Electronic Journal of Statistics
Vol. 14 (2020) 731–768
ISSN: 1935-7524
https://doi.org/10.1214/19-EJS1667

# Profile likelihood biclustering

**Cheryl Flynn**

*AT&T Labs Research*
*New York, NY*
*e-mail:* cflynn@research.att.com

**and**

**Patrick Perry**

*Oscar Health*
*New York, NY*
*e-mail:* patperry@gmail.com

**Abstract:** Biclustering, the process of simultaneously clustering the rows and columns of a data matrix, is a popular and effective tool for finding structure in a high-dimensional dataset. Many biclustering procedures appear to work well in practice, but most do not have associated consistency guarantees. To address this shortcoming, we propose a new biclustering procedure based on profile likelihood. The procedure applies to a broad range of data modalities, including binary, count, and continuous observations. We prove that the procedure recovers the true row and column classes when the dimensions of the data matrix tend to infinity, even if the functional form of the data distribution is misspecified. The procedure requires computing a combinatorial search, which can be expensive in practice. Rather than performing this search directly, we propose a new heuristic optimization procedure based on the Kernighan-Lin heuristic, which has nice computational properties and performs well in simulations. We demonstrate our procedure with applications to congressional voting records, and microarray analysis.

**MSC 2010 subject classifications:** Primary 62-07; secondary 62G20.
**Keywords and phrases:** Biclustering, block model, profile likelihood, congressional voting, microarray data.

Received July 2018.

## 1. Introduction

Suppose we are given a data matrix $\boldsymbol{X} = [X_{ij}]$, and our goal is to cluster the rows and columns of $\boldsymbol{X}$ into meaningful groups. For example, $X_{ij}$ can indicate whether or not user $i$ has an interest in product $j$, and our goal is to segment users and products into relevant subgroups. Alternatively, $X_{ij}$ could be the log activation level of gene $j$ in patient $i$; our goal is to seek groups of patients with similar genetic profiles, while at the same time finding groups of genes with similar activation levels. The general simultaneous clustering problem is known by many names, including direct clustering [24], block modeling [5], biclustering [35], and co-clustering [15].

Empirical results from a broad range of disciplines indicate that biclustering is useful in practice. Ungar and Foster [49] and Hofmann [25] found that biclustering helps identify structure in collaborative filtering contexts with heterogeneous users and sparsely-observed preferences. Eisen et al. [16] used microarray data to simultaneously cluster genes and conditions, finding that genes with similar functions often cluster together. Harpaz et al. [23] applied biclustering methods to a Food and Drug Adminstration report database, identifying associations between certain active ingredients and adverse medical reactions. Several other applications of biclustering exist [10, 19, 30, 29]; Madeira and Oliveira [33] give a comprehensive survey.

Practitioners interested in biclustering have used a variety of different algorithms to achieve their results. Clearly, many of these algorithms work well in practice, but they are often ad-hoc, and there are no rigorous guarantees as to their performance. Without these guarantees practicioners cannot be assured that their discoveries from biclustering will generalize or be reproducible; collecting more data may lead to completely different clusters.

There are two approaches to evaluating the theoretical performance of these procedures. The first is to define a higher-level learning task, and evaluate procedures using a task-dependent measure of generalization performance [44, 45]. We instead consider the alternative approach, which is to consider the problem purely as an unsupervised learning task. In this case, the procedure is evaluated based on the identified biclusters, where a reasonable goal is consistent biclustering.

Our first contribution is to formalize biclustering as an estimation problem. We do this by introducing a probabilistic model for the data matrix $\boldsymbol{X}$, where up to permutations of the rows and columns, the expectation of $\boldsymbol{X}$ has a block structure. Next, we make distributional assumptions on the elements of $\boldsymbol{X}$ and derive a clustering objective function via profile likelihood [37]. Finally, we show that the maximum profile likelihood estimator performs well when the distributional assumptions are not satisfied, in the sense that it is still consistent. To our knowledge, this is the first general consistency result for a biclustering algorithm.

Unfortunately, it is computationally intractable to compute the maximum profile likelihood estimator. It is for this reason that Ungar and Foster [49], who used a similar probabilistic model for the data matrix, dismissed likelihood-based approaches as computationally infeasible. Our second contribution, then, is to propose a new approximation algorithm for finding a local maximizer of the biclustering profile likelihood. Our algorithm is based on the Kernigham-Lin heuristic [28], which was employed by Newman [38] for clustering network data. This is a greedy optimization procedure, so it is not guaranteed to find a global optimum. To mitigate this, we run the fitting procedure repeatedly with many random initializations; as we increase the number of initializations, the probability that the algorithm finds the global optimum increases. We show that this procedure has low computational complexity and it performs well in practice.

Our work was inspired by clustering methods for symmetric binary networks.

In that context, $\boldsymbol{X}$ is an $n$-by-$n$ symmetric binary matrix, and the clusters for the rows of $\boldsymbol{X}$ are the same as the clusters for the columns of $\boldsymbol{X}$. Bickel and Chen [7] used methods similar to those used for proving consistency of M-estimators to derive results for network clustering when $n$ tends to infinity. This work was later extended by Choi et al. [13], who allow the number of clusters to increase with $n$; Zhao et al. [53], who allow for nodes not belonging to any cluster; and Zhao et al. [54], who incorporate individual-specific effects. In parallel to this work, theoretical and computational advancements have been made for community detection in symmetric networks using pseudo-likelihood methods [3], variational methods [14, 9, 6], belief propagation [36], spectral clustering [42, 17, 31, 27], and semidefinite programming [4, 22]. We refer the reader to Zhao [52] and Abbe [1] for recent surveys. In the context of biclustering, Rohe and Yu [43] study spectral methods for unsymmetric binary networks; Ames [2] study recovering the correct partitioning of bipartite graphs into disjoint subgraphs; Gao et al. [18] study problems related to biclustering, but focus on the recovery of the mean matrix rather than recovery of the correct biclustering; Choi and Wolfe [12] study community detection in binary arrays when the blockmodel assumption is misspecified; Mariadassou and Matias [34] study latent and stochastic block models where the row and column cluster assignments are treated as random variables; and Razaee et al. [41] study the problem of matched community detection in bipartite networks with node covariates.

In our report we have extended methods originally developed for an extremely specialized context (symmetric binary networks) to handle clustering for arbitrary data matrices, including non-symmetic networks and real-valued data entries. Using standard conditions, we have been able to generalize the Bickel and Chen [7] results beyond Bernoulli random variables. To our knowledge, this is the first time methodologies for binary networks have been used to study general biclustering methods. Notably, our extensions can handle a variety of data distributions, and they can handle both dense and sparse data matrices.

The main text of the paper is organized as follows. First Section 2 describes the theoretical setup and Section 3 presents our main result with a heuristic proof. Then, Section 4 describes the formal theoretical framework and states the rigorous consistency results. Next, Section 5 presents our approximation algorithm. Using this algorithm, Section 6 corroborates the theoretical findings through a simulation study, and Section 7 presents applications to a microarray and a congressional voting dataset. Section 8 presents some concluding remarks. The appendices include additional proofs, empirical results, and an application to a movie review dataset.

## 2. Estimation problem and criterion functions

Our first task is to formalize biclustering as an estimation problem. To this end, let $\boldsymbol{X} = [X_{ij}] \in \mathbb{R}^{m \times n}$ be a data matrix. We follow the network clustering literature and posit existence of $K$ *row classes* and $L$ *column classes*, such that the mean value of entry $X_{ij}$ is determined solely by the classes of row $i$ and column $j$. That is, there is an unknown row class membership vector $\boldsymbol{c} \in K^m$, an unknown column class membership vector $\boldsymbol{d} \in L^n$, and an unknown mean matrix $\boldsymbol{M} = [\mu_{kl}] \in \mathbb{R}^{K \times L}$ such that

$$\operatorname{E} X_{ij} = \mu_{c_i d_j}. \tag{2.1}$$

We refer to model (2.1) as a *block model*, after the related model for undirected networks proposed by Holland et al. [26]. Under the assumptions of the block model, biclustering the rows and columns of the data matrix is equivalent to estimating $\boldsymbol{c}$ and $\boldsymbol{d}$.

Not all block models give rise to well-defined estimation problems. To ensure that $K$ and $L$ are well-defined, we require that each class has at least one member, and that no two classes have the same mean vector. Formally, define row class proportion vector $\boldsymbol{p} \in \mathbb{R}^K$ with element $p_a = m^{-1} \sum_i \operatorname{I}(c_i = a)$ equal the proportion of nodes with row class $a$. Also, define column class proportion vector $\boldsymbol{q} \in \mathbb{R}^L$ with element $q_b = n^{-1} \sum_j \operatorname{I}(d_j = b)$ equal to the proportion of nodes with column class $b$. We require that every element of $\boldsymbol{p}$ and $\boldsymbol{q}$ be nonzero. To ensure that the mean vectors of the row classes are distinct, we require that no two rows of $\boldsymbol{M}$ are identical. Similarly, we require that no two columns of $\boldsymbol{M}$ are identical.

We estimate the clusters by assigning labels to the rows and columns of $\boldsymbol{X}$, codified in vectors $\boldsymbol{g} \in K^m$ and $\boldsymbol{h} \in L^n$. Ideally, $\boldsymbol{g}$ and $\boldsymbol{h}$ match $\boldsymbol{c}$ and $\boldsymbol{d}$. Note we are assuming that the true numbers of row and column clusters, $K$ and $L$, are known, or they have been correctly estimated by some model selection procedure. We measure the performance of a particular label assignment through the corresponding confusion matrix. Specifically, for row and column label assignments $\boldsymbol{g}$ and $\boldsymbol{h}$, define normalized confusion matrices $\boldsymbol{C} \in \mathbb{R}^{K \times K}$ and $\boldsymbol{D} \in \mathbb{R}^{L \times L}$ by

$$C_{ak} = \frac{1}{m} \sum_i I(c_i = a, g_i = k), \qquad D_{bl} = \frac{1}{n} \sum_j I(d_j = b, h_j = l).$$

Entry $C_{ak}$ is the proportion of nodes with class $a$ and label $k$; entry $D_{bl}$ is defined similarly. These matrices are normalized so that $\boldsymbol{C}\mathbf{1} = \boldsymbol{p}$ and $\boldsymbol{D}\mathbf{1} = \boldsymbol{q}$ are the class proportion vectors, and $\boldsymbol{C}^T\mathbf{1} = \hat{\boldsymbol{p}}$ and $\boldsymbol{D}^T\mathbf{1} = \hat{\boldsymbol{q}}$ are the label proportion vectors. If $\boldsymbol{C}$ and $\boldsymbol{D}$ are diagonal, then the assigned labels match the true classes. More generally, if $\boldsymbol{C}$ and $\boldsymbol{D}$ can be made diagonal by permuting their columns, then the partition induced by the labels matches the partition induced by the classes. The goal, then, is to find row and column labellings such that $\boldsymbol{C}$ and $\boldsymbol{D}$ are permutations of diagonal matrices.

In practice, we cannot estimate $\boldsymbol{C}$ and $\boldsymbol{D}$ directly, because we do not have knowledge of the true row and column classes. To evaluate the quality of a biclustering, we need a surrogate criterion function. Analogously to Bickel and Chen [7], we employ profile likelihood for this purpose.

In Bickel and Chen's setting, the data are binary, so there is a natural data likelihood which arises from the Bernoulli distribution. Our setting is more general, with $\boldsymbol{X}_{ij}$ allowed to be a count or a continuous measurement, so there are many possible choices for the element densities. We proceed by initially assuming that the elements of $\boldsymbol{X}$ are sampled from distributions in a single-parameter exponential family. Conditional on $\boldsymbol{c}$ and $\boldsymbol{d}$, the elements of $\boldsymbol{X}$ are independent, and entry $X_{ij}$ has density $g(x; \eta_{c_i d_j})$ with respect to some base measure $\nu$, where

$$g(x; \eta) = \exp\{x\eta - \psi(\eta)\};$$

$\psi(\eta)$ is the cumulant generating function, and $\eta_{kl} = (\psi')^{-1}(\mu_{kl})$ is the natural parameter. Later, we will relax the assumption of the specific distributional form.

With labels $\boldsymbol{g}$ and $\boldsymbol{h}$, the complete data log-likelihood is

$$\begin{aligned} l(\boldsymbol{g}, \boldsymbol{h}, \boldsymbol{M}) &= \sum_{k,l} \sum_{i,j} \{X_{ij}\eta_{kl} - \psi(\eta_{kl})\} \operatorname{I}(g_i = k, h_j = l) \\ &= mn \sum_{k,l} \hat{p}_k \, \hat{q}_l \, \{\bar{X}_{kl} \, \eta_{kl} - \psi(\eta_{kl})\}, \end{aligned}$$

where $\hat{p}_k = m^{-1} \sum_i \operatorname{I}(g_i = k)$ and $\hat{q}_l = n^{-1} \sum_j \operatorname{I}(h_j = l)$ are the estimated class proportions and $\bar{X}_{kl} = \{\sum_{i,j} \operatorname{I}(g_i = k, h_j = l)\}^{-1} \sum_{i,j} X_{ij} \operatorname{I}(g_i = k, h_j = l)$ is the estimated cluster mean. We get the profile log-likelihood by maximizing the log-likelihood over the mean parameter matrix $\boldsymbol{M}$:

$$pl(\boldsymbol{g}, \boldsymbol{h}) = \sup_{\boldsymbol{M}} l(\boldsymbol{g}, \boldsymbol{h}, \boldsymbol{M}) = mn \sum_{k,l} \hat{p}_k \, \hat{q}_l \, \psi^*(\bar{X}_{kl}),$$

where $\psi^*(x) = \sup_\eta \{x\eta - \psi(\eta)\}$ is the convex conjugate of $\psi$. We refer to $\psi^*$ as the relative entropy function since $\psi^*(\mu)$ is equal to the Kullback-Leiber divergence of the base measure $\nu$ from the distribution in the exponential family with mean $\mu$ [8].

Following the above derivation, a natural criterion for the quality of labeling $(\boldsymbol{g}, \boldsymbol{h})$ is the profile log-likelihood $pl(\boldsymbol{g}, \boldsymbol{h})$. In the sequel, we consider a far more general setting. We consider criterion functions of the form

$$F(\boldsymbol{g}, \boldsymbol{h}) = \sum_{k,l} \hat{p}_k \, \hat{q}_l \, f(\bar{X}_{kl}), \tag{2.2}$$

where $f$ is any smooth convex function. Following the derivation above, we refer to $F$ as a profile likelihood and we refer to $f$ as the corresponding relative entropy function. However, we do not assume that likelihood has been correctly specified. In particular, the elements of $\boldsymbol{X}$ can have arbitrary distributional forms under the assumptions of the block model (2.1), not necessarily belonging to any exponential family. We explicitly allow for heteroscedasticity and distributional misspecification. We show that under mild technical conditions, the maximizer of $F$ is a consistent estimator of the true row and column classes.

## 3. Heuristic justification

In Section 2, we defined a formal biclustering estimation problem and we motivated a class of criterion functions for this problem based on profile likelihood.

In this section, we investigate the behavior of the criterion functions. In particular, we outline a heuristic argument which shows that the row and column labels found by maximizing these criterion functions are good estimates of the true row and column classes. Formal statements of the results and their proofs are given in Section 4 and Appendix A.

As noted in Section 1, the main thrust of our theoretical results are similar to that used in the literature on clustering for symmetric binary networks initiated by Bickel and Chen [7] and extended by Choi et al. [13], Zhao et al. [53] and Zhao et al. [54]. The main point of departure from this previous work are that we work with arbitrary data modalities instead of symmetric binary matrices.

Let $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ be a data matrix drawn from an identifiable block model (2.1) with row and column classes $\boldsymbol{c} \in K^m$ and $\boldsymbol{d} \in L^n$ and mean matrix $\boldsymbol{M} \in \mathbb{R}^{K \times L}$. Let $\boldsymbol{p}$, and $\boldsymbol{q}$ be as defined in Section 2. For any row and column labeling $\boldsymbol{g}$ and $\boldsymbol{h}$, let $\hat{\boldsymbol{p}}$, $\hat{\boldsymbol{q}}$, and $\bar{\boldsymbol{X}}$ be the corresponding estimates of $\boldsymbol{p}$, $\boldsymbol{q}$, and $\boldsymbol{M}$, and let $\boldsymbol{C}$ and $\boldsymbol{D}$ be the confusion matrices. Let $F$ be a profile likelihood criterion function as in (2.2) with corresponding relative entropy function $f$, assumed to be smooth and strictly convex.

We now outline a series of results which show that the maximizers of $F$ are good estimates of the true row and column classes.

**Proposition 3.1.** *The criterion function $F$ is uniformly close to a "population criterion function" $G$ which only depends on the confusion matrices.*

If $n$ and $m$ are large, then for any choice of $\boldsymbol{g}$ and $\boldsymbol{h}$, the estimated cluster mean $\bar{X}_{kl}$ will be close to $E_{kl}$, the average value of $\operatorname{E} X_{ij}$ over the block defined by labels $k$ and $l$. This quantity can be computed in terms of the confusion matrices as

$$E_{kl} = \frac{\sum_{i,j} \sum_{a,b} \mu_{ab} \operatorname{I}(c_i = a, g_i = k) \operatorname{I}(d_j = b, h_j = l)}{\sum_{i,j} \operatorname{I}(g_i = k, h_j = l)} = \frac{[\boldsymbol{C}^T \boldsymbol{M} \boldsymbol{D}]_{kl}}{[\boldsymbol{C}^T \boldsymbol{1}]_k [\boldsymbol{D}^T \boldsymbol{1}]_l}.$$

By applying Bernstein's inequality, one can show that $E_{kl}$ is close to $\bar{X}_{kl}$ uniformly over all choices of $\boldsymbol{g}$ and $\boldsymbol{h}$. Thus, we get the population criterion function by replacing $\bar{X}_{kl}$ with $E_{kl}$.

For each non-negative vector $\boldsymbol{t} \in \mathbb{R}_+^N$ define $\mathcal{C}_{\boldsymbol{t}}$ to be the set of $N \times N$ normalized confusion matrices with fixed row sums: $\mathcal{C}_{\boldsymbol{t}} = \{\boldsymbol{W} \in \mathbb{R}_+^{N \times N} : \boldsymbol{W}\boldsymbol{1} = \boldsymbol{t}\}$. The population version of $F$ is a function of the row and column confusion matrices, $G : \mathcal{C}_{\boldsymbol{p}} \times \mathcal{C}_{\boldsymbol{q}} \to \mathbb{R}$, with

$$G(\boldsymbol{C}, \boldsymbol{D}) = \sum_{k,l} [\boldsymbol{C}^T \boldsymbol{1}]_k \, [\boldsymbol{D}^T \boldsymbol{1}]_l \, f\Big( \frac{[\boldsymbol{C}^T \boldsymbol{M} \boldsymbol{D}]_{kl}}{[\boldsymbol{C}^T \boldsymbol{1}]_k \, [\boldsymbol{D}^T \boldsymbol{1}]_l} \Big).$$

Since $\bar{X}_{kl}$ is uniformly close to $E_{kl}$, under mild regularity conditions on $f$, the criterion $F(\boldsymbol{g}, \boldsymbol{h})$ is uniformly close to $G(\boldsymbol{C}, \boldsymbol{D})$. Proposition A.2 contains a rigorous statement of this result.

**Proposition 3.2.** *The population criterion function $G$ is self-consistent.*

Self-consistency is an important property for any criterion function, which implies that in the absence of noise, the criterion function will be maximized at the truth [48]. In our context, self-consistency means that $G$ is maximized when $\boldsymbol{C}$ and $\boldsymbol{D}$ are permutations of diagonal matrices.

The self-consistency of $G$ follows from the strict convexity of $f$:

$$\begin{aligned} G(\boldsymbol{C}, \boldsymbol{D}) &= \sum_{k,l} [\boldsymbol{C}^T \mathbf{1}]_k \, [\boldsymbol{D}^T \mathbf{1}]_l \, f\Big( \frac{[\boldsymbol{C}^T \boldsymbol{M} \boldsymbol{D}]_{kl}}{[\boldsymbol{C}^T \mathbf{1}]_k \, [\boldsymbol{D}^T \mathbf{1}]_l} \Big) \\ &\leq \sum_{k,l} \sum_{a,b} C_{ak} D_{bl} f(\mu_{ab}) \\ &= \sum_{a,b} p_a \, q_b \, f(\mu_{ab}). \end{aligned}$$

If $\boldsymbol{M}$ has no two identical rows and no two identical columns, then exact equality holds only when $\boldsymbol{C}$ and $\boldsymbol{D}$ are permutations of diagonal matrices. Thus, $G$ is maximized when the row and column class partitions match the label partitions. Proposition A.3 gives a refined self-consistency result with a more complete characterization of the behavior of $G$ near its maxima.

**Proposition 3.3.** *Under enough regularity, the maximizer of the criterion function $F$ is close to the true row and column class partition.*

This is a direct consequence of Propositions 3.1 and 3.2. The criterion $F$ is uniformly close to the population criterion $G$, and $G$ is maximized at the true class partitions. Thus, the maximizer of $F$ is close to the maximizer of $G$. Importantly, Proposition 3.3 does not require any distributional assumptions on the data matrix $\boldsymbol{X}$ beyond its expectation satisfying the block model. In particular this result can be applied to binary matrices, count data, and continuous data. Theorems 4.1 and 4.2 contain precise statements analogous to Proposition 3.3.

## 4. Rigorous theoretical results

Here we provide formal statements of the main result from Section 3. The proofs of these results are contained in Appendix A.

We work in an asymptotic framework, where the dimensions of the data matrix tend to infinity. Let $\boldsymbol{X}_n \in \mathbb{R}^{m \times n}$ be a sequence of data matrices indexed by $n$, with $m = m(n)$ and $m(n) \to \infty$ as $n \to \infty$. We will also suppose that $n/m \to \gamma$ for some finite constant $\gamma > 0$; this assumption is not essential, but it simplifies the assumption and result statements.

Suppose that for each $n$ there exists a row class membership vector $\boldsymbol{c}_n \in K^m$ and a column class membership vector $\boldsymbol{d}_n \in L^n$. We assume that there exist vectors $\boldsymbol{p} \in \mathbb{R}^K$ and $\boldsymbol{q} \in \mathbb{R}^L$ such that $\hat{p}_k(\boldsymbol{c}) \to p_k$ and $\hat{q}_l(\boldsymbol{d}) \to q_l$ as $n \to \infty$ almost surely for all $k$ and $l$; this assumption is satisfied, for example, if the class labels are independently drawn from a multinomial distribution. When there is no ambiguity, we omit the subscript $n$.

We define the mean matrix $\boldsymbol{M} = [\mu_{kl}] \in \mathbb{R}^{K \times L}$ as in Section 3, but allow it to possibly vary with $n$. To model sparsity in $\boldsymbol{X}$, we allow $\boldsymbol{M}$ to tend to $\boldsymbol{0}$. To avoid degeneracy, we suppose that there exists a sequence $\rho$ and a fixed matrix $\boldsymbol{M}_0 \in \mathbb{R}^{K \times L}$ such that $\rho^{-1}\ \boldsymbol{M}\ \to\ \boldsymbol{M}_0$. Denote by $\mathcal{M}_0 \in \mathbb{R}$ the convex hull of the entries of $\boldsymbol{M}_0$ defined as

$$\mathcal{M}_0 \coloneqq \left\{ \sum_{k=1}^{K} \sum_{l=1}^{L} \lambda_{kl} [\boldsymbol{M}_0]_{kl} : \sum_{k=1}^{K} \sum_{l=1}^{L} \lambda_{kl} = 1, \lambda_{ij} \in [0,1] \right\}.$$

Let $\mathcal{M}$ be a neighborhood of $\mathcal{M}_0$.

To adjust for the sparsity, we redefine the criterion and population criterion functions as

$$F(\boldsymbol{g}, \boldsymbol{h}) = \sum_{k,l} \hat{p}_k \hat{q}_l f(\rho^{-1} \bar{X}_{kl}),$$

$$G(\boldsymbol{C}, \boldsymbol{D}) = \sum_{k,l} [\boldsymbol{C}^T \boldsymbol{1}]_k [\boldsymbol{D}^T \boldsymbol{1}]_l f\Big( \frac{[\boldsymbol{C}^T \boldsymbol{M}_0 \boldsymbol{D}]_{kl}}{[\boldsymbol{C}^T \boldsymbol{1}]_k \, [\boldsymbol{D}^T \boldsymbol{1}]_l} \Big).$$

We discuss these modifications and the role of $\rho$ in Section A.1.4.

We only consider nontrivial partitions; to this end, for $\varepsilon > 0$, define $\mathcal{J}_\varepsilon$, the set of nontrivial labellings as

$$\mathcal{J}_\varepsilon = \{ \boldsymbol{g}, \boldsymbol{h} : \hat{p}_k(\boldsymbol{g}) > \varepsilon, \hat{q}_l(\boldsymbol{h}) > \varepsilon \}.$$

### 4.1. Assumptions

We require the following regularity conditions:

(C1) The biclusters are identifiable: no two rows of the $\boldsymbol{M}_0$ are equal, and no two columns of $\boldsymbol{M}_0$ are equal.

(C2) The relative entropy function is locally strictly convex: $f''(\mu) > 0$ when $\mu \in \mathcal{M}$.

(C3) The third derivative of the relative entropy function is locally bounded: $|f'''(\mu)|$ is bounded when $\mu \in \mathcal{M}$.

(C4) The average variance of the elements is of the same order as $\rho$:

$$\limsup_{n \to \infty} \frac{1}{\rho m n} \sum_{i,j} \mathrm{E}[(X_{ij} - \mu_{c_i d_j})^2] < \infty.$$

(C5) The mean matrix does not converge to zero too quickly:

$$\limsup_{n \to \infty} \rho \sqrt{nm} = \infty.$$

(C6) The elements satisfy a Lindeberg condition: for all $\varepsilon > 0$,

$$\lim_{n \to \infty} \frac{1}{\rho^2 m n} \sum_{i,j} \mathrm{E}[(X_{ij} - \mu_{c_i d_j})^2 \, \mathrm{I}(|X_{ij} - \mu_{c_i d_j}| > \varepsilon \rho \sqrt{mn})] = 0.$$

Condition (C1) is necessary for the biclusters to be identifiable, while (C2) and (C3) are mild regularity conditions on the entropy function.

Condition (C4) is trivially satisfied for dense data and is satisfied for Binomial and Poisson data so long as $\rho^{-1}\boldsymbol{M} \to \boldsymbol{M}_0$. However, this condition cannot handle arbitrary sparsity. For example, if the elements of $\boldsymbol{X}$ are distributed as Negative Binomial random variables, then condition (C4) requires that the product of the mean and the dispersion parameter does not tend to infinity. In other words, for sparse count data, the counts cannot be too heterogeneous.

Condition (C5) places a sparsity restriction on the mean matrix. Zhao et al. [54] used the same assumption to establish weak consistency for network clustering. A variant Lyaponuv's condition [50] implies (C6). That is, if

$$\lim_{n\to\infty} \frac{1}{(\rho\sqrt{mn})^{2+\delta}} \sum_{i,j} \mathrm{E}\,|X_{ij} - \mu_{c_i d_j}|^{2+\delta} = 0$$

for some $\delta > 0$, then (C6) follows. In particular, for dense data ($\rho$ bounded away from zero), uniformly bounded $(2+\delta)$ absolute central moments for any $\delta > 0$ is sufficient. For many types of sparse data, including Bernoulli or Poisson data with $\rho$ converging to zero, (C5) is a sufficient condition for (C6).

**Theorem 4.1.** *Fix any $\varepsilon > 0$ with $\varepsilon < \min_a\{p_a\}$ and $\varepsilon < \min_b\{q_b\}$. Let $(\hat{\boldsymbol{g}}, \hat{\boldsymbol{h}})$ satisfy $F(\hat{\boldsymbol{g}}, \hat{\boldsymbol{h}}) = \max_{\mathcal{J}_\varepsilon} F(\boldsymbol{g}, \boldsymbol{h})$. If conditions (C1)–(C6) hold, then all limit points of $\boldsymbol{C}(\hat{\boldsymbol{g}})$ and $\boldsymbol{D}(\hat{\boldsymbol{h}})$ are permutations of diagonal matrices, i.e. the proportions of mislabeled rows and columns converge to zero in probability.*

Our focus is on the cluster assignments, but, using the methods involved to prove Theorem 4.1, it is possible to show that when the assumptions of Theorem 4.1 are satisfied, then the scaled estimate of the mean, $\rho^{-1}\bar{X}_{kl}$ converges in probability to the population quantity. (This follows from Theorem 4.1 and Lemma A.1.)

Under stronger distributional assumptions, we can use the methods of the proof to establish finite-sample results. For example, if we assume that the elements of $\boldsymbol{X}$ are Gaussian, then the following result holds.

**Theorem 4.2.** *Fix any $\varepsilon > 0$. Let $(\hat{\boldsymbol{g}}, \hat{\boldsymbol{h}})$ satisfy $F(\hat{\boldsymbol{g}}, \hat{\boldsymbol{h}}) = \max_{\mathcal{J}_\varepsilon} F(\boldsymbol{g}, \boldsymbol{h})$. If the elements of $\boldsymbol{X}$ are independent Gaussian random variables with constant variance $\sigma^2$ and conditions (C1)–(C3) hold, then for any $0 < \delta < \min\left\{1, \frac{8c\sigma\max\{K^2,L^2\}}{\tau\varepsilon^2}\right\}$,*

$$\Pr\Big(\big(\boldsymbol{C}(\hat{\boldsymbol{g}}), \boldsymbol{D}(\hat{\boldsymbol{h}})\big) \notin \mathcal{P}_\delta \cap \mathcal{Q}_\delta\Big) \le 2K^{m+1}L^{n+1}\exp\Big\{-\frac{T_n\tau^2\varepsilon^4\delta^2}{256c^2\sigma^2\max\{K^4,L^4\}}\Big\},$$

*where $c = \sup|f'(\mu)|$ for $\mu$ in $\mathcal{M}$,*

$$T_n = \inf_{k,l}\Big\{\sum_i\sum_j \mathrm{I}(g_i = k, h_j = l) | (\boldsymbol{g}, \boldsymbol{h}) \in \mathcal{J}_\varepsilon\Big\}.$$

The proof of this finite-sample result follows the same outline as the asymptotic result; Appendix A.3 gives details.

## 5. Approximation algorithm

Proposition 3.3 shows that the maximizer of the profile log-likelihood $F$ will give a good estimate of the true clusters. Unfortunately, finding this maximizer is an NP-hard problem [47]. Maximizing $F$ is a combinatorial optimization problem with an exponentially-large state space. To get around this, we will settle for finding a local optimum rather than a global one. We present an algorithm for finding a local optimum that, in practice, has good estimation performance.

Our approach is based on the Kernighan-Lin heuristic [28], which Newman [38] used for a related problem, network community detection. After inputting initial partitions for the rows and columns, we iteratively update the cluster assignments in a greedy manner. The algorithm works as follows:

1. Initialize the row and column labels $\boldsymbol{g}$ and $\boldsymbol{h}$ arbitrarily, and compute $F$.
2. Repeat until no local improvement in the profile likelihood is found:
   (a) For each row $i$, determine which of the $K$ possible label assignments for this row is optimal, keeping all other row and column labels fixed. Do not perform this assignment, but record the optimal label and the local improvement to $F$ that would result if this assignment were to be made.

   (b) For each column $j$, determine which of the $L$ possible label assignments for this column is optimal, keeping all other row and column labels fixed. As in step 2a, do not perform this assignment, but record the optimal label and the local improvement to $F$ that would result if this assignment were to be made.

   (c) In order of the local improvements recorded in steps 2a and 2b, sequentially perform the individual cluster reassignments determined in these steps, and record the profile likelihood after each reassignment. Note that these assignments are no longer locally optimal since the labels of many of the rows and columns change during this step. Thus, the profile likelihood could increase or decrease as we move sequentially through the assignments.

   (d) Out of the sequence of cluster assignments considered in step 2c, choose the one that has the highest profile likelihood.

At each complete iteration, a finite sequence of values is considered and we select the cluster assignment with the highest profile likelihood. This allows for efficient consideration of multiple cluster reassignments in each iteration of the algorithm as opposed to re-optimizing after each individual row or column reassignment. The criterion function increases at each complete iteration and we stop when no local improvement in the profile likelihood is found. Thus, the algorithm will converge to a local optimum.

There is no guarantee that the local optimum found by the algorithm will be the global optimum of the objective function $F$. To mitigate this deficiency, we will run the algorithm repeatedly with many different random initializations for $\boldsymbol{g}$ and $\boldsymbol{h}$. Each initialization can give rise to a different local optimum. We choose the cluster assignment with the highest value of $F$ among all local optima found by the procedure. As we increase the number of random initializations, the probability that the global optimum will be in this set will increase. We found that 100–250 initializations seem to suffice in the simulations and data examples. Appendix B provides additional results on the stability of the algorithm.

The main computational bottleneck is updating the value of $F(\boldsymbol{g}, \boldsymbol{h})$ as we update the labels $\boldsymbol{g}$ and $\boldsymbol{h}$. We can do this efficiently by storing and incrementally updating the cluster proportions $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{q}}$, the within-cluster row and column sums

$$R_{il} = \sum_{j=1}^{n} X_{ij}\, \mathrm{I}(h_j = l) \qquad \text{and} \qquad C_{kj} = \sum_{i=1}^{m} X_{ij}\, \mathrm{I}(g_i = k),$$

and the block sums $B_{kl} = \sum_{i=1}^{m} R_{il}\, \mathrm{I}(g_i = k)$. Given the values of these quantities, we can compute the criterion $F(\boldsymbol{g}, \boldsymbol{h})$ with $\mathcal{O}(KL)$ operations.

If we reassign the label of row $i$ from $k$ to $k'$, then it is straightforward to update $\hat{\boldsymbol{p}}$ with $\mathcal{O}(1)$ operations. The values of the within-cluster row sums $R_{il}$ remain unchanged. The new values of the block sums are $B'_{kl} = B_{kl} - R_{il}$ and $B'_{k'l} = B_{k'l} + R_{il}$ for $l = 1, \dots, L$; the other block sums are unchanged. The expensive part of the update is recomputing the within-cluster column sums $C_{kj}$ for row labels $k$ and $k'$ and each column $j$. These computations require $\mathcal{O}(mn)$ operations if $\boldsymbol{X}$ is dense, and $\mathcal{O}(N)$ operations if $\boldsymbol{X}$ is sparse with at most $N$ nonzero elements and $N \geq \max\{m, n\}$.

Overall we must perform $\mathcal{O}(N + KL)$ operations to reassign the label of row $i$. Reassigning the label of column $j$ has the same computational cost. Thus, one loop iteration in step 2 requires $\mathcal{O}\big((m+n)(N+KL)\big)$ operations. For dense data, one iteration requires $\mathcal{O}\big((m+n)(mn+KL)\big)$ operations. We do not have an upper bound on the number of iterations until the algorithm converges, but in our experiments we found that empirically, 25 to 30 iterations suffice. These iteration counts may seem small, but in fact each iteration performs $m$ row label assignments and $n$ column label assignments. The convergence here is not a result of early stopping—we found that after 25–30 iterations, no possible local improvement was possible.

For comparison, a spectral-based biclustering algorithm requires the top singular vectors of the data matrix, which can be calculated in roughly $\mathcal{O}(mn)$ operations using Lanczos or another indirect method [20].

## 6. Empirical evaluation

Here we evaluate the performance of the profile likelihood based biclustering algorithm from Section 5. We simulate data from a variety of regimes, including sparse binary data and dense heavy-tailed continuous measurements. In these settings, we employ the following three relative entropy functions:

$$f_{\text{Bernoulli}}(\mu) = \mu \log \mu + (1 - \mu) \log(1 - \mu), \tag{6.1a}$$

$$f_{\text{Poisson}}(\mu) = \mu \log \mu - \mu, \tag{6.1b}$$

$$f_{\text{Gaussian}}(\mu) = \mu^2/2. \tag{6.1c}$$

We evaluate performance both when the profile likelihood is correctly specified and when the relative entropy function does not match the data distribution.

In our simulations, we report the proportion of misclassified rows and columns by the profile likelihood based method (PL). We initialize partitions randomly, and then run the improvement algorithm from Section 5 until convergence. We use multiple random starting values to minimize the possibility of finding a non-optimal stationary point. Our code for implementing the profile likelihood based biclustering method is available on GitHub (https://github.com/patperry/biclustpl).

We compare our method to three other biclustering algorithms. The first algorithm is a spectral biclustering algorithm, DI-SIM, motivated by a block model similar to ours [43]. The algorithm finds the singular value decomposition of the data matrix $\boldsymbol{X}$, and then applies $k$-means clustering to the top left and right singular vector loadings. The second algorithm, Sparse Biclustering (SBC) [46], assumes the data follows a normal distribution and maximizes the $\ell_1$-penalized log likelihood. The $\ell_1$-penalty allows for sparse biclusters, where the amount of sparsity is controlled by a regularization parameter $\lambda \geq 0$. We implement the method using the R `sparseBC` package. In our simulations, we evaluate the performance of SBC over a grid of $\lambda$ values and report the best case performance of the method. Lastly, we compare against KM [32], which ignores the interactions between the clusters and applies $k$-means separately to the rows and columns of $\boldsymbol{X}$.

In our first simulation, we generate sparse Poisson count data from a block model with $K = 2$ row clusters and $L = 3$ column clusters. We vary the number of columns, $n$, between 200 to 1400 and we take the number of rows to be $m = \gamma n$ for $\gamma \in \{0.5, 1, 2\}$. To assign the true row and column classes $\boldsymbol{c}$ and $\boldsymbol{d}$, we sample independent multinomials with probabilities $\boldsymbol{p} = (0.3, 0.7)$ and $\boldsymbol{q} = (0.2, 0.3, 0.5)$. We choose the matrix of block parameters to be

$$\boldsymbol{M} = [\mu_{ab}] = \frac{b}{\sqrt{n}} \begin{pmatrix} 0.92 & 0.77 & 1.66 \\ 0.17 & 1.41 & 1.45 \end{pmatrix},$$

where $b$ is chosen between 5 and 20; the entries of the matrix were chosen randomly, uniformly on the interval $[0, 2]$. We chose the $1/\sqrt{n}$ scaling so that the data matrix would be sparse, with $\mathcal{O}(\sqrt{n})$ elements in each row. We generate the data conditional on the row and column classes as $X_{ij} \mid \boldsymbol{c}, \boldsymbol{d} \sim \text{Poisson}(\mu_{c_i d_j})$. We run all three methods with 250 random starting values.

Figure 1 presents the average bicluster misclassification rates for each sample size and Table 1 reports the standard deviations. In all of the scenarios considered, biclustering based on the poisson profile likelihood criterion performs at least as well as the other methods. When the relative entropy function is misspecified (using PL-Gaus instead of PL-Pois), biclustering based on the profile likelihood criterion performs at least as well as DI-SIM and KM, but SBC

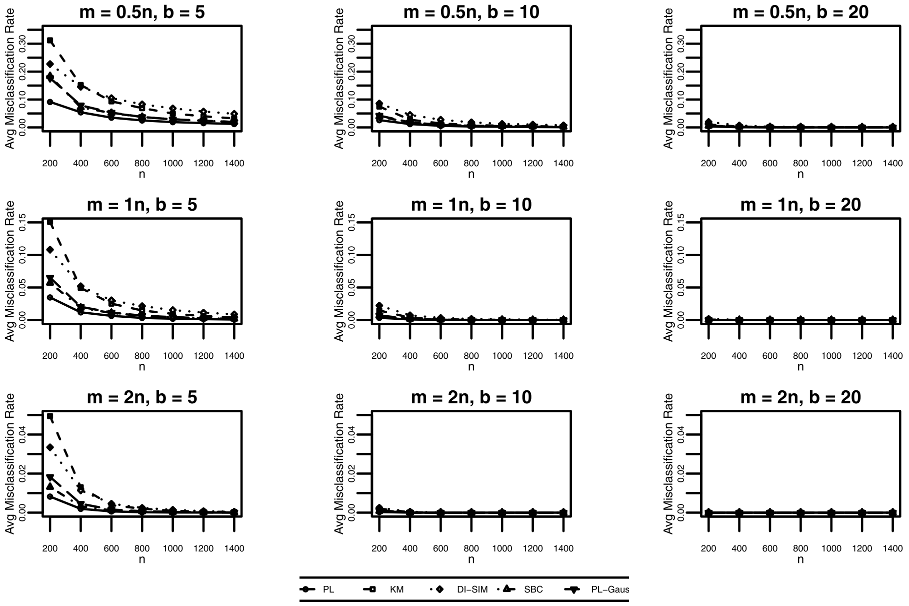

FIG 1. *Average misclassification rates for Poisson example over 100 simulations.*

has a lower average misclassification rate in some scenarios when $n$ is small. As $n$ increases, SBC tends to select a regularization parameter close to zero, and the two methods perform similarly. Moreover by looking at the standard deviations, we see that for the PL methods, the misclassification rate seems to be converging to zero as we increase $n$.

TABLE 1
*Standard deviations for Poisson example over 100 simulations.*

| $m = 0.5n$ | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PL-Pois | | | PL-Norm | | | KM | | | DS | | | SBC | | |
| n | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 |
| 200 | 0.0356 | 0.0147 | 0.0047 | 0.0878 | 0.0239 | 0.0070 | 0.0957 | 0.0438 | 0.0097 | 0.0403 | 0.0286 | 0.0114 | 0.0935 | 0.0204 | 0.0063 |
| 400 | 0.0182 | 0.0064 | 0.0013 | 0.0313 | 0.0091 | 0.0025 | 0.0478 | 0.0145 | 0.0033 | 0.0279 | 0.0144 | 0.0043 | 0.0238 | 0.0089 | 0.0025 |
| 600 | 0.0103 | 0.0034 | 0.0004 | 0.0153 | 0.0045 | 0.0011 | 0.0253 | 0.0068 | 0.0013 | 0.0186 | 0.0083 | 0.0026 | 0.0138 | 0.0043 | 0.0011 |
| 800 | 0.0076 | 0.0024 | 0.0003 | 0.0104 | 0.0039 | 0.0006 | 0.0188 | 0.0050 | 0.0009 | 0.0140 | 0.0069 | 0.0013 | 0.0102 | 0.0040 | 0.0006 |
| 1000 | 0.0054 | 0.0018 | 0.0003 | 0.0076 | 0.0029 | 0.0003 | 0.0122 | 0.0033 | 0.0004 | 0.0111 | 0.0040 | 0.0008 | 0.0076 | 0.0029 | 0.0003 |
| 1200 | 0.0041 | 0.0010 | 0.0001 | 0.0061 | 0.0017 | 0.0003 | 0.0092 | 0.0021 | 0.0003 | 0.0092 | 0.0034 | 0.0007 | 0.0062 | 0.0017 | 0.0003 |
| 1400 | 0.0036 | 0.0009 | 0.0001 | 0.0047 | 0.0014 | 0.0001 | 0.0077 | 0.0016 | 0.0001 | 0.0071 | 0.0023 | 0.0005 | 0.0047 | 0.0014 | 0.0001 |

| $m = n$ | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 |
| 200 | 0.0160 | 0.0039 | 0.0009 | 0.0415 | 0.0069 | 0.0010 | 0.0663 | 0.0119 | 0.0019 | 0.0339 | 0.0128 | 0.0030 | 0.0345 | 0.0061 | 0.0010 |
| 400 | 0.0071 | 0.0017 | 0.0000 | 0.0101 | 0.0021 | 0.0004 | 0.0183 | 0.0034 | 0.0004 | 0.0152 | 0.0049 | 0.0009 | 0.0092 | 0.0021 | 0.0004 |
| 600 | 0.0036 | 0.0005 | 0.0000 | 0.0051 | 0.0008 | 0.0000 | 0.0100 | 0.0014 | 0.0000 | 0.0088 | 0.0024 | 0.0002 | 0.0050 | 0.0008 | 0.0000 |
| 800 | 0.0021 | 0.0003 | 0.0000 | 0.0035 | 0.0005 | 0.0000 | 0.0056 | 0.0009 | 0.0000 | 0.0064 | 0.0014 | 0.0001 | 0.0034 | 0.0005 | 0.0000 |
| 1000 | 0.0017 | 0.0002 | 0.0000 | 0.0026 | 0.0004 | 0.0000 | 0.0043 | 0.0006 | 0.0000 | 0.0050 | 0.0009 | 0.0000 | 0.0026 | 0.0003 | 0.0000 |
| 1200 | 0.0012 | 0.0001 | 0.0000 | 0.0020 | 0.0002 | 0.0000 | 0.0030 | 0.0004 | 0.0000 | 0.0036 | 0.0008 | 0.0000 | 0.0020 | 0.0002 | 0.0000 |
| 1400 | 0.0007 | 0.0001 | 0.0000 | 0.0012 | 0.0001 | 0.0000 | 0.0018 | 0.0002 | 0.0000 | 0.0025 | 0.0005 | 0.0000 | 0.0012 | 0.0001 | 0.0000 |

| $m = 2n$ | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 |
| 200 | 0.0059 | 0.0010 | 0.0000 | 0.0118 | 0.0018 | 0.0000 | 0.0204 | 0.0027 | 0.0000 | 0.0141 | 0.0039 | 0.0000 | 0.0096 | 0.0017 | 0.0000 |
| 400 | 0.0022 | 0.0001 | 0.0000 | 0.0033 | 0.0003 | 0.0000 | 0.0071 | 0.0005 | 0.0000 | 0.0056 | 0.0010 | 0.0000 | 0.0028 | 0.0003 | 0.0000 |
| 600 | 0.0010 | 0.0000 | 0.0000 | 0.0014 | 0.0001 | 0.0000 | 0.0024 | 0.0002 | 0.0000 | 0.0025 | 0.0004 | 0.0000 | 0.0011 | 0.0001 | 0.0000 |
| 800 | 0.0004 | 0.0000 | 0.0000 | 0.0008 | 0.0000 | 0.0000 | 0.0015 | 0.0000 | 0.0000 | 0.0017 | 0.0001 | 0.0000 | 0.0007 | 0.0000 | 0.0000 |
| 1000 | 0.0002 | 0.0000 | 0.0000 | 0.0005 | 0.0000 | 0.0000 | 0.0009 | 0.0000 | 0.0000 | 0.0013 | 0.0000 | 0.0000 | 0.0004 | 0.0000 | 0.0000 |
| 1200 | 0.0002 | 0.0000 | 0.0000 | 0.0003 | 0.0001 | 0.0000 | 0.0006 | 0.0001 | 0.0000 | 0.0009 | 0.0001 | 0.0000 | 0.0003 | 0.0001 | 0.0000 |
| 1400 | 0.0000 | 0.0000 | 0.0000 | 0.0002 | 0.0000 | 0.0000 | 0.0005 | 0.0000 | 0.0000 | 0.0006 | 0.0000 | 0.0000 | 0.0002 | 0.0000 | 0.0000 |

Appendix B describes in detail the simulations for Bernoulli, Gaussian, and heavy-tailed $t$ data. These results are similar to the Poisson case. Our method performs at least as well as the other procedures in all cases and shows signs of convergence.

Overall, the simulations confirm the conclusions of Proposition 3.3, and they show that our approximate maximization algorithm performs well. These results give us confidence that profile likelihood based biclustering can be used in practice.

## 7. Applications

In this section we use profile-likelihood-based biclustering to reveal structure in two high-dimensional datasets. For each example, we maximize the profile log-likelihood using the algorithm described in Section 5. An additional application example is provided in Appendix B.

In all three examples, we compare the biclusters selected using the profile-likelihood based procedure to those selected by DI-SIM, SBC and KM. For SBC, we compare performance using the $K$ and $L$ selected for PL and the $K$ and $L$ selected using the function provided in the R package. We also compare the biclusters selected by PL to those selected by the Convex Biclustering Algorithm (CBC) [11]. CBC is a biclustering procedure designed for continuous data. The algorithm is based on a convex criterion function and computes a biclustering solution path for varying values of a tuning parameter, $\gamma$. We implement CBC using the R package `cvxbiclustr`. The selected $K$ and $L$ are not specified explicitly, and are instead dependent on $\gamma$. In our experiments, we select $\gamma$ using the function provided in the R package. All other parameters are set to the default package settings.

### *7.1. GovTrack*

In our first application of the proposed method, we cluster legislators and motions based on the roll-call votes from the 113th United States House of Representatives (years 2013–2014). We validate our method by showing that the clusters found by the method agree with the political parties of the legislators.

After downloading the roll-call votes from `govtrack.org`, we form a data matrix $X$ with rows corresponding to the 444 legislators who voted in the House of Representatives, and columns corresponding to the 545 motions voted upon. Even though there are only 435 seats in the House of Representatives, 9 legislators were replaced mid-session when they resigned or died. We code the non-missing votes as

$$X_{ij} = \begin{cases} 1 & \text{if legislator } i \text{ voted “Yea” on motion } j, \\ 0 & \text{if legislator } i \text{ voted “Nay” on motion } j, \\ \text{NA} & \text{if legislator } i \text{ did not vote on motion } j. \end{cases}$$

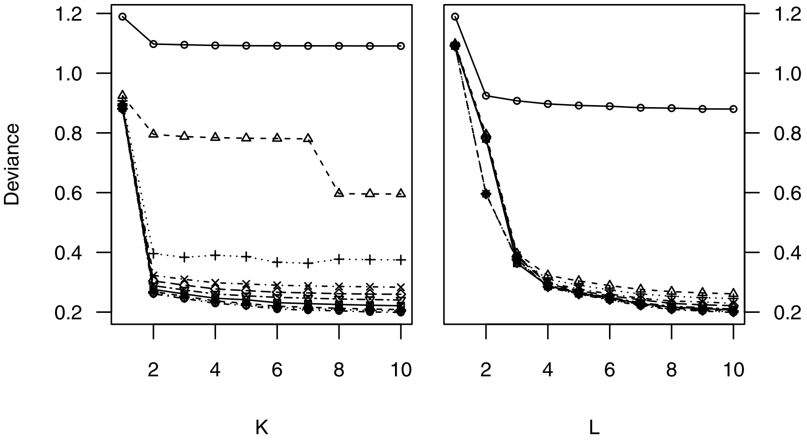

FIG 2. *GovTrack likelihood for different values of $K$ and $L$.*

Not all legislators vote on all motions, and 7% of the data matrix entries are missing, coded as NA. If we assume that the missing data mechanism is ignorable, then it is straightforward to extend our fitting procedure to handle incomplete data matrices. Specifically, to handle the missing data, we replace sums over all matrix entries with sums over all non-missing matrix entries.

To choose the number of row clusters, $K$, and the number of column clusters, $L$, we fit all 100 candidate models with $K$ and $L$ each ranging between 1 and 10. Figure 2 shows the deviance (twice the negative log-likelihood) plotted as a function of $K$ and $L$. The left "scree" plot shows that for most values of $K$, increasing $L$ from 1 to 4 has a large effect of the deviance, but increases $L$ from 4 to a larger value has only a minor effect. Similarly, the right plot shows that increasing $K$ from 1 to 2 causes a large drop in the deviance, but increasing $K$ to a larger value has only a minor effect. Together, these plot suggest that we should pick $K = 2$ and $L = 4$.

To guard against the possibility of the bicluster algorithm finding a local rather than global optimum, we used 100 random initializations, and chose the row and column cluster assignments with the highest log-likelihood. To check the robustness of this assignment, we increased the number of random initializations up to 1000. Even with 10 times as many restarts, we still found the same optimal log-likelihood.

Figure 3 shows a heatmap constructed after biclustering the data into 2 row clusters and 4 column clusters. The two row-clusters found by the algorithm completely recover the political parties of the legislators (every legislator in row cluster 1 is a Democrat, and every legislator in row cluster 2 is a Republican). The fact that we were able to recover the political party provides us some confidence that the algorithm can find meaningful clusters in practice. The column clusters reveal four types of motions: (1) motions with strong support from both parties; (2) motions with moderate support from both parties; (3) motions with strong Democratic support; (4) motions with strong Republican support.

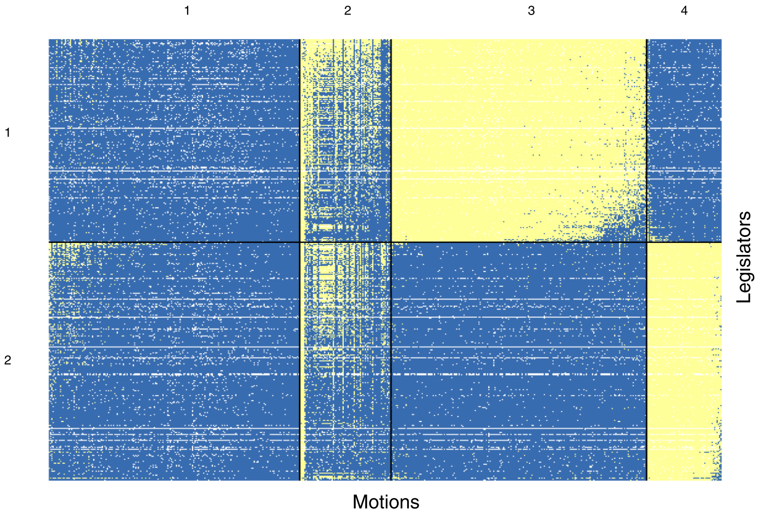

Fig 3. *Heatmap generated from GovTrack data reflecting the voting patterns in the in the different biclusters. Blue (dark) identifies "Yea" votes, yellow (light) identifies "Nay" votes, and white identifies missing votes.*

We compared the clusters found by our method with those found by the four competing methods. The competing methods do not directly handle missing data, so for these methods we code "Yea" as +1, "Nay" as −1, and "Missing" as 0.

Setting $K = 2$ and $L = 4$, DI-SIM returns the same row-clusters, but classified 50 columns (9%) differently. KM placed 10 Republicans into the majority-Democrat cluster; it classified 45 motions (8%) differently from profile likelihood. SBC classified 3 Republicans into the majority-Democrat cluster, and had less agreement with the column-clusters with 137 motions (25.1%) classified differently from PL.

Allowing $K$ and $L$ to vary, SBC and CBC all selected more row and column clusters. SBC selected 9 row clusters, where only one cluster contained a mix of Republicans and Democrats, and 11 column clusters. Although the number of clusters differ, the Rand Index between the row clusters and column clusters found by PL and SBC is 0.746 and 0.653, respectively, suggesting that similar clusterings are found by the two methods. On there other hand, CBC selected 111 row clusters and 520 column clusters, where many of the clusters only contained one row/column. CBC is designed for continuous data and the default settings do not appear to produce meaningful clusters in this example.

Overall, the agreement between PL and DI-SIM, KM and SBC gives us confidence that the column clusters are meaningful.

## *7.2. AGEMAP*

Biclustering is commonly used for microarray data to visualize the activation patterns of thousands of genes simultaneously. It is used to identify patterns and discover distinguishing properties between genes and individuals. We use the AGEMAP dataset [51] to illustrate this process.

AGEMAP is a large microarray data set containing the log expression levels for 40 mice across 8,932 genes measured on 16 different tissue types. For this analysis, we restrict attention to two tissue types: cerebellum and cerebrum. The 40 mice in the dataset belong to four age groups, with five males and five females in each group. One of the mice is missing data for the cerebrum tissue so it has been removed from the dataset.

Our goal is to uncover structure in the gene expression matrix. We bicluster the 39 × 17,864 residual matrix computed from the least squares solution to the multiple linear regression model

$$Y_{ij} = \beta_{0j} + \beta_{1j} A_i + \beta_{2j} S_i + \varepsilon_{ij},$$

where $Y_{ij}$ is the log-activation of gene $j$ in mouse $i$, $A_i$ is the age of mouse $i$, $S_i$ indicates if mouse $i$ is male, $\varepsilon_{ij}$ is a random error, and $(\beta_{0j}, \beta_{1j}, \beta_{2j})$ is a gene-specific coefficient vector. Here, we are biclustering the residual matrix rather than the raw gene expression matrix because we are interested in the structure remaining after adjusting for the observed covariates.

The entries of the residual matrix are not independent (for example, the sum of each column is zero). Also, the responses of many genes are likely correlated with each other. Thus, the block model assumption required by Theorem 4.1 is not satisfied, so its conclusion will not hold unless the dependence between the residuals is negligible. In light of this caveat, the example should be considered as exploratory data analysis.

We perform biclustering using profile likelihood based on the Gaussian criterion (6.1c) with 100 random initializations. To determine an appropriate number of mice clusters, $K$, and gene clusters, $L$, we experiment with values of $K$ and $L$ between 1 and 15. Figure 4 presents the scree plots. The left plot shows that increasing $L$ beyond 5 has a relatively small impact on the deviance, and similarly, the right plot shows that increasing $K$ beyond 3 has a relatively minor effect. This suggests we should set $K = 3$ and $L = 5$. For this choice of $K$ and $L$, we experimented with using up to 1000 random starting values, but found no change to the resulting log-likelihood.

The heatmap presented in Figure 5 shows the results. The expression levels for gene group 1 and 3 appear to be fairly neutral across the three mouse groups, but the other three gene groups have a more visually apparent pattern. It appears that a mouse can have expression levels for at most two of gene groups 2, 4, and 5. Mouse group 2 has high expression for gene groups 2 and 4; mouse group 2 has high expression for gene group 4; and mouse group 3 has high expression for gene group 5.

With the exception of CBC, the competing methods found similar gene groups to those found by PL. Setting $L = 5$, KM and SBC generally agree with PL, with 89.5% and 99.7% cluster agreement, respectively. DI-SIM agreed less, but still had 62.5% cluster agreement with PL. Allowing the number of clusters to to vary, SBC still selects a relatively small number of gene groups with $L = 8$, with a Rand Index with PL of 0.695.

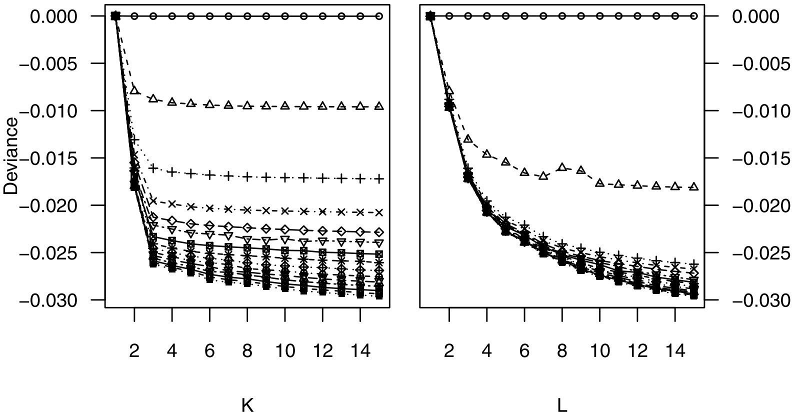

FIG 4. *AgeMap likelihood for different values of K and L.*

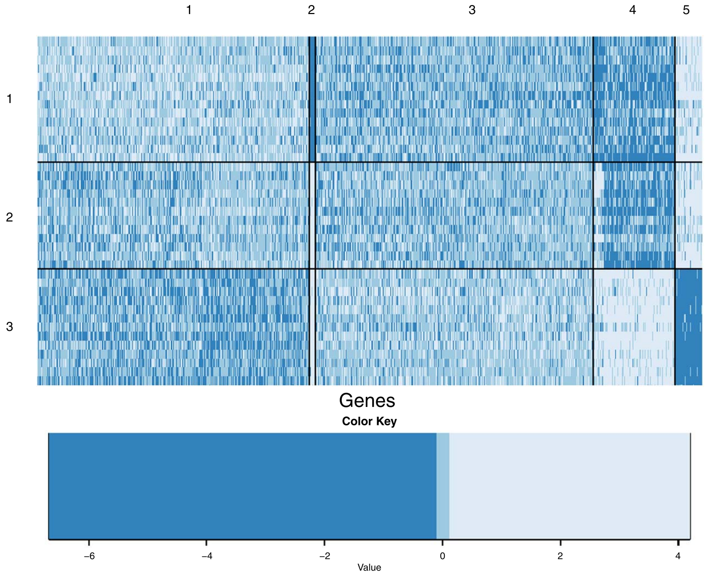

FIG 5. *Heatmap generated from AGEMAP residual data reflecting the varying expression patterns in the different biclusters. The colors for the matrix entries correspond encode to the first quartile, the middle two quartiles, and the upper quartile.*

CBC selected a much larger number of gene groups than any of the competing methods, with $L = 394$. The gene groups include 1 very large cluster with 16,870 genes and 351 clusters each with only 1 gene. Based on discussions with the authors, the default selections for CBC may not be appropriate when the dimensions of the matrix vary by such a large amount. Adjusting the weights used in the method could produce more meaningful results, but this goes beyond the scope of our analysis.

CBC, DI-SIM, KM and SBC all select the same mice groups as PL, even when $K$ is allowed to vary for CBC and SBC. The agreement between all of the methods suggests that the three mice groups are meaningful. The three clusters of mice found by the methods also agree with those found by Perry and Owen [39]. That analysis identified the mouse clusters, but could not attribute meaning to them. The bicluster based analysis has deepened our understanding of the three mouse clusters while suggesting some interesting interactions between the genes.

## 8. Discussion

We have developed a statistical setting for studying the performance of biclustering algorithms. Under the assumption that the data follows a stochastic block model, we derived sufficient conditions for an algorithm based on maximizing a profile-likelihood based criterion function to be consistent. This is the first theoretical guarantee for any biclustering algorithm which can be applied to a broad range of data distributions and can handle both sparse and dense data matrices.

Since maximizing the criterion function exactly is computationally infeasible, we have proposed an approximate algorithm for obtaining a local optimum rather than a global one. We have shown through simulations that the approximation algorithm has good performance in practice. Our empirical comparisons demonstrated that the method performs well in a variety of situations and can outperform existing procedures.

Applying the profile-likelihood based biclustering algorithm to real data revealed several interesting findings. Our results from the GovTrack dataset demonstrated our methods ability to recover ground truth labels when available, and identified motion clusters that were robust across different methods. Biclustering the genes and mice in the AGEMAP data exposed an interesting pattern in the expression of certain genes and we found that at most two gene groups can have high expression levels for any one mouse. The consistency theorem proved in this report gives conditions under which we can have confidence in the robustness of these findings.

## Appendix A: Additional proofs and theoretical results

### *A.1. Proof of formal consistency theorem*

Our proof of the main theoretical results follows the same outline as Section 3. In particular, Proposition A.2 is a rigorous statement analogous to Proposition 3.1; Proposition A.3 is analogous to Proposition 3.2; Theorems 4.1 and 4.2 are analogous to Proposition 3.3.

### *A.1.1. Population criterion*

Here, we give a rigorous statement of Proposition 3.1. That is, we establish that in the limit, $F$ is close to its nonrandom population version, $G$, which depends only on the confusion matrices.

We will need some additional notation and a concentration result. Define normalized residual matrix $\boldsymbol{R}(\boldsymbol{g}, \boldsymbol{h}) \in \mathbb{R}^{K \times L}$ by

$$\boldsymbol{R}(\boldsymbol{g}, \boldsymbol{h}) = \rho^{-1}\{\bar{\boldsymbol{X}}(\boldsymbol{g}, \boldsymbol{h}) - \boldsymbol{E}(\boldsymbol{g}, \boldsymbol{h})\}.$$

The law of large numbers establishes that for fixed $\boldsymbol{g}$ and $\boldsymbol{h}$, the convergence $R_{kl}(\boldsymbol{g}, \boldsymbol{h}) \overset{P}{\to} 0$ holds. We can prove a stronger concentration result, that this convergence is uniform over all $\boldsymbol{g}$ and $\boldsymbol{h}$.

**Lemma A.1.** *Under conditions (C1)–(C6), for all $\varepsilon > 0$,*

$$\sup_{\mathcal{J}_\varepsilon} \|\boldsymbol{R}(\boldsymbol{g}, \boldsymbol{h})\|_\infty \overset{P}{\to} 0,$$

*where $\|\mathbf{A}\|_\infty = \max_{k,l} |A_{kl}|$ for any matrix $\mathbf{A}$.*

*Proof.* For all $t > 0$,

$$\Pr\Big(\sup_{\mathcal{J}_\varepsilon} \|\boldsymbol{R}(\boldsymbol{g}, \boldsymbol{h})\|_\infty > t\Big) \leq KL \Pr\Big(\sup_{I \in \mathcal{I}_n} \rho^{-1}\Big|\sum_{\{i,j\} \in I} (X_{ij} - \mu_{c_i d_j})\Big| > t|I|\Big),$$

where $\mathcal{I}_n \subset 2^{[n]} \times 2^{[m]}$ is the set of all biclusters such that $\hat{p}_k > \varepsilon$ for all $k$ and $\hat{q}_l > \varepsilon$ for all $l$. Since $\mathcal{I}_n$ is a subset of the power set $2^{[nm]}$, by Lemma A.4 in Appendix A.2, it follows that the right hand side tends to zero. □

With Lemma A.1, we can establish that in the limit, $F(\boldsymbol{g}, \boldsymbol{h})$ is close to $G(\boldsymbol{C}, \boldsymbol{D})$.

**Proposition A.2.** *$F$ is close to its population version in the sense that, for all $\varepsilon > 0$,*

$$\sup_{\mathcal{J}_\varepsilon} |F(\boldsymbol{g}, \boldsymbol{h}) - G(\boldsymbol{C}, \boldsymbol{D})| \overset{P}{\to} 0.$$

*Proof.* The technical assumptions of $f$ imply that its first derivative is bounded. Therefore, $f$ is locally Lipschitz continuous with Lipschitz constant $c = \sup |f'(\mu)|$ for $\mu$ in a neighborhood of $\mathcal{M}$ and

$$|F(\boldsymbol{g}, \boldsymbol{h}) - G(\boldsymbol{C}, \boldsymbol{D})| \leq c\|\boldsymbol{R}(\boldsymbol{g}, \boldsymbol{h})\|_\infty.$$

From Proposition A.1, the right hand side converges to zero almost surely and the result follows. □

### *A.1.2. Self-consistency*

Once we have established that $F$ is close to its population version, our next task is to show that the population version is self-consistent. We will need a more precise version of Proposition 3.2.

To state the result, for $\delta > 0$, define

$$\mathcal{P}_\delta = \{\boldsymbol{C} \in \mathcal{C}_{\boldsymbol{p}} : \max_{a \neq a'} C_{ak} C_{a'k} < \delta\}, \tag{A.1a}$$

$$\mathcal{Q}_\delta = \{\boldsymbol{D} \in \mathcal{C}_{\boldsymbol{q}} : \max_{b \neq b'} D_{bl} D_{b'l} < \delta\}. \tag{A.1b}$$

A permutation of a diagonal matrix has only one non-zero entry in each column, so taking $\delta$ close to zero restricts the confusion matrices to be close to permutations of diagonal matrices.

**Proposition A.3.** *If* $\min_a\{p_a\} > \eta$, $\min_b\{q_b\} > \eta$, *and* $(\boldsymbol{C}, \boldsymbol{D}) \notin \mathcal{P}_\delta \times \mathcal{Q}_\delta$, *then* $G(\boldsymbol{C}, \boldsymbol{D})$ *is small, in the sense that*

$$G(\boldsymbol{C}, \boldsymbol{D}) - \sum_{a,b} p_a \, q_b \, f(\rho^{-1} \mu_{ab}) \leq -\kappa \eta^2 \delta,$$

*where* $\kappa$ *is a constant independent of* $\delta$ *and* $\eta$.

*Proof.* If $\boldsymbol{D} \notin \mathcal{Q}_\delta$, then for some $l$ and some $b \neq b'$, $D_{bl} D_{b'l} \geq \delta$. Since no two columns of $\boldsymbol{M}$ are identical, there exists an $a$ such that $\mu_{ab} \neq \mu_{ab'}$. Let $k$ be the index of the largest element in row $a$ of matrix $\boldsymbol{C}$; this element must be at least as large as the mean, i.e.

$$C_{ak} \geq \frac{[\boldsymbol{C}\mathbf{1}]_a}{K} \geq \frac{\eta}{K}.$$

Let $W = [\boldsymbol{C}^T \mathbf{1}]_k [\boldsymbol{D}^T \mathbf{1}]_l$; this is nonzero. Now, there exists $\mu_* \in \mathcal{M}$ such that

$$[\boldsymbol{C}^T \boldsymbol{M} \boldsymbol{D}]_{kl} = C_{ak} D_{bl} \mu_{ab} + C_{ak} D_{b'l} \mu_{ab'} + (W - C_{ak} D_{bl} - C_{ak} D_{b'l}) \mu_*.$$

Let $z = [\boldsymbol{C}^T \boldsymbol{M} \boldsymbol{D}]_{kl} / W$. Set $\kappa_0 = \inf_{\mu \in \mathcal{M}} f''(\mu)$ and define $\boldsymbol{N} = [\nu_{ab}] \in \mathbb{R}^{A \times B}$ with $\nu_{ab} = f(\rho^{-1} \mu_{ab})$. By a refined Jensen's inequality (Lemma A.5 in Appendix A.2), it follows that

$$\begin{aligned}
\frac{[\boldsymbol{C}^T \boldsymbol{N} \boldsymbol{D}]_{kl}}{W} - f(z) &\geq \kappa_0 \frac{C_{ak}^2 D_{bl} D_{b'l}}{W^2} \Big( \frac{1}{2} (\mu_{ab} - z)^2 + \frac{1}{2} (z - \mu_{ab'})^2 \Big) \\
&\geq \kappa_0 \frac{C_{ak}^2 D_{bl} D_{b'l}}{W^2} \Big( \frac{1}{2} (\mu_{ab} - z) + \frac{1}{2} (z - \mu_{ab'}) \Big)^2 \\
&= \kappa_0 \frac{C_{ak}^2 D_{bl} D_{b'l}}{4W^2} (\mu_{ab} - \mu_{ab'})^2.
\end{aligned}$$

Thus

$$[\boldsymbol{C}^T\mathbf{1}]_k[\boldsymbol{D}^T\mathbf{1}]_l f\left(\frac{[\boldsymbol{C}^T\boldsymbol{M}_0\boldsymbol{D}]_{kl}}{[\boldsymbol{C}^T\mathbf{1}]_k[\boldsymbol{D}^T\mathbf{1}]_l}\right) - [\boldsymbol{C}^T\boldsymbol{N}\boldsymbol{D}]_{kl}$$
$$\leq -\frac{\kappa_0}{2}(\mu_{ab}-\mu_{ab'})^2\frac{C_{ak}^2 D_{bl}D_{b'l}}{W} \leq -\frac{\kappa_0\eta^2\delta}{4K^2}(\mu_{ab}-\mu_{ab'})^2.$$

This inequality only holds for one particular choice of $k$ and $l$; for other choices, the left hand side is nonpositive by Jensen's inequality. Defining

$$\kappa_1 = \frac{\kappa_0}{4}\min_{a,b\neq b'}(\mu_{ab}-\mu_{ab'})^2,$$

it follows that

$$G(\boldsymbol{C},\boldsymbol{D}) - \sum_{a,b} p_a\, q_b\, f(\rho^{-1}\mu_{ab}) \leq -\frac{\kappa_1\,\eta^2\delta}{K^2}.$$

Similarly, if $\boldsymbol{C}\notin\mathcal{P}_\delta$, then the right hand side is bounded by

$$-\kappa_2\,\eta^2\delta/L^2$$

where

$$\kappa_2 = \frac{\kappa_0}{4}\min_{a\neq a',b}(\mu_{ab}-\mu_{a'b})^2.$$

The result of the proposition follows with $\kappa = \min(\kappa_1,\kappa_2)/\max\{K^2,L^2\}$. □

### *A.1.3. Consistency*

We are now ready to state and prove the formal consistency theorem.

*Proof of Theorem 4.1.* Fix $\delta > 0$ and define $\mathcal{P}_\delta$ and $\mathcal{Q}_\delta$ as in (A.1). We will show that if $(\boldsymbol{g},\boldsymbol{h})\in\mathcal{J}_\varepsilon$ and if $(\boldsymbol{C}(\boldsymbol{g}),\boldsymbol{D}(\boldsymbol{h}))\notin(\mathcal{P}_\delta,\mathcal{Q}_\delta)$, then $F(\boldsymbol{g},\boldsymbol{h}) < F(\boldsymbol{c},\boldsymbol{d})$ with probability approaching one. Moreover, this inequality holds uniformly over all such choices of $(\boldsymbol{g},\boldsymbol{h})$. Since $\delta$ is arbitrary, this implies that $\boldsymbol{C}(\hat{\boldsymbol{g}})$ and $\boldsymbol{D}(\hat{\boldsymbol{h}})$ converge to permutations of diagonal matrices, i.e. the proportions of misclassified rows and columns converge to zero.

Set $r_n = \sup_{\mathcal{J}_\varepsilon}|F(\boldsymbol{g},\boldsymbol{h}) - G(\boldsymbol{C}(\boldsymbol{g}),\boldsymbol{D}(\boldsymbol{h})|$. Suppose $(\boldsymbol{g},\boldsymbol{h})\in\mathcal{J}_\varepsilon$. In this case,

$$F(\boldsymbol{g},\boldsymbol{h}) - F(\boldsymbol{c},\boldsymbol{d}) \leq 2r_n + \{G(\boldsymbol{C}(\boldsymbol{g}),\boldsymbol{D}(\boldsymbol{h})) - G(\boldsymbol{C}(\boldsymbol{c}),\boldsymbol{D}(\boldsymbol{d}))\}$$
$$= 2r_n + \Big\{G(\boldsymbol{C}(\boldsymbol{g}),\boldsymbol{D}(\boldsymbol{h})) - \sum_{a,b}[\boldsymbol{C}\mathbf{1}]_a[\boldsymbol{D}\mathbf{1}]_b f([\boldsymbol{M}_0]_{ab})\Big\}.$$

Pick $\eta > 0$ smaller than $\min_a\{p_a\}$ and $\min_b\{q_b\}$. By assumption, the true row and column class proportions converge to $\boldsymbol{p}$ and $\boldsymbol{q}$. Thus, for all $\boldsymbol{g}\in K^m$ and $\boldsymbol{h}\in L^n$, for $n$ large enough, $[\boldsymbol{C}(\boldsymbol{g})]_a \geq \eta$ and $[\boldsymbol{D}(\boldsymbol{h})]_b \geq \eta$; this holds uniformly over all choices of $(\boldsymbol{g},\boldsymbol{h})$.

Applying Proposition A.3, to the second term in the inequality, we get that with probability approaching one,

$$F(\boldsymbol{g},\boldsymbol{h}) - F(\boldsymbol{c},\boldsymbol{d}) \leq 2r_n - \kappa\eta^2\delta$$

for all $(\boldsymbol{g}, \boldsymbol{h}) \in \mathcal{J}_\varepsilon$ such that $(\boldsymbol{C}(\boldsymbol{g}), \boldsymbol{D}(\boldsymbol{h})) \notin \mathcal{P}_\delta \times \mathcal{Q}_\delta$. By Proposition A.2, $r_n \xrightarrow{P} 0$. Thus, with probability approaching one, $(\boldsymbol{C}(\hat{\boldsymbol{g}}), \boldsymbol{D}(\hat{\boldsymbol{h}})) \in \mathcal{P}_\delta \times \mathcal{Q}_\delta$. Since this result holds for all $\delta$, all limit points of $\boldsymbol{C}(\hat{\boldsymbol{g}})$ and $\boldsymbol{D}(\hat{\boldsymbol{h}})$ must be permutations of diagonal matrices. □

### *A.1.4. Empirical treatment of ρ*

For the Poisson and Gaussian relative entropy functions (6.1b) and (6.1c), the maximizer of the criterion function (2.2) does not depend on the scale factor $\rho$. This is immediately obvious in the Gaussian case. For the Poisson case, the function $f_{\text{Poisson}}(\mu/\rho) = \frac{1}{\rho}\mu\log(\mu) - \frac{1}{\rho}\mu(1+\log(\rho))$. When summed over all biclusters, the second term in this sum is equal to a constant so the value of $\mu$ which maximizes $f_{\text{Poisson}}(\mu/\rho)$ does not depend on the value of $\rho$.

This is not the case for the Binomial relative entropy function (6.1a), but the parameter $\rho$ is not identifiable in practice so it does not make sense to try to estimate it. For our simulations we use which maximizes $\rho = 1$ in the fitting procedure, regardless of the true scale factor for the mean matrix $\boldsymbol{M}$. Even though in the simulations the identifiability condition doesn't hold for this choice of $\rho$, we still get consistency, because the maximizer of the criterion with $f_{\text{Bernoulli}}(\mu)$ is close to the maximizer with $f_{\text{Poisson}}(\mu)$. See Perry and Wolfe [40] for discussion of a related phenomenon.

## ***A.2. Additional technical results***

**Lemma A.4.** *For each $n$, let $X_{n,m}$, $1 \leq m \leq n$, be independent random variables with $\operatorname{E} X_{n,m} = 0$. Let $\rho_n$ be a sequence of positive numbers. Let $\mathcal{I}_n$ be a subset of the powerset $2^{[n]}$, with $\inf\{|I| : I \in \mathcal{I}_n\} \geq L_n$. Suppose*

*(i)* $\frac{1}{n\rho_n}\sum_{m=1}^n \operatorname{E}|X_{n,m}|^2$ *is uniformly bounded in $n$;*
*(ii)* *For all $\varepsilon > 0$,* $\frac{1}{n\rho_n^2}\sum_{m=1}^n \operatorname{E}(|X_{n,m}|^2; |X_{n,m}| > \varepsilon\sqrt{n}\rho_n) \to 0$;
*(iii)* $\overline{\lim}_{n\to\infty} \frac{n}{L_n} < \infty$;
*(iv)* $\overline{\lim}_{n\to\infty} \frac{\log|\mathcal{I}_n|}{\sqrt{n}} < \infty$.
*(v)* $\overline{\lim}_{n\to\infty} \rho_n\sqrt{n} = \infty$.

*Then*

$$\sup_{I \in \mathcal{I}_n} \left| \frac{1}{\rho_n |I|} \sum_{m \in I} X_{n,m} \right| \xrightarrow{P} 0.$$

*Proof.* Let $\varepsilon > 0$ be arbitrary. Define $Y_{n,m} = \rho_n^{-1} X_{n,m} \operatorname{I}(|X_{n,m}| \leq \varepsilon\sqrt{n}\rho_n)$, and note that

$$\begin{aligned}\Pr(Y_{n,m} \neq \rho_n^{-1} X_{n,m} \text{ for some } 1 \leq m \leq n) &\leq \sum_{m=1}^n \Pr(|X_{n,m}| > \varepsilon\sqrt{n}\rho_n) \\ &\leq \frac{1}{\varepsilon^2 n \rho_n^2} \sum_{m=1}^n \operatorname{E}(|X_{n,m}|^2; |X_{n,m}| > \varepsilon\sqrt{n}\rho_n).\end{aligned}$$

Fix an arbitrary $t > 0$. Set $\mu_{n,m} = \mathrm{E}\, Y_{n,m}$ and for $I \in \mathcal{I}_n$ define

$$\mu_n(I) = \frac{1}{|I|} \sum_{m\in I} \mu_{n,m}.$$

For $I \in \mathcal{I}_n$, write

$$\Pr\Big(\sum_{m\in I} Y_{n,m} > t\,|I|\Big) = \Pr\Big(\sum_{m\in I} (Y_{n,m} - \mu_{n,m}) > |I|\big(t - \mu_n(I)\big)\Big).$$

Note that since $\mathrm{E}\, X_{n,m} = 0$, it follows that

$$\begin{aligned}|\mu_{n,m}| &= |-\mathrm{E}(\rho_n^{-1} X_{n,m}; |X_{n,m}| > \varepsilon\sqrt{n}\rho_n)| \\ &\le \frac{1}{\varepsilon\sqrt{n}\rho_n^2}\, \mathrm{E}(|X_{n,m}|^2; |X_{n,m}| > \varepsilon\sqrt{n}\rho_n).\end{aligned}$$

Thus, by (ii) and (iii) we have that $\sup_{I\in\mathcal{I}_n}\{|\mu_n(I)|\} \to 0$; in particular, for $n$ large enough, $\sup_{I\in\mathcal{I}_n}\{|\mu_n(I)|\} < \frac{t}{2}$. Consequently, for $n$ large enough, uniformly for all $I$,

$$\Pr\Big(\sum_{m\in I} Y_{n,m} > t\,|I|\Big) \le \Pr\Big(\sum_{m\in I} (Y_{n,m} - \mu_{n,m}) > t\,|I|/2\Big).$$

Similarly,

$$\Pr\Big(\sum_{m\in I} Y_{n,m} < -t\,|I|\Big) \le \Pr\Big(\sum_{m\in I} (Y_{n,m} - \mu_{n,m}) < -t\,|I|/2\Big).$$

We apply Bernstein's inequality to the bound. Define $\sigma_{n,m}^2 = \mathrm{E}(Y_{n,m} - \mu_{n,m})^2$ and $v_n(I) = \sum_{m\in I} \sigma_{n,m}^2$. Note that $|Y_{n,m} - \mu_{n,m}| \le 2\varepsilon\sqrt{n}$. By Bernstein's inequality,

$$\Pr\Big(\Big|\sum_{m\in I} (Y_{n,m} - \mu_{n,m})\Big| > t\,|I|/2\Big) \le 2\exp\Big\{-\frac{t^2|I|^2/8}{v_n(I) + \varepsilon t|I|\sqrt{n}/3}\Big\}.$$

By (i), (iv), and (v), it follows that for $n$ large enough, $v_n(I) < \varepsilon t|I|\sqrt{n}/3$, so

$$\Pr\Big(\Big|\sum_{m\in I_n} (Y_{n,m} - \mu_{n,m})\Big| > t|I|/2\Big) \le 2\exp\Big\{-\frac{t|I|}{8\varepsilon\sqrt{n}}\Big\}.$$

We use this bound for each $I$ to get the union bound:

$$\begin{aligned}\Pr\Big(\sup_{I\in\mathcal{I}_n}\Big|\frac{1}{|I|}\sum_{m\in I} Y_{n,m}\Big| > t\Big) &\le 2|\mathcal{I}_n| \exp\Big\{-\frac{tL_n}{8\varepsilon\sqrt{n}}\Big\} \\ &= 2\exp\Big\{\log|\mathcal{I}_n| - \frac{tL_n}{8\varepsilon\sqrt{n}}\Big\}.\end{aligned}$$

By (iii) and (iv), it is possible to choose $\varepsilon$ such that the right hand side goes to zero. It follows then that

$$\begin{aligned}\Pr\Big(\sup_{I\in\mathcal{I}_n}\Big|\frac{1}{\rho_n|I|}\sum_{m\in I}X_{n,m}\Big|>t\Big) &\leq \Pr(Y_{n,m}\neq\rho_n^{-1}X_{n,m}\text{ for some }1\leq m\leq n)\\ &\quad+\Pr\Big(\sup_{I\in\mathcal{I}_n}\Big|\frac{1}{|I|}\sum_{m\in I}Y_{n,m}\Big|>t\Big)\\ &\quad\to 0. \qquad\square\end{aligned}$$

**Lemma A.5** (Refined Jensen's Inequality)**.** *Let $f:\mathbb{R}\to\mathbb{R}$ be twice differentiable and let $\mathcal{N}$ be a convex set in $\mathbb{R}$. If $x_1,\ldots,x_n$ are points in $\mathcal{N}$, and if $w_1,\ldots,w_n$ are nonnegative numbers summing to one, then*

$$\sum_{i=1}^{n}w_if(x_i)-f(z)\geq\frac{1}{2}\inf_{y\in\mathcal{N}}f''(y)\sum_{i=1}^{n}w_i(x_i-z)^2,$$

*where $z=\sum_{i=1}^{n}w_ix_i$.*

*Proof.* Define $\kappa_0=\inf_{y\in\mathcal{N}}f''(y)$ and use the bound

$$f(x_i)\geq f(z)+f'(z)(x_i-z)+\frac{\kappa_0}{2}(x_i-z)^2. \qquad\square$$

### *A.3. Finite sample results*

In this appendix we derive a finite sample tail bound for the probability that the class assignments that maximize the profile likelihood are close to the true class labels. To proceed in this setting, we make stronger distributional assumptions than in the asymptotic case. Specifically, we assume here that the entries $X_{ij}|\boldsymbol{c},\boldsymbol{d}$ follow a Gaussian distribution with mean $\mu_{c_id_j}$ and finite variance $\sigma^2$. We proceed with the notation from the main text.

**Proposition A.6.** *For all $\varepsilon>0$, if $t<\sigma$ then*

$$\Pr\Big(\sup_{\mathcal{J}_\varepsilon}\|\boldsymbol{R}(\boldsymbol{g},\boldsymbol{h})\|_\infty>t\Big)\leq 2K^{m+1}L^{n+1}\exp\Big(-\frac{L_nt^2}{4\sigma^2}\Big),$$

*and if $t\geq\sigma$ then*

$$\Pr\Big(\sup_{\mathcal{J}_\varepsilon}\|\boldsymbol{R}(\boldsymbol{g},\boldsymbol{h})\|_\infty>t\Big)\leq 2K^{m+1}L^{n+1}\exp\Big(-\frac{L_nt}{4\sigma}\Big)$$

*where $\|\boldsymbol{A}\|_\infty=\max_{k,l}|A_{kl}|$ for any matrix $\boldsymbol{A}$.*

*Proof.* If the entries $X_{ij}$ follow a Gaussian distribution with mean $\mu_{c_id_j}$ and variance $\sigma^2$ then

$$\mathrm{E}\left(|X_{ij}-\mu_{c_id_j}|^l\right)\leq\frac{\sigma^2}{2}\sigma^{l-2}l!$$

so the conditions of Bernstein's inequality hold. It follows that for any bicluster $I$, for all $t > 0$,

$$\begin{aligned}\Pr\Big(|\sum_{i,j\in I} X_{ij} - \mu_{c_i d_j}| > t|I|\Big) &\leq 2\exp\Big\{-\frac{|I|^2t^2}{2(\sigma^2|I| + \sigma|I|t)}\Big\}\\ &\leq 2\exp\Big\{-\frac{L_n t^2}{4\max\{\sigma^2, \sigma t\}}\Big\}.\end{aligned}$$

Applying a union bound,

$$\Pr\Big(\sup_{\mathcal{J}_\varepsilon} \|\boldsymbol{R}(\boldsymbol{g},\boldsymbol{h})\|_\infty > t\Big) \leq 2K^{m+1}L^{n+1}\exp\Big\{-\frac{L_n t^2}{4\max\{\sigma^2,\sigma t\}}\Big\}. \qquad \square$$

Proposition A.6 is used to establish a finite sample bound on the difference between $F(\boldsymbol{g},\boldsymbol{h})$ and its population version.

**Proposition A.7.** *Under conditions (C2) and (C3), for any $t > 0$,*

$$\Pr\Big(\sup_{\mathcal{J}_\varepsilon} |F(\boldsymbol{g},\boldsymbol{h}) - G(\boldsymbol{C},\boldsymbol{D})| > t\Big) \leq \Pr\Big(\sup_{\mathcal{J}_\varepsilon} \|\boldsymbol{R}(\boldsymbol{g},\boldsymbol{h})\|_\infty > \frac{t}{c}\Big)$$

*where $c = \sup|f'(\mu)|$ for $\mu$ in $\mathcal{M}$.*

Proposition A.7 is a direct consequence of the fact that $f$ is locally Lipschitz continuous under conditions (C2) and (C3). The details are similar to proof of Proposition A.2.

The next step is to show that population version is maximized at the true class labels.

**Proposition A.8.** *Choose $\tau > 0$ such that $\min_{a\neq a',b}(\mu_{ab} - \mu_{a'b})^2 \geq \tau$ and $\min_{a,b\neq b'}(\mu_{ab} - \mu_{ab'})^2 \geq \tau$. Then for all $\varepsilon > 0$, for $(\boldsymbol{g},\boldsymbol{h}) \in \mathcal{J}_\varepsilon$ and $(\boldsymbol{C},\boldsymbol{D}) \notin \mathcal{P}_\delta \cap \mathcal{Q}_\delta$, $G(\boldsymbol{C},\boldsymbol{D})$ is small in the sense that*

$$G(\boldsymbol{C},\boldsymbol{D}) - \sum_{a,b} p_a q_b f(\rho^{-1}\mu_{ab}) \leq -\frac{\tau\varepsilon^2\delta}{4\max\{K^2,L^2\}}.$$

The proof of Proposition A.8 is similar to the proof of Proposition A.3 except that the bound on the difference uses $\varepsilon$ in place of the value $\eta$. Some details follow.

*Proof of Proposition A.8.* First note that in Proposition A.3, we can let $a$ be the index of the largest element in column $k$ of matrix $\boldsymbol{C}$; then, since we are restricted to the set $\mathcal{J}_\varepsilon$, this element must be at least as large as

$$C_{ak} \geq \frac{[\boldsymbol{C}^T\boldsymbol{1}]_k}{K} \geq \frac{\varepsilon}{K}.$$

Noting that for the Gaussian relative entropy function $f''(\mu) = 1$ for all $\mu \in \mathcal{M}$, the remainder of the proof is similar to the proof of Proposition A.3. $\square$

We establish a finite sample bound by combining these results.

*Proof of Theorem 4.2.* Fix $\delta > 0$ and define $\mathcal{P}_\delta$ and $\mathcal{Q}_\delta$ as in Proposition A.3. Set $r_n = \sup_{\mathcal{J}_\varepsilon} |F(\boldsymbol{g}, \boldsymbol{h}) - G(\boldsymbol{C}(\boldsymbol{g}), \boldsymbol{D}(\boldsymbol{h})|$. Suppose $(\boldsymbol{g}, \boldsymbol{h}) \in \mathcal{J}_\varepsilon$. In this case,

$$\begin{aligned} F(\boldsymbol{g}, \boldsymbol{h}) - F(\boldsymbol{c}, \boldsymbol{d}) &\leq 2r_n + \{G(\boldsymbol{C}(\boldsymbol{g}), \boldsymbol{D}(\boldsymbol{h})) - G(\boldsymbol{C}(\boldsymbol{c}), \boldsymbol{D}(\boldsymbol{d}))\} \\ &= 2r_n + \Big\{G(\boldsymbol{C}(\boldsymbol{g}), \boldsymbol{D}(\boldsymbol{h})) - \sum_{a,b} [\boldsymbol{C}\boldsymbol{1}]_a [\boldsymbol{D}\boldsymbol{1}]_b f([\boldsymbol{M}_0]_{ab})\Big\}. \end{aligned}$$

Applying Proposition A.8 to the second term in the inequality, we get that

$$F(\boldsymbol{g}, \boldsymbol{h}) - F(\boldsymbol{c}, \boldsymbol{d}) \leq 2r_n - \frac{\tau \varepsilon^2 \delta}{4 \max\{K^2, L^2\}}$$

for all $(\boldsymbol{g}, \boldsymbol{h}) \in \mathcal{J}_\varepsilon$ such that $(\boldsymbol{C}, \boldsymbol{D}) \notin \mathcal{P}_\delta \cap \mathcal{Q}_\delta$. The result follows by applying Proposition A.7. □

## Appendix B: Additional empirical and application results

This appendix reports additional empirical results for Bernoulli, Gaussian, and Student's $t$ distributed data as well as an additional application example and stability results for the proposed biclustering algorithm.

### *B.1. Additional empirical results*

Figures 6–8 present the average bicluster misclassification rates for each sample size and Tables 2–4 report the standard deviations for the Bernoulli, Gaussian, and $t$ simulations, respectively. Since the normalization for the DI-SIM algorithm is only specified for non-negative data, the algorithm is run on the un-normalized matrix for the Gaussian and non-standardized Student's $t$ examples.

For the Bernoulli simulation, we simulate from a block model with $K = 2$ row clusters and $L = 3$ column clusters. We vary the number of columns, $n$, between 200 to 1400 and we take the number of rows as $m = \gamma n$ where $\gamma \in \{0.5, 1, 2\}$.

We set the row and column class membership probabilities as $\boldsymbol{p} = (0.3, 0.7)$ and $\boldsymbol{q} = (0.2, 0.3, 0.5)$. We choose the matrix of block parameters to be

$$\boldsymbol{M} = \frac{b}{\sqrt{n}} \begin{pmatrix} 0.43 & 0.06 & 0.13 \\ 0.10 & 0.34 & 0.17 \end{pmatrix},$$

where the entries were selected to be on the same scale as Bickel and Chen (2009). We vary $b$ between 5 and 20. We generate the data conditional on the row and column classes as $X_{ij} \mid \boldsymbol{c}, \boldsymbol{d} \sim \text{Bernoulli}(\mu_{c_i d_j})$. We initialize all the methods with 250 random starting values.

For the Gaussian simulation, we simulate from a block model with $K = 2$ row clusters and $L = 3$ column clusters. We vary the number of columns, $n$, between 50 to 400 and we take the number of rows as $m = \gamma n$ where $\gamma \in \{0.5, 1, 2\}$.

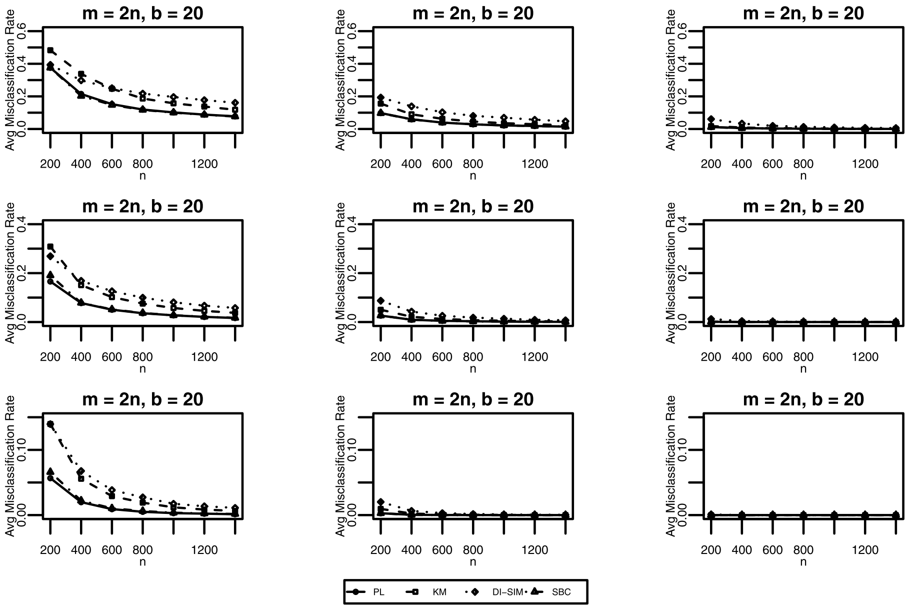

FIG 6. *Average misclassification rates for Bernoulli example over 100 simulations.*

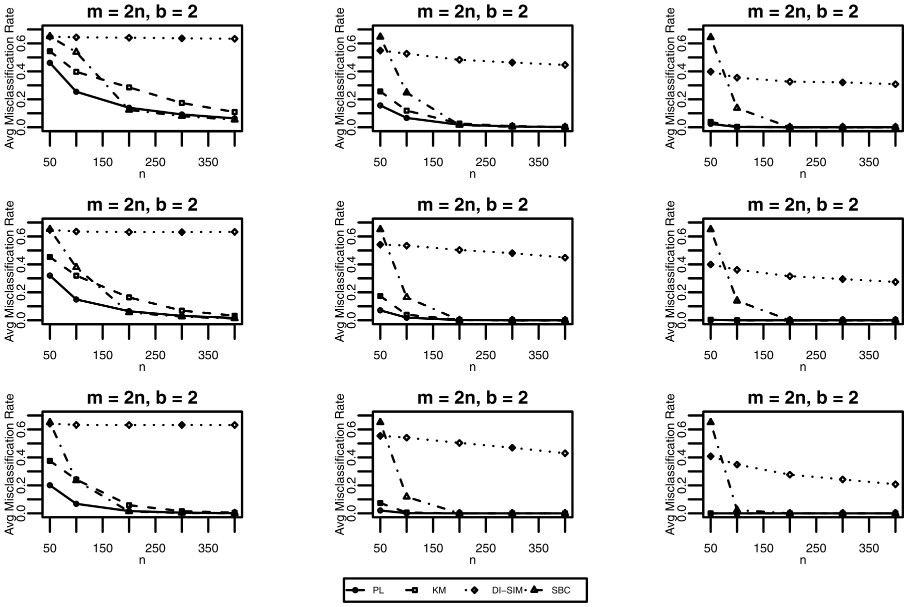

FIG 7. *Average misclassification rates for Gaussian example over 500 simulations.*

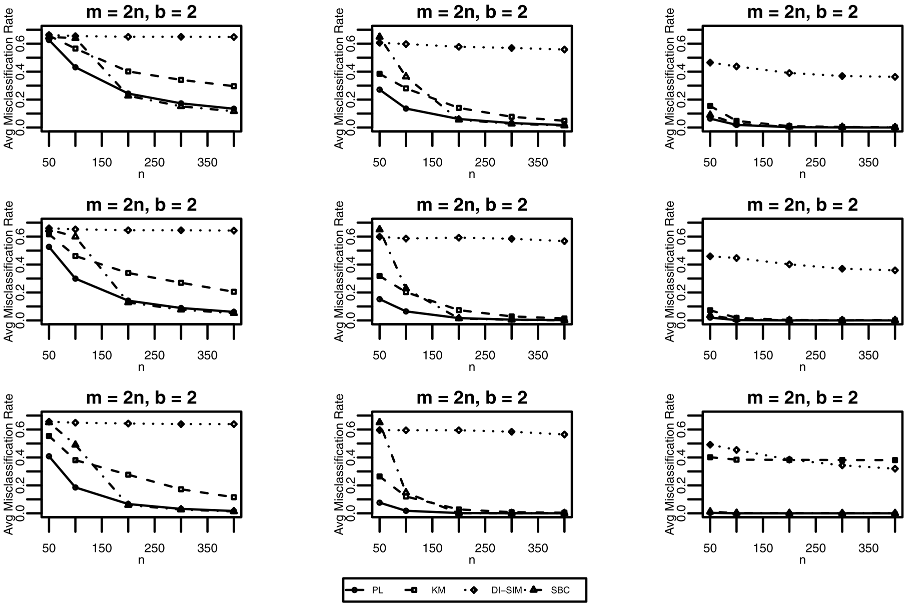

FIG 8. *Average misclassification rates for t example over 500 simulations.*

We set the row and column class membership probabilities as $\boldsymbol{p} = (0.3, 0.7)$ and $\boldsymbol{q} = (0.2, 0.3, 0.5)$. We choose the matrix of block parameters to be

$$\boldsymbol{M} = b \begin{pmatrix} 0.47 & 0.15 & -0.60 \\ -0.26 & 0.82 & 0.80 \end{pmatrix}$$

where the entries were simulated from a uniform distribution on $[-1, 1]$. We vary

TABLE 2
*Standard deviations for Bernoulli example over 100 simulations.*

| $m = 0.5n$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PL | | | KM | | | DS | | | SBC | | |
| n | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 |
| 200 | 0.0834 | 0.0268 | 0.0075 | 0.1141 | 0.0515 | 0.0089 | 0.0760 | 0.0440 | 0.0249 | 0.1193 | 0.0249 | 0.0075 |
| 400 | 0.0473 | 0.0145 | 0.0035 | 0.0932 | 0.0168 | 0.0051 | 0.0400 | 0.0346 | 0.0139 | 0.0417 | 0.0125 | 0.0030 |
| 600 | 0.0224 | 0.0089 | 0.0021 | 0.0654 | 0.0118 | 0.0027 | 0.0347 | 0.0196 | 0.0075 | 0.0199 | 0.0082 | 0.0022 |
| 800 | 0.0168 | 0.0061 | 0.0018 | 0.0266 | 0.0069 | 0.0024 | 0.0273 | 0.0173 | 0.0048 | 0.0135 | 0.0060 | 0.0017 |
| 1000 | 0.0118 | 0.0049 | 0.0010 | 0.0156 | 0.0055 | 0.0014 | 0.0218 | 0.0134 | 0.0033 | 0.0098 | 0.0046 | 0.0010 |
| 1200 | 0.0094 | 0.0041 | 0.0007 | 0.0122 | 0.0052 | 0.0011 | 0.0205 | 0.0114 | 0.0031 | 0.0086 | 0.0041 | 0.0008 |
| 1400 | 0.0079 | 0.0031 | 0.0006 | 0.0100 | 0.0038 | 0.0010 | 0.0161 | 0.0089 | 0.0023 | 0.0076 | 0.0033 | 0.0006 |

| $m = n$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 |
| 200 | 0.0528 | 0.0125 | 0.0020 | 0.1084 | 0.0181 | 0.0031 | 0.0514 | 0.0352 | 0.0104 | 0.0833 | 0.0110 | 0.0018 |
| 400 | 0.0192 | 0.0053 | 0.0006 | 0.0265 | 0.0080 | 0.0012 | 0.0348 | 0.0147 | 0.0034 | 0.0163 | 0.0050 | 0.0005 |
| 600 | 0.0114 | 0.0027 | 0.0003 | 0.0156 | 0.0049 | 0.0007 | 0.0248 | 0.0097 | 0.0018 | 0.0102 | 0.0031 | 0.0003 |
| 800 | 0.0071 | 0.0021 | 0.0002 | 0.0107 | 0.0032 | 0.0003 | 0.0179 | 0.0064 | 0.0011 | 0.0070 | 0.0020 | 0.0001 |
| 1000 | 0.0052 | 0.0013 | 0.0000 | 0.0073 | 0.0022 | 0.0002 | 0.0140 | 0.0046 | 0.0008 | 0.0052 | 0.0013 | 0.0002 |
| 1200 | 0.0043 | 0.0008 | 0.0000 | 0.0067 | 0.0017 | 0.0001 | 0.0120 | 0.0033 | 0.0005 | 0.0040 | 0.0010 | 0.0000 |
| 1400 | 0.0034 | 0.0008 | 0.0000 | 0.0057 | 0.0015 | 0.0001 | 0.0097 | 0.0026 | 0.0005 | 0.0032 | 0.0009 | 0.0001 |

| $m = 2n$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 | b=5 | b=10 | b=20 |
| 200 | 0.0197 | 0.0037 | 0.0000 | 0.0505 | 0.0076 | 0.0000 | 0.0388 | 0.0112 | 0.0025 | 0.0390 | 0.0036 | 0.0000 |
| 400 | 0.0062 | 0.0009 | 0.0000 | 0.0120 | 0.0026 | 0.0000 | 0.0212 | 0.0044 | 0.0003 | 0.0074 | 0.0008 | 0.0000 |
| 600 | 0.0036 | 0.0004 | 0.0000 | 0.0073 | 0.0016 | 0.0000 | 0.0103 | 0.0025 | 0.0000 | 0.0041 | 0.0004 | 0.0000 |
| 800 | 0.0025 | 0.0003 | 0.0000 | 0.0049 | 0.0008 | 0.0000 | 0.0065 | 0.0014 | 0.0001 | 0.0025 | 0.0003 | 0.0000 |
| 1000 | 0.0015 | 0.0000 | 0.0000 | 0.0035 | 0.0005 | 0.0000 | 0.0045 | 0.0011 | 0.0001 | 0.0016 | 0.0001 | 0.0000 |
| 1200 | 0.0012 | 0.0001 | 0.0000 | 0.0029 | 0.0003 | 0.0000 | 0.0041 | 0.0007 | 0.0000 | 0.0012 | 0.0001 | 0.0000 |
| 1400 | 0.0009 | 0.0000 | 0.0000 | 0.0024 | 0.0002 | 0.0000 | 0.0032 | 0.0006 | 0.0000 | 0.0010 | 0.0000 | 0.0000 |

Table 3

*Standard deviations for Gaussian example over 500 simulations.*

| $m = 0.5n$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PL | | | KM | | | DS | | | SBC | | |
| n | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 |
| 50 | 0.1142 | 0.0718 | 0.0306 | 0.0928 | 0.0923 | 0.0491 | 0.0681 | 0.1105 | 0.1472 | 0.0642 | 0.0732 | 0.0836 |
| 100 | 0.0610 | 0.0345 | 0.0051 | 0.0657 | 0.0675 | 0.0074 | 0.0580 | 0.1243 | 0.1374 | 0.1802 | 0.2503 | 0.2487 |
| 200 | 0.0329 | 0.0127 | 0.0005 | 0.0722 | 0.0181 | 0.0005 | 0.0526 | 0.1251 | 0.1049 | 0.0289 | 0.0120 | 0.0003 |
| 300 | 0.0235 | 0.0052 | 0.0000 | 0.0552 | 0.0068 | 0.0000 | 0.0481 | 0.1195 | 0.1029 | 0.0194 | 0.0043 | 0.0000 |
| 400 | 0.0169 | 0.0023 | 0.0000 | 0.0363 | 0.0027 | 0.0000 | 0.0502 | 0.1134 | 0.1200 | 0.0146 | 0.0017 | 0.0000 |

| $m = n$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 |
| 50 | 0.0968 | 0.0503 | 0.0080 | 0.0856 | 0.0975 | 0.0122 | 0.0627 | 0.1171 | 0.1541 | 0.0563 | 0.0596 | 0.0621 |
| 100 | 0.0441 | 0.0175 | 0.0008 | 0.0710 | 0.0343 | 0.0010 | 0.0600 | 0.1193 | 0.1365 | 0.2415 | 0.2503 | 0.2546 |
| 200 | 0.0235 | 0.0030 | 0.0000 | 0.0629 | 0.0046 | 0.0000 | 0.0447 | 0.1165 | 0.1188 | 0.0196 | 0.0025 | 0.0000 |
| 300 | 0.0124 | 0.0007 | 0.0000 | 0.0285 | 0.0011 | 0.0000 | 0.0424 | 0.1166 | 0.1202 | 0.0107 | 0.0005 | 0.0000 |
| 400 | 0.0075 | 0.0002 | 0.0000 | 0.0132 | 0.0003 | 0.0000 | 0.0428 | 0.1147 | 0.1261 | 0.0064 | 0.0001 | 0.0000 |

| $m = 2n$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 |
| 50 | 0.0783 | 0.0235 | 0.0013 | 0.0795 | 0.0753 | 0.0013 | 0.0596 | 0.1041 | 0.1558 | 0.0519 | 0.0519 | 0.0519 |
| 100 | 0.0323 | 0.0040 | 0.0000 | 0.0839 | 0.0086 | 0.0000 | 0.0507 | 0.1084 | 0.1459 | 0.2528 | 0.2358 | 0.0837 |
| 200 | 0.0106 | 0.0004 | 0.0000 | 0.0310 | 0.0006 | 0.0000 | 0.0409 | 0.1153 | 0.1329 | 0.0094 | 0.0002 | 0.0000 |
| 300 | 0.0045 | 0.0000 | 0.0000 | 0.0085 | 0.0000 | 0.0000 | 0.0353 | 0.1172 | 0.1416 | 0.0035 | 0.0000 | 0.0000 |
| 400 | 0.0020 | 0.0000 | 0.0000 | 0.0041 | 0.0000 | 0.0000 | 0.0354 | 0.1177 | 0.1469 | 0.0014 | 0.0000 | 0.0000 |

Table 4

*Standard deviations for t example over 500 simulations.*

| $m = 0.5n$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PL | | | KM | | | DS | | | SBC | | |
| n | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 |
| 50 | 0.0958 | 0.1018 | 0.0526 | 0.0792 | 0.1028 | 0.1176 | 0.0662 | 0.0845 | 0.1426 | 0.0643 | 0.0669 | 0.1000 |
| 100 | 0.0877 | 0.0485 | 0.0175 | 0.0884 | 0.0871 | 0.0775 | 0.0524 | 0.0838 | 0.1394 | 0.0694 | 0.2431 | 0.0535 |
| 200 | 0.0429 | 0.0233 | 0.0040 | 0.0620 | 0.0892 | 0.0430 | 0.0409 | 0.0836 | 0.1138 | 0.0441 | 0.0227 | 0.0239 |
| 300 | 0.0298 | 0.0137 | 0.0013 | 0.0620 | 0.0831 | 0.0322 | 0.0370 | 0.0880 | 0.1020 | 0.0228 | 0.0116 | 0.0232 |
| 400 | 0.0235 | 0.0093 | 0.0007 | 0.0640 | 0.0769 | 0.0192 | 0.0335 | 0.0922 | 0.0910 | 0.0189 | 0.0078 | 0.0126 |

| $m = n$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 |
| 50 | 0.1073 | 0.0724 | 0.0229 | 0.0798 | 0.0895 | 0.1034 | 0.0561 | 0.0877 | 0.1439 | 0.0561 | 0.0574 | 0.0762 |
| 100 | 0.0690 | 0.0344 | 0.0052 | 0.0856 | 0.1021 | 0.0633 | 0.0452 | 0.0808 | 0.1384 | 0.1358 | 0.2477 | 0.0416 |
| 200 | 0.0324 | 0.0108 | 0.0009 | 0.0659 | 0.0979 | 0.0298 | 0.0372 | 0.0691 | 0.1206 | 0.0444 | 0.0097 | 0.0130 |
| 300 | 0.0220 | 0.0048 | 0.0003 | 0.0668 | 0.0706 | 0.0194 | 0.0298 | 0.0761 | 0.1006 | 0.0176 | 0.0039 | 0.0003 |
| 400 | 0.0155 | 0.0024 | 0.0002 | 0.0827 | 0.0556 | 0.0134 | 0.0278 | 0.0846 | 0.1032 | 0.0125 | 0.0019 | 0.0134 |

| $m = 2n$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 | b=0.5 | b=1 | b=2 |
| 50 | 0.0976 | 0.0514 | 0.0086 | 0.0843 | 0.1057 | 0.0659 | 0.0520 | 0.0828 | 0.1343 | 0.0519 | 0.0519 | 0.0484 |
| 100 | 0.0528 | 0.0147 | 0.0007 | 0.0774 | 0.1141 | 0.0440 | 0.0443 | 0.0702 | 0.1357 | 0.2231 | 0.2397 | 0.0225 |
| 200 | 0.0211 | 0.0035 | 0.0002 | 0.0752 | 0.0807 | 0.0288 | 0.0342 | 0.0646 | 0.1189 | 0.0191 | 0.0025 | 0.0139 |
| 300 | 0.0114 | 0.0009 | 0.0000 | 0.0989 | 0.0469 | 0.0216 | 0.0274 | 0.0746 | 0.0965 | 0.0096 | 0.0007 | 0.0000 |
| 400 | 0.0068 | 0.0003 | 0.0000 | 0.1086 | 0.0361 | 0.0176 | 0.0260 | 0.0868 | 0.0938 | 0.0058 | 0.0003 | 0.0000 |

$b$ between 0.5 and 2. We generate the data conditional on the row and column classes as $X_{ij} \mid \boldsymbol{c}, \boldsymbol{d} \sim \text{Gaussian}(\mu_{c_i d_j}, \sigma = 1)$. We initialize all the methods with 100 random starting values.

For the non-standardized Student's $t$ simulation, we use the same parameters as in the Gaussian simulation and we generate the data conditional on the row and column classes as $X_{ij} \mid \boldsymbol{c}, \boldsymbol{d} \sim \text{t}(\mu_{c_i d_j}, \sigma = 1)$ with four degrees of freedom. We initialize all the methods with 100 random starting values.

Similar to the Poisson simulation, biclustering based on the profile log-likelihood criterion performs at least as well as the other methods and shows signs of convergence in all three examples. These results provide further verification of the theoretical findings and support the use of biclustering based on the profile log-likelihood criterion.

### *B.2. Additional application - MovieLens*

Since consumer habits likely vary depending on products, biclustering review-website data can help identify structure in the data and identify groups of consumers and groups of products with similar patterns. As an application of

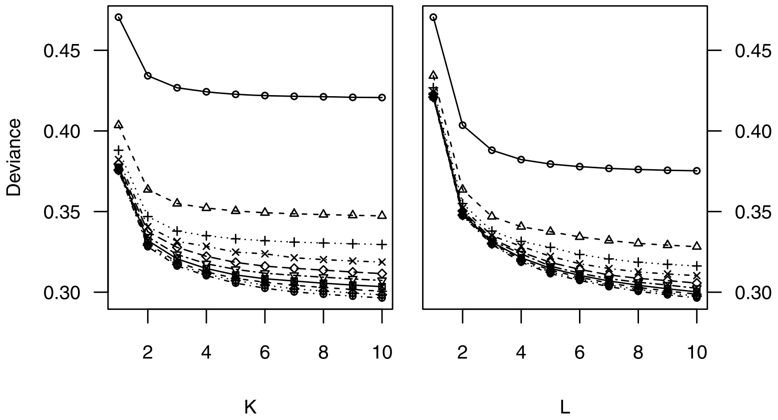

FIG 9. *MovieLens likelihood for different values of $K$ and $L$.*

this we apply biclustering to the MovieLens dataset [21].

The MovieLens dataset consists of 100,000 movie reviews on 1682 movies by 943 users. Each user has rated at least 20 movies and each movie is rated on a scale from one to five. In addition to the review rating, the release date and genre of each movie is available as well as some demographic information about each user including gender, age, occupation and zip code.

In order to retain customers, movie-renting services strive to recommend new movies to individuals who are likely to view them. Since most users only review movies that they have already seen, we can use the structure of the user-movie review matrix to identify associations between users and viewing habits of movies. Specifically, we consider the 943×1682 binary matrix $\boldsymbol{X}$ where $X_{ij} = 1$ if user $i$ has rated movie $j$ and $X_{ij} = 0$ otherwise. To find structure in $\boldsymbol{X}$, we biclustered the rows and columns of $\boldsymbol{X}$ using the profile likelihood based on the Bernoulli criterion (6.1a).

To determine a reasonable selection for the number of biclusters we varied the number of user groups, $K$, and the number of movie groups, $L$, each from 1 to 10. For each combination of $K$ and $L$, we computed the optimal cluster assignments based on 250 random starting values. Figure 9 presents the scree plots as functions of $K$ and $L$. From the left scree plot, we see little change to the deviance when increasing $L$ beyond 4. From the right scree plot, the deviance does not decrease much beyond when increasing $K$ beyond 3. Based on these two plots, we set $K = 3$ and $L = 4$. For $K = 3$ and $L = 4$, we experimented with using up to 2000 random starting values, but found no change to the resulting log-likelihood.

We compared the resulting bicluster assignments to those found by DI-SIM, KM and SBC when $K = 3$ and $L = 4$. The cluster assignments varied between the four methods, with the least amount of disagreement between PL and SBC (73.0% of cluster assignments agreed) and the most disagreement between Di-Sim and the profile-likelihood method (24.2% of clusters agreed).

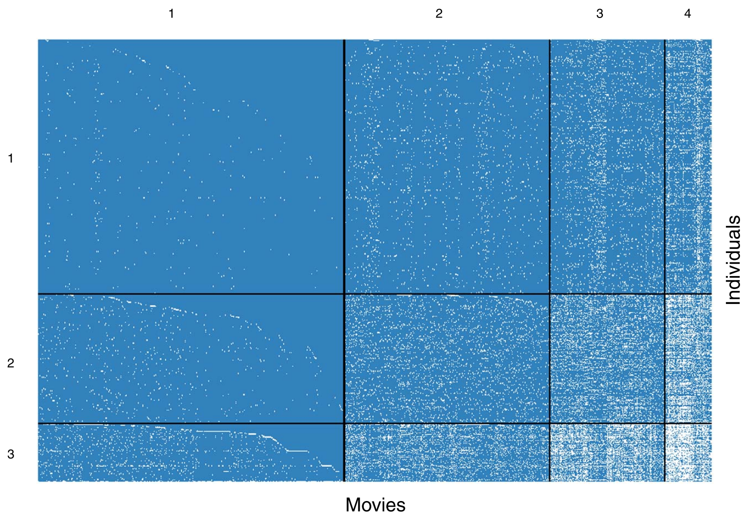

Fig 10. *Heatmap generated from MovieLens data reflecting the varying review patterns in the different biclusters. Blue identifies movies with no review and white identifies rated movies.*

Using the built-in package functions, SBC and CBC select different values of $K$ and $L$. SBC selects $K = 13$ and $L = 14$. The cluster assignments result in a Rand Index of 0.842 and 0.650 when compared to the PL row and column cluster assignments, respectively. CBC selected 1 row cluster and 1,519 column clusters, where many of the column clusters only contained one row/column. CBC is designed for continuous data and the default settings do not appear to produce meaningful clusters in this example.

Figure 10 presents the heatmap of the data based on the resulting bicluster assignments from the profile-likelihood method, with the ordering of the clusters determined by the total number of a reviews in each cluster. Roughly speaking, user group 3 is consistently active across all movie groups with increasing activity as the popularity of the movie increases. The reviewing habits of user group 2 follow a similar pattern but to a lesser extent. In contrast, user group 1 is consistently inactive with the only exceptions being movie group 4.

The median ages within the user group were 33, 30, and 29, and the percentages of male users within each group were 68.7%, 72.8%, and 77.4%. These statistics suggest that there is some age and gender effect on the reviewing habits of the users.

Table 5 reports the top ten movies in each group. The eclectic mix of genres within each movie group suggests that the rating behavior of users is not explained by genre alone.

Figure 11 presents a boxplot comparing the distributions of the movie release years for each group. We can see a clear ordering of the movie groups by median release date. It appears that the users in all three groups rate movies from all time periods, but reviewing behavior varies based on movie popularity. The

TABLE 5
*The top ten movies in each cluster based on the total number of reviews.*

| Group 1 | Group 2 |
|---|---|
| Mrs. Parker and the Vicious Circle (1994) | Santa Clause, The (1994) |
| Miserables, Les (1995) | Sleeper (1973) |
| Lawnmower Man 2: Beyond Cyberspace (1996) | Sword in the Stone, The (1963) |
| Richie Rich (1994) | Cook the Thief His Wife & Her Lover, The (1989) |
| Candyman: Farewell to the Flesh (1995) | Somewhere in Time (1980) |
| Ice Storm, The (1997) | Mulholland Falls (1996) |
| Funny Face (1957) | Crumb (1994) |
| Umbrellas of Cherbourg, The (1964) | I.Q. (1994) |
| My Family (1995) | Legends of the Fall (1994) |
| Top Hat (1935) | Alice in Wonderland (1951) |

| Group 3 | Group 4 |
|---|---|
| Beauty and the Beast (1991) | Star Wars (1977) |
| Batman (1989) | Contact (1997) |
| Young Frankenstein (1974) | Fargo (1996) |
| Star Trek IV: The Voyage Home (1986) | Return of the Jedi (1983) |
| Citizen Kane (1941) | Liar Liar (1997) |
| Fifth Element, The (1997) | English Patient, The (1996) |
| Gandhi (1982) | Scream (1996) |
| Face/Off (1997) | Toy Story (1995) |
| Dr. Strangelove Or (1963) | Air Force One (1997) |
| Tin Cup (1996) | Independence Day (ID4) (1996) |

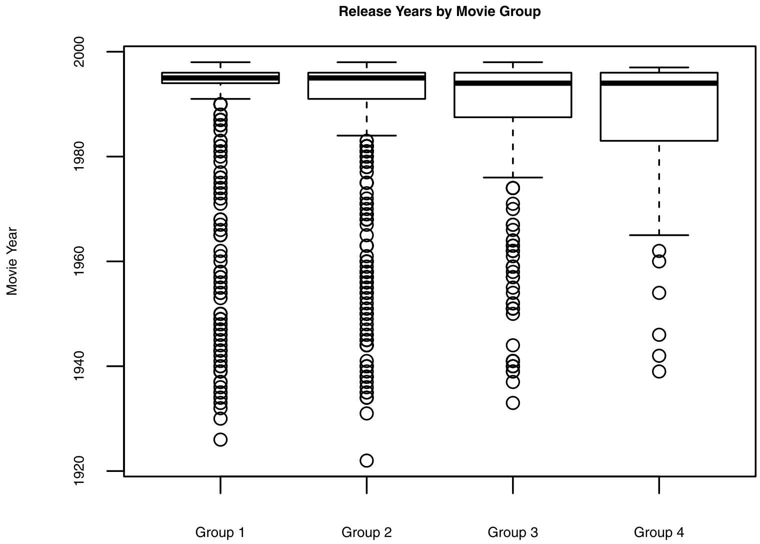

FIG 11. *Boxplot comparing the different clusters based on movie release dates.*

biclusters here suggest that individuals in group 3 are more likely to rate under-reviewed movies, whereas individuals in group 1 primarily rate popular movies.

### *B.3. Algorithm stability analysis*

For each of the three data applications (GovTrack, AGEMAP, and MovieLens), we report the local optima found after running our biclustering method with 1000 random initializations. The number of local optima (“modes”) found in each of the three data sets are 4, 3, and 10, respectively.

For each of the modes, we report the log-likelihood value ("Log-Lik."), the number of replicates that found the mode ("Count"), and the row and column cluster differences ("Row Diff." and "Column Diff.") when compared to the best mode out of the 1000 replicates. In reporting the difference we give both the absolute difference ("Abs."), the number of rows or columns with different labels, and the relative difference ("Rel."), the proportion of rows or columns with different labels.

Even in the AGEMAP application, where the best local optimum was rare, the results of the algorithm are very stable, giving consistent cluster assignments for over 98% of the rows and columns. We would not recommend using our algorithm with just a single random initialization, but it appears that using 100–1000 random initializations suffices to give reliable results.

*GovTrack*

The algorithm found the best local optimum 36.8% of the time. The result of the algorithm agreed with the best local optimum found in all of the row labels and at least 98.7% of the column labels 99.9% of the time.

| Mode | Log-Lik. | Count | Row Diff. | | Column Diff. | |
|---|---|---|---|---|---|---|
| | | | Abs. | Rel. (%) | Abs. | Rel. (%) |
| 1 | −36246 | 368 | – | – | – | – |
| 2 | −36248 | 27 | 0 | 0 | 2 | 0.4 |
| 3 | −36430 | 604 | 0 | 0 | 6 | 1.3 |
| 4 | −44732 | 1 | 0 | 0 | 70 | 15.4 |

444 rows, 545 columns; $K = 2$ row clusters, $L = 4$ column clusters.

*AGEMAP*

In all cases, the value of the log-likelihood found was extremely similar, though the best local optimum was found only 0.2% of the time. The row clusters were always the same, and at least 99.8% of the column labels agreed in all reported local optima.

| Mode | Log-Lik. | Count | Row Diff. | | Column Diff. | |
|---|---|---|---|---|---|---|
| | | | Abs. | Rel. (%) | Abs. | Rel. (%) |
| 1 | 7408.883 | 2 | – | – | – | – |
| 2 | 7408.881 | 909 | 0 | 0 | 18 | 0.1 |
| 3 | 7408.872 | 89 | 0 | 0 | 34 | 0.2 |

39 rows, 17864 columns; $K = 3$ row clusters, $L = 5$ column clusters.

*MovieLens*

There was much more variability in the output than in the other two applications, though the first 5 modes (which appeared in 99.1% of the replicates) had very similar log-likelihood values, and agreed for over 98.1% of their row labels and over 97.9% of their column labels.

| Mode | Log-Lik. | Count | Row Diff. | | Column Diff. | |
|---|---|---|---|---|---|---|
| | | | Abs. | Rel. (%) | Abs. | Rel. (%) |
| 1 | −262910 | 137 | – | – | – | – |
| 2 | −262915 | 54 | 18 | 1.9 | 1 | 0.6 |
| 3 | −262916 | 3 | 0 | 0.0 | 19 | 1.1 |
| 4 | −262919 | 39 | 17 | 1.8 | 13 | 0.8 |
| 5 | −262923 | 758 | 3 | 0.3 | 36 | 2.1 |
| 6 | −263892 | 4 | 142 | 15.1 | 310 | 18.4 |
| 7 | −264085 | 1 | 87 | 9.2 | 294 | 17.5 |
| 8 | −265183 | 2 | 56 | 5.9 | 417 | 24.8 |
| 9 | −265183 | 1 | 57 | 6.0 | 416 | 24.7 |
| 10 | −265185 | 1 | 62 | 6.6 | 418 | 24.9 |

943 rows, 1682 columns; $K = 3$ row clusters, $L = 4$ column clusters.

## References

[1] Abbe, E. (2018). Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18(177):1–86. MR3827065

[2] Ames, B. P. (2014). Guaranteed clustering and biclustering via semidefinite programming. *Math. Program.*, 147(1–2):429–465. MR3258531

[3] Amini, A. A., Chen, A., Bickel, P. J., and Levina, E. (2013). Pseudo-likelihood methods for community detection in large sparse networks. *Ann. Statist.*, 41(4):2097–2122. MR3127859

[4] Amini, A. A. and Levina, E. (2018). On semidefinite relaxations for the block model. *Ann. Statist.*, 46(1):149–179. MR3766949

[5] Arabie, P., Boorman, S. A., and Levitt, P. R. (1978). Constructing blockmodels: How and why. *J. Math. Pscyh.*, 17(1):21–63.

[6] Bickel, P., Choi, D., Chang, X., and Zhang, H. (2013). Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *Ann. Statist.*, 41(4):1922–1943. MR3127853

[7] Bickel, P. J. and Chen, A. (2009). A nonparametric view of network models and Newman-Girvan and other modularities. *Proc. Nat. Acad. Sci. USA*, 106:21068–21073.

[8] Brown, L. D. (1986). *Fundamentals of Statistical Exponential Families with Applications in Statistical Decision Theory*, volume 9 of *Lecture Notes – Monograph Series*. Institute of Mathematical Statistics, Hayward, CA. MR0882001

[9] Celisse, A., Daudin, J.-J., and Pierre, L. (2012). Consistency of maximum-likelihood and variational estimators in the stochastic block model. *Electron. J. Statist.*, 6:1847–1899. MR2988467

[10] Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. In *Proceedings International Conference on Intelligent Systems for Molecular*

*Biology; ISMB. International Conference on Intelligent Systems for Molecular Biology*, 8:93–103.

[11] Chi, E. C., Allen, G. I., and Baraniuk, R. G. (2017). Convex biclustering. *Biometrics*, 73(1):10–19. MR3632347

[12] Choi, D. and Wolfe, P. J. (2014). Co-clustering separately exchangeable network data. *Annals of Statistics*, 42:29–63. MR3161460

[13] Choi, D., Wolfe, P. J., and Airoldi, E. M. (2012). Stochastic blockmodels with growing number of classes. *Biometrika*, 99:273–284. MR2931253

[14] Daudin, J. J., Picard, F., and Robin, S. (2008). A mixture model for random graphs. *Statistics and Computing*, 18:173–183. MR2390817

[15] Dhillon, I. S. (2001). 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 (KDD)*, pages 26–29, San Francisco.

[16] Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc. Nat. Acad. Sci. USA*, 95(25):14863–14868.

[17] Fishkind, D., Sussman, D., Tang, M., Vogelstein, J., and Priebe, C. (2012). Consistent adjacency-spectral partitioning for the stochastic block model when the model parameters are unknown. *SIAM Journal on Matrix Analysis and Applications*, 34. MR3032990

[18] Gao, C., Lu, Y., Ma, Z., and Zhou, H. H. (2016). Optimal estimation and completion of matrices with biclustering structures. *J. Mach. Learn. Res.*, 17(1):5602–5630. MR3569248

[19] Getz, G., Levine, E., and Domany, E. (2000). Coupled two-way clustering analysis of gene microarray data. *Proc. Nat. Acad. Sci. USA*, 97:12079–12084.

[20] Golub, G. H. and Loan, C. F. V. (1996). *Matrix Computation*. Johns Hopkins University Press. MR1417720

[21] GroupLens (1998). MovieLens Dataset. http://grouplens.org/datasets/movielens/.

[22] Guédon, O. and Vershynin, R. (2014). Community detection in sparse networks via grothendieck's inequality. *Probability Theory and Related Fields*, 165:1025–1049. MR3520025

[23] Harpaz, R., Perez, H., Chase, H. S., Rabadan, R., Hripcsak, G., and Friedman, C. (2010). Biclustering of Adverse Drug Events in the FDA's Spontaneous Reporting System. *Clinical Pharmacology & Therapeutics*, 89(2):243–250.

[24] Hartigan, J. A. (1972). Direct clustering of a data matrix. *J. Amer. Statist. Assoc.*, 67(337):123–129.

[25] Hofmann, T. (1999). Latent class models for collaborative filtering. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 688–693.

[26] Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, 5:109–137. MR0718088

[27] Jin, J. (2015). Fast community detection by score. *Ann. Statist.*, 43(1):57–

89. MR3285600

[28] Kernighan, B. W. and Lin, S. (1970). An Efficient Heuristic Procedure for Partitioning Graphs. *The Bell system technical journal*, 49(1):291–307.

[29] Kluger, Y., Basri, R., Chang, J. T., and Gerstein, M. (2003). Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research*, 13:703–716.

[30] Lazzeroni, L. and Owen, A. (2002). Plaid models for gene expression data. *Statist. Sinica*, 12:61–86. MR1894189

[31] Lei, J. and Rinaldo, A. (2015). Consistency of spectral clustering in stochastic block models. *Ann. Statist.*, 43(1):215–237. MR3285605

[32] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif. University of California Press. MR0214227

[33] Madeira, S. C. and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE T. Comput. Bi.*, 1:24–45.

[34] Mariadassou, M. and Matias, C. (2015). Convergence of the groups posterior distribution in latent or stochastic block models. *Bernoulli*, 1:537–573. MR3322330

[35] Mirkin, B. (1996). *Mathematical classification and clustering*. Kluwer Academic Press. MR1480413

[36] Mossel, E., Neeman, J., and Sly, A. (2016). Belief propagation, robust reconstruction and optimal recovery of block models. *Ann. Appl. Probab.*, 26(4):2211–2256. MR3543895

[37] Murphy, S. A. and van der Vaart, A. W. (2000). On Profile Likelihood. *J. Amer. Statist. Assoc.*, 95(450):449–465. MR1803168

[38] Newman, M. E. J. (2006). Modularity and community structure in networks. *Proc. Nat. Acad. Sci. USA*, 103(23):8577–8582. MR2676073

[39] Perry, P. O. and Owen, A. B. (2010). A rotationn test to verify latent structure. *J. Mach. Learn. Res.*, 11:603–624. MR2600622

[40] Perry, P. O. and Wolfe, P. J. (2012). Null models for network data. Preprint arXiv:1201.5871.

[41] Razaee, Z. S., Amini, A. A., and Li, J. J. (2019). Matched bipartite block model with covariates. *Journal of Machine Learning Research*, 20:1–44. MR3948074

[42] Rohe, K., Chatterjee, S., and Yu, B. (2011). Spectral clustering and the high-dimensional stochastic blockmodel. *Ann. Statist.*, 39(4):1878–1915. MR2893856

[43] Rohe, K. and Yu, B. (2012). Co-clustering for directed graphs; the stochastic co-blockmodel and a spectral algorithm. Preprint arXiv:1204.2296.

[44] Seldin, Y. and Tishby, N. (2009). Pac-bayesian generalization bound for density estimation with application to co-clustering. In *12th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[45] Seldin, Y. and Tishby, N. (2010). Pac-bayesian analysis of co-clustering and beyond. *Journal of Machine Learning Research*, 11:3595–3646. MR2756195

[46] Tan, K. M. and Witten, D. M. (2014). Sparse biclustering of transposable

data. *Journal of Computational and Graphical Statistics*, 23(4):985–1008. MR3270707

[47] Tanay, A., Sharan, R., and Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18 Suppl 1:S136–44.

[48] Tarpey, T. and Flury, B. (1996). Self-consistency: A fundamental concept in statistics. *Statist. Sci.*, 11(3):229–243. MR1436648

[49] Ungar, L. and Foster, D. P. (1998). A formal statistical approach to collaborative filtering. In *CONALD'98*.

[50] Varadhan, S. R. S. (2001). *Probability Theory (Courant Lecture Notes)*. American Mathematical Society. MR1852999

[51] Zahn, J. M., Poosala, S., Owen, A. B., Ingram, D. K., Lustig, A., Carter, A., Weeraratna, A. T., Taub, D. D., Gorospe, M., Mazan-Mamczarz, K., Lakatta, E. G., Boheler, K. R., Xu, X., Mattson, M. P., Falco, G., Ko, M. S. H., Schlessinger, D., Firman, J., Kummerfeld, S. K., Wood, W. H., Zonderman, A. B., Kim, S. K., and Becker, K. G. (2007). AGEMAP: A Gene Expression Database for Aging in Mice. *PLOS Genetics*.

[52] Zhao, Y. (2017). A survey on theoretical advances of community detection in networks. *ArXiv*, abs/1809.07691. MR3696558

[53] Zhao, Y., Levina, E., and Zhu, J. (2011). Community extraction for social networks. *P. Natl. Acad. Sci. USA*, 108:7321–7326.

[54] Zhao, Y., Levina, E., and Zhu, J. (2012). Consistency of community detection in networks under degree-corrected stochastic block models. *Ann. Stat.*, 40(4):2266–2292. MR3059083