over different biclustering patterns. In order of increasing complexity, these four cases are (1) a single distinct bicluster, $N = 4000, R = C = 50, p = 0.8$; (2) two distinct biclusters, $N = 4000, R = C = 20, p = 0.5$; (3) 3 biclusters with one overlap $N = 4000, R = C = 50, p = 0.7$; (4) 5 distinct biclusters, $N = 10000, R = C = 100, p = 0.7$ (see Figure 3 for an example).

To compare the performance of CDP to existing methods, we use the Jaccard score, defined as $J(\mathcal{B}_1, \mathcal{B}_2) = \min_{(\mathcal{A},\mathcal{B})} \frac{1}{|\mathcal{A}|} \sum_{A \in \mathcal{A}} \max_{B \in \mathcal{B}} \frac{|A \cap B|}{|A \cup B|}$, where $\mathcal{B}_1, \mathcal{B}_2$ are two sets of biclusters, with the minimum taken over $(\mathcal{A}, \mathcal{B}) \in \{(\mathcal{B}_1, \mathcal{B}_2), (\mathcal{B}_2, \mathcal{B}_1)\}$. The Jaccard score is a symmetric similarity metric taking values $0 \leq J(\mathcal{B}_1, \mathcal{B}_2) \leq 1$, with the lower bound attained only when all sets in $\mathcal{B}_1$ are disjoint with all sets in $\mathcal{B}_2$ and the upper bound attained only when $\mathcal{B}_1 = \mathcal{B}_2$. In the context of the simulation study, $\mathcal{B}_1$ is the set of estimated biclusters from a given method, and $\mathcal{B}_2$ is the set of true biclusters from the generative model.

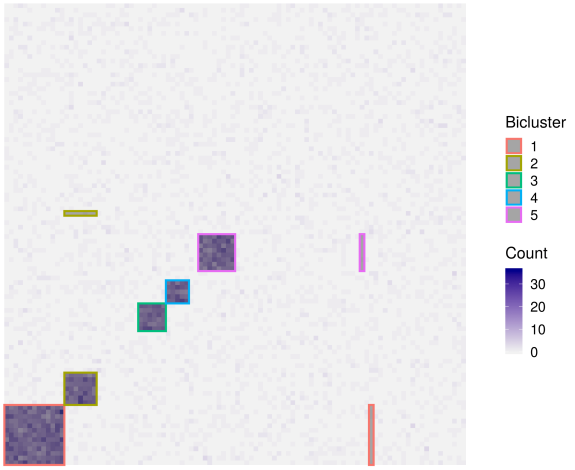


Figure 3. Example results from CDP for simulated data (case 4). CDP correctly identifies the heavy biclusters (0.938 Jaccard score), with only a small number of spurious elements included (e.g. in biclusters 1, 2 and 5). The data shown here is approximately 70% sparse.

5.1.2. Real-life Data

1. **Condensed 20 Newsgroups**: Collection of 100 words across 16,242 newsgroup documents ("netnews"). The data is organized into 17 different newsgroups and 4 main topics. This data set is 95.97% sparse.

2. **Single cell RNA sequencing (scRNA seq) Data**: Collection of 23,226 genes across 5,053 transcriptomes from 10 distinct regions of murine juvenile and adult central nervous system (Marques et al., 2016). All cells were profiled using the Fluidigm C1 system and sequenced on an Illumina HiSeq 2000 instrument. This data set is 87.57% sparse.

5.2. Parameter Settings

For CDP, we need to set the number of iterations, the Dirichlet process concentration parameter γ, and the Dirichlet distribution hyperprior β. Note that both our text and biological data are discrete counts so we assume a multinomial base distribution. If we had continuous data we would instead assume a Gaussian base distribution (or another continuous distribution) and set the values for a Normal–Inverse–Wishart hyperprior. The hyperparameters for ϕ^r, ϕ^c and θ are set to zero by default. We set all concentration parameters and hyperpriors to be small to obtain larger cluster sizes. Table 1 shows the parameter values for the two real data sets. We did not include the two β hyperparameters or the λ hyperparameter in the table since we set those values to zero. In practice, if one has strong prior knowledge regarding a row or column element, setting a value greater than zero for those hyperparameters will result in a more accurate clustering. However, we are doing strictly exploratory work for this paper.

Table 1. Parameter settings for the DPMM part of CDP on two data sets.

Data set	Row/Col	Iterations	γ	β
Newsgroups	Row	1000	10	1
	Col	1000	100	1
scRNA Seq	Row	500	10	0.1
	Col	500	10	1

5.3. Results for Synthetic Data

Results for the four synthetic data cases are provided in Table 2. In each of the cases considered, plaid, DT2B, and CDP exhibit the highest accuracy in bicluster estimation as measured by the Jaccard score. We also tested the spectral method, but the accuracy was so low we excluded it from the table. In cases 2, 3, and 4, CDP outperforms all other methods, and gives substantially better performance in the most complicated setting (case 4), with a mean Jaccard similarity of 0.756, compared to DT2B with a mean score of 0.522. CDP also exhibits lower variance over repeated simulations compared to DT2B. Only in the simplest setting of a single bicluster (case 1) does DT2B show better mean similarity score, with 0.806 for DT2B compared to 0.69 for CDP. However, DT2B shows high variance in the accuracy of its estimates over repeated runs in this case, whereas CDP shows lower variance over all scenarios. Together, these

Table 2. Comparison of mean (standard deviation) Jaccard similarity scores for various biclustering algorithms on the simulated data sets.

CASE	PLAID	C & C	BIMAX	DT2B	CDP
1	0.133 (0.06)	0.16 (0.00)	0.141 (0.005)	0.806 (0.204)	0.69 (0.088)
2	0.862 (0.265)	0.016 (0.000)	0.098 (0.021)	0.889 (0.118)	0.985 (0.081)
3	0.172 (0.087)	0.048 (0.000)	0.105 (0.012)	0.236 (0.032)	0.25 (0.014)
4	0.316 (0.343)	0.008 (0.000)	0.101 (0.014)	0.522 (0.087)	0.756 (0.033)

results suggest that CDP is practical for bicluster extraction, and may be significantly more accurate compared to existing methods.

5.4. Simulation Runtime Comparisons

The DT2B method is conceptually similar to CDP, but requires selecting a maximum number of row and column clusters to determine the parameters of the underlying LDA models. In general, DT2B is most efficient and accurate when the maximum number of row clusters (K_r) and column clusters (K_c) are set to the true number of row and column clusters respectively, but these values will be unknown in practice. DT2B runs in $O(NK_rK_c)$ time (Rugeles et al., 2017), thus setting K_r and K_c to the number of rows and columns respectively may be computationally prohibitive for applications to single cell analysis and other large data settings. Table 3 shows the runtime of DT2B for different choices of K_r and K_c on a simulated data set, compared to CDP.

Table 3. Comparison of runtimes for CDP and DT2B with various choices of (K_r, K_c) on a simulated data set (case 2). Runtimes for DT2B scale linearly in both K_r and K_c.

Method	Mean Jaccard (s.d.)	Runtime (s)
CDP	0.96 (0.02)	13.22
DT2B(5, 5)	0.41 (0.11)	4.23
DT2B(10, 10)	0.94 (0.13)	12.85
DT2B(25, 25)	0.98 (0.01)	73.45

5.5. Results for Text Data

Biclustering text data from a document corpus allows for identification of document-word combinations with high co-occurrence. Extracted biclusters represent combinations of words and documents that form a (latent) topic. This is distinguished from traditional LDA topic modeling in that LDA does not cluster documents directly, and words which co-occur across many documents may be clustered even if the shared vocabulary of those documents is small overall. Instead, a biclustering such as CDP encourages heavy topics which exhibit high co-occurrence of words across documents and documents across words.

The condensed version of the 20 Newsgroup data set is organized into 17 different newsgroups corresponding to four main topics: comp (e.g. computing, graphics), rec (e.g. recreational, sports), sci (e.g. medicine, electronics, space) and talk (e.g. politics, guns), and two smaller topics: religion and miscellaneous for sale.

CDP found 5 word clusters, 3 news groups, and 3 heavy biclusters. There is generally no ground truth for biclusters on text data, and due to the overlapping nature of this "netnews" data set, we chose to evaluate the biclusters by visual inspection. We present Table 4 showing the words with the highest co-occurrences across documents. The first grouping is predominantly about space and political topics, while the second grouping is comprised of recreational, religious and medical topics. The third heavy bicluster consists of computational topics.

Table 4. Selection of the top six words with the highest co-occurrence values across the documents.

TOPIC 1	TOPIC 2	TOPIC 3
MARS	CHILDREN	FTP
SOLAR	DISEASE	FANS
TECHNOLOGY	BIBLE	FILES
SATELLITE	BASEBALL	FORMAT
SHUTTLE	CANCER	FACT
PRESIDENT	PATIENTS	GAMES

5.6. Results for Single cell RNA Sequencing Data

Biclustering scRNA seq data is commonly used to define developmental stages based solely on the transcriptome in addition to accounting for variation in the data, and identifying biologically important genes and their signatures for each cell stage. Each bicluster is an association between groups of cell stages and their genetic drivers.

A key contribution of CDP is the ability to identify the cell stages and their genetic drivers without having to find highly expressed genes *á priori*. Furthermore, cell stages are dynamic in time and a probabilistic clustering assignment allows us to capture part of this dynamic without a true time series model. This contribution is a vital reason as to why we utilize the MAP to determine the most probable number of clusters instead of running the two DPMMS until they converge on a single value.

We apply CDP to the scRNA seq data set in (Marques et al., 2016). The authors performed a biclustering analysis using BackSPIN (Zeisel et al., 2015) and found 13 cell clusters.

CDP found 7 gene clusters, 12 cell clusters, and 4 strong biclusters. Like text data, there is generally no ground truth for biclusters on scRNA seq data. We evaluate our method using the PANTHER classification system and tools (Mi et al., 2018) (Thomas et al., 2006) (Mi et al., 2019), and also compare it to (Marques et al., 2016)'s results.

The four biclusters with the strongest co-occurrence values consist of myelin-forming oligodendrocytes (MFOL2), and several stages of mature oligodendrocytes (MOL5, MOL4 and MOL3). Biclusters with weaker co-occurrence values consist of newly formed oligodendrocytes (NFOL1) and oligodendrocyte precursor cells (OPC). The oligodendrocyte precursor cells can differentiate into newly formed oligodendrocytes, which produce myelin and continue maturing. Since there are multiple stages of maturation, the composition of the strong biclusters are expected and are corroborated by (Marques et al., 2016). The majority of the oligodendrocyte cells are no longer precursor cells or newly formed; they are in differing stages of maturation.

Furthermore, CDP shows that the oligodendrocyte classes also correspond to different regions of the central nervous system. For example, oligodendrocytes classified as MFOL2 are also found in abundance in the substantia nigra ventral tegmental (SN-VTA) and hypothalamus regions of the central nervous system. Likewise, oligodendrocytes classified as MOL5 are found in abundance in the dorsal horn.

With respect to the genes, CDP did not find distinct gene groupings. However, CDP did find two overlapping groupings and multiple groupings with weak co-occurrence values. Using PANTHER, we find that the two overlapping groupings are strongly affiliated with binding, particularly enzymatic binding, and catalytic activity. One group is more involved with cytoskeletal protein binding, and at a higher cellular level, is associated with cellular response to stimulus and cellular metabolic processes. The second group is more involved with signaling receptor binding, and with cell component organization and signal transduction at a higher level. Genes associated with other biological processes such as the lipid metabolic process or the multicellular organismal process are in the biclusters with weaker co-occurrence values.

6. Discussion

In this paper, we presented a novel, non-parametric probabilistic biclustering method designed to address the challenges of model and parameter selection required by competing methods. By utilizing two infinite mixture models and calculating their mutual dependence, we are able to estimate the number of biclusters strictly from the data and prior, and identify the biclusters without strong modeling assumptions.

CDP currently requires hyperparameter specifications, but putting a prior on these hyperparameters may improve accuracy without the need for running the model over a range of parameters. Furthermore, CDP is focused on partitioning discrete data since text and scRNA seq data naturally have count data. However, other applications such as audio retrieval do not. CDP has the ability to model continuous data as well by changing the multinomial base distribution to a Normal–Inverse–Wishart base distribution and modifying the mutual dependence calculation steps.

Simulation results suggest CDP significantly improves upon DT2B and current standard methods, with more accurate estimation of biclusters, and lower variance estimates. Experimental results on real data with high sparsity ($> 85\%$) demonstrate that CDP is able to extract meaningful heavy biclusters. In single cell analyses, this advantage is particularly useful as the data is extremely sparse and noisy.

As a probabilistic model leveraging DPMMs for bicluster estimation, CDP can easily be extended to include additional structure and assumptions. For instance, in the context of single cell analysis, known results on gene networks may be incorporated through the DPMM priors. Furthermore, by choosing continuous DPMM base measures G^r, G^c, CDP can be applied for biclustering a matrix of continuous values, providing an important advantage over DT2B, which can only accommodate discrete values.

7. Data and Software

All data sets are publicly available. The condensed 20 Newsgroup data set is available on Sam Roweis's website (Roweis). The scRNA seq data set is part of the Hemberg lab's collection of publicly available scRNA seq data sets (Kiselev & Hemberg, 2017) as a SingleCellExperiment Bioconductor S4 class (Lun & Risso, 2019).

We removed rows and columns where the entire vector consisted of zeros. For the scRNA seq data set, we also combined the counts of genes that had been split into multiple entries based on loci position.

Source code for CDP can be found at https://github.com/micnngo/CDP. DPMMs were run using the 'exposed_parr' branch of DPMMSubClusters (Dinari et al., 2019). The main CDP script is written in R with a wrapper for Julia and C++. Plaid, Cheng and Church, Spectral and BiMax algorithms were run using the package 'biclust' in R (Kaiser et al., 2018). The source code for DT2B is available on Github (Rugeles et al., 2017).

Acknowledgements

This work was supported by NSF grants DMS1936833 and DMS1763272, Simons Foundation grant (594598, QN), and NIH grant R01MH115697. We also thank Or Dinari for sharing a modification to his DPMMSubClusters package.

References

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. URL http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf.

Chang, J. and Fisher III, J. W. Parallel sampling of dp mixture models using sub-clusters splits. In *Proceedings of the Neural Information Process Systems (NIPS)*, pp. 620–628, 2013.

Cheng, Y. and Church St, G. M. Biclustering of Expression Data. 8:93–103, 2000. URL www.aaai.org.

Dhillon, I. S. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 269–274. ACM, 2001.

Dhillon, I. S., Mallela, S., and Modha, D. S. Information-Theoretic Co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003.

Dinari, O., Yu, A., Freifeld, O., and Fisher III, J. W. Distributed MCMC Inference in Dirichlet Process Mixture Models Using Julia. In *19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, 2019. doi: 10.1109/CCGRID.2019.00066. URL https://ieeexplore.ieee.org/document/8752661.

Hartigan, J. A. Direct Clustering of a Data Matrix. *Journal of the American Statistical Association*, 67(337):123, mar 1972. ISSN 01621459. doi: 10.2307/2284710.

Kaiser, S., Santamaria, R., Khamiakova, T., Sill, M., Theron, R., Quintales, L., Leisch, F., and De Troyer., E. *biclust: BiCluster Algorithms*, 2018. URL https://CRAN.R-project.org/package=biclust. R package version 2.0.1.

Kiselev, V. and Hemberg, M. Collection of public scrna-seq datasets used by our group, 2017. URL https://hemberg-lab.github.io/scRNA.seq.datasets/.

Kluger, Y., Basri, R., Chang, J. T., and Gerstein, M. Spectral biclustering of microarray data: Coclustering genes and conditions. 2003. ISSN 10889051. doi: 10.1101/gr.648603.

Lazzeroni, L. and Owen, A. Plaid Models for Gene Expression Data. *Statistica Sinica*, 12(1):61–86, 2002. doi: 10.1007/s13398-014-0173-7.2.

Lun, A. and Risso, D. *SingleCellExperiment: S4 Classes for Single Cell Data*, 2019. R package version 1.8.0.

Marques, S., Zeisel, A., Codeluppi, S., van Bruggen, D., Mendanha Falcão, A., Xiao, L., Li, H., Häring, M., Hochgerner, H., Romanov, R. A., Gyllborg, D., Muñoz Manchado, A., La Manno, G., Lönnerberg, P., Floriddia, E. M., Rezayee, F., Ernfors, P., Arenas, E., Hjerling-Leffler, J., Harkany, T., Richardson, W. D., Linnarsson, S., and Castelo-Branco, G. Oligodendrocyte heterogeneity in the mouse juvenile and adult central nervous system. *Science (New York, N.Y.)*, 352(6291):1326–1329, 6 2016. ISSN 1095-9203. doi: 10.1126/science.aaf6463.

Mi, H., Muruganujan, A., Ebert, D., Huang, X., and Thomas, P. D. PANTHER version 14: more genomes, a new PANTHER GO-slim and improvements in enrichment analysis tools. *Nucleic Acids Research*, 47:419–426, 2018. doi: 10.1093/nar/gky1038. URL http://geneontology.

Mi, H., Muruganujan, A., Huang, X., Ebert, D., Mills, C., Guo, X., and Thomas, P. D. Protocol Update for large-scale genome and gene function analysis with the PANTHER classification system (v.14.0). *Nature Protocols*, 14(3):703–721, mar 2019. doi: 10.1038/s41596-019-0128-8.

Neal, R. M. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 2000. ISSN 15372715. doi: 10.1080/10618600.2000.10474879.

Padilha, V. A. and Campello, R. J. G. B. A systematic comparative evaluation of biclustering techniques. *BMC Bioinformatics*, 18(1):55, 12 2017. ISSN 1471-2105. doi: 10.1186/s12859-017-1487-1. URL http://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1487-1.

Preli, A., Bleuler, S., Zimmermann, P., Wille, A., Bhlmann, P., Gruissem, W., Hennig, L., Thiele, L., and Zitzler, E. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129, 02 2006. ISSN 1367-4803. doi: 10.1093/bioinformatics/btl060. URL https://doi.org/10.1093/bioinformatics/btl060.

Pritchard, J. K., Stephens, M., and Donnelly, P. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

Roweis, S. Data for matlab hackers. URL https://cs.nyu.edu/~roweis/data.html.

Rugeles, D., Zhao, K., Gao, C., Dash, M., and Krishnaswamy, S. Biclustering: An application of Dual Topic Models. *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 453–461, 2017. doi: 10.1137/1.9781611974973.51. URL http://www.siam.org/journals/ojsa.php.

Sudderth, E. B. and Freeman, W. T. *Graphical Models for Visual Object Recognition and Tracking*. PhD thesis, Massachusetts Institute of Technology, 2006.

Thomas, P. D., Kejariwal, A., Guo, N., Mi, H., Campbell, M. J., Muruganujan, A., and Lazareva-Ulitsky, B. Applications for protein sequence-function evolution data: mRNA/protein expression analysis and coding SNP scoring tools. *Nucleic Acids Research*, 34(Suppl_2): W645–W650, 2006. doi: 10.1093/nar/gkl229. URL http://www.pantherdb.org/tools.

Xie, J., Ma, A., Zhang, Y., Liu, B., Cao, S., Wang, C., Xu, J., Zhang, C., and Ma, Q. QUBIC2: a novel and robust biclustering algorithm for analyses and interpretation of large-scale RNA-Seq data. *Bioinformatics*, sep 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz692. URL https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/btz692/5567116.

Zeisel, A., Moz-Manchado, A. B., Codeluppi, S., Lönnerberg, P., Manno, G. L., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., Rolny, C., Castelo-Branco, G., Hjerling-Leffler, J., and Linnarsson, S. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 347 (6226):1138–1142, mar 2015. ISSN 10959203. doi: 10.1126/science.aaa1934.