Department of Computer Science
University of Toronto
http://learning.cs.toronto.edu

6 King's College Rd, Toronto
M5S 3G4, Canada
fax: +1 416 978 1455

June 22, 2007

UTML TR 2007–001

Nonparametric Bayesian Biclustering

Edward Meeds and Sam Roweis
Department of Computer Science, University of Toronto

Abstract

We present a probabilistic block-constant biclustering model that simultaneously clusters rows and columns of a data matrix. All entries with the same row cluster and column cluster form a bicluster. Each cluster is part of a mixture having a nonparametric Bayesian prior. The number of biclusters is therefore treated as a nuisance parameter and is implicitly integrated over during simulation. Missing entries are completely integrated out of the model, allowing us to completely bipass the common requirement for biclustering algorithms that missing values be filled before analysis, but also makes it robust to high rates of missing values. By using a Gaussian model for the density of entries in bliclusters, an efficient sampling algorithm is produced because bicluster parameters are analytically integrated out. We present several inference procedures for sampling cluster indicators, including Gibbs and split-merge moves. We show that our method is competitive, if not superior, to existing imputation methods, especially for high missing rates, despite imputing constant values for entire blocks of data. We present imputation experiments and exploratory biclustering results.

Nonparametric Bayesian Biclustering

Edward Meeds and Sam Roweis
Department of Computer Science, University of Toronto

1 Biclustering

Biclustering (also known as co-clustering or 2-way clustering) refers to the the simultaneous grouping of rows and columns of a data matrix. Each bicluster is a submatrix of the full (possibly reordered) data matrix and entries in a bicluster should have some coherent structure (the details of which depend on the method employed). This coherence could be, for example, constant values for all entries in the submatrix, or similar row/column patterns within a bicluster. Biclustering algorithms are also characterized by how the rows and columns are assigned to clusters. Rows/columns can either belong to multiple clusters (as shown in Figure 2A & 2B) or to only a single cluster (as shown in 2C); clusters can overlap (2A) or not (2B & 2C). Some matrix entries may also belong to a "background" noise model which is not part of any bicluster (2A & 2B). Most representations assume that there exists a single permutation of the matrix rows/columns after which all the biclusters are contiguous blocks. (Matrix tile analysis [4] is an exception.) Our approach produces biclusters like those in Figure 2C: each row and column belongs to a single, non-overlapping cluster.

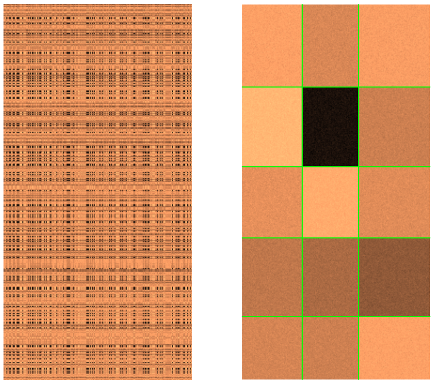

Figure 1: **Left**: Original data. **Right**: Data after biclustering.

Assessing the significance of partitions discovered by biclustering is problematic for several reasons. First, there are few available data sets which are annotated with ground truth partitions. Second, those that are annotated may have partitions that do not correspond to any possible result of the clustering algorithm. Third, most algorithms have parameters which modify the scale/size of partitions which are discovered. Deciding which scale is best in a purely unsupervised manner is difficult and poorly defined. For example, a common goal when clustering microarray data is to group genes and/or experiments in such a way that the partitions are biologically "significant" or "plausible"; often this is assessed by examining the clusters by hand [2, 8].

Several modeling issues must be addressed by any biclustering method. The most important is a method for assessing when a partition represents a significant bicluster. This is closely related to the choice of the number of clusters to use. Most biclustering use greedy procedures for fitting biclusters one at a time until either a global fitting objective or a pre-specified number of clusters has been reached [2, 8]. Of course, if the only objective is to reduce some measure of fitting residual, overfitting will occur unless the model is highly regularized, especially in very flexible non-probabilistic models. By restricting the permissible types of clusters we can control capacity; we can also use a probabilistic model of the data and use marginal likelihood as a guide. The biclustering algorithm that we present here is a fully probabilistic model which uses Bayesian nonparametric priors over row and column clusters. This allows us to treat the number of biclusters as a nuisance parameter and implicitly integrate it out

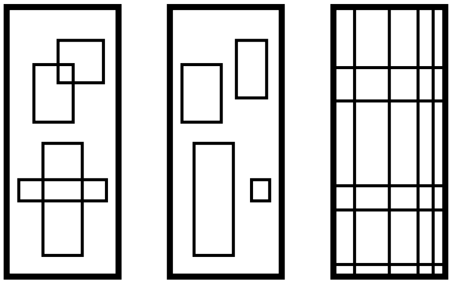

Figure 2: **A**: Multiple, overlapping biclusters. **B**: Multiple, non-overlapping biclusters. **C**: Single, non-overlapping biclusters.

during inference. In fact, the algorithm we present in this paper does not return a single partition, but a distribution over partitions, including groupsings at different scales.

A final important issue is missing values in the data matrix. Many biclustering procedures require complete data matrices, and therefore data with missing values must use an imputation algorithm as a preprocessing step. Our model completely integrates over missing values,[1] avoiding ad hoc preprocessing which undoubtably affect clustering results for other methods.

The remainder of the paper is organized as follows. In Section 2 we give a general description of our model, including a high-level description and an introduction to Bayesian nonparametric priors for clustering models. General inference algorithms for both small moves in state-space (Gibbs) and large moves (split-merge) are presented in Section 4. We follow with a concrete example of a generative model of data using a Gaussian-Gamma prior for bicluster parameters. In Section 6 we present two types of experiments. First, imputation experiments to compare our method with other methods designed for imputation, not biclustering. This is a common way of assessing unsupervised algorithms. Second, cluster analysis experiments on gene expression, text, and collaberative filtering data. Finally, we discuss and conclude in Section 7.

2 A Bayesian biclustering model

Our new model can be thought of as an infinite mixture of very simple biclusterings in which each row belongs to exactly one of K row clusters and each column to exactly one of L column clusters. The novel contribution is that we incorporate a flexible, fully Bayesian, non-parametric prior over row and column partitions and implicitly average over partitions according to their posterior probabilities given the observed data. This is achieved using Markov Chain Monte Carlo (MCMC) sampling , which causes the number of row and column clusters to change during inference (such dynamics will be explained in more detail in Section 2.1). For any particular setting of the row and column cluster assignments, the density of entries in a bicluster (i.e. the subset of rows and columns having a particular joint setting of cluster assignments) is governed by a set of parameters indexed by both the row and column cluster.

To perform imputation (filling in) or cluster analysis with out model, we first run many iterations of MCMC inference, gathering samples of partitions at each iteration (after discarding burn-in samples). We can then compute quantities of interest by averaging over these samples.For imputation, this means averaging over predictions for missing values; for cluster analysis we average partitions by forming a symmetric neighbourhood graph in which the weight of the edge between i and j is fraction of partitions in which i were found in the same cluster or bicluster. (The objects i, j may be rows, columns or individual entries.)

2.1 Nonparametric prior over partitions

In Bayesian (or MAP) mixture modeling with finite mixtures (which can be used for either soft or hard partitionings of N objects into K clusters), Dirichlet distributions are often used as priors for the mixture weights, which has the effect of smoothing the maximum likelihood mixture distributions. If the number of clusters K is unknown, one common procedure for selecting its value is to chose the K which maximizes the likelihod of held-out data. A more Bayesian approach would put a prior probability distribution over K and weight with different K by their posterior probabilities given the observed data.

[1] We assume entries are missing completely at random (MCAR).

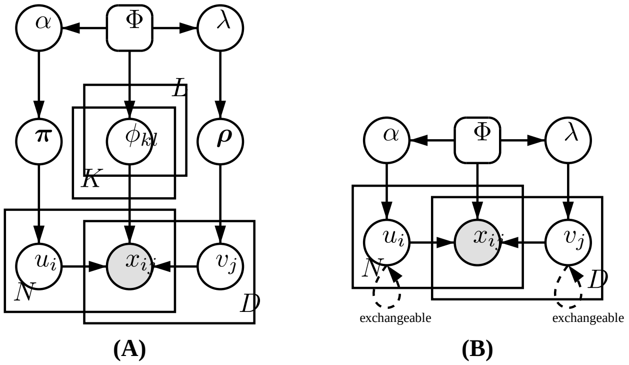

Figure 3: **(A)** Graphical representation of the general BIC model, with using a stick-breaking representation and with cluster parameters ϕ_{kl} explicitly represented. **(B)** The same model but using a Polya urn representation and the cluster parameters explicitly integrated out of the model. We have use a dashed line to represent exchangeability.

Bayesian nonparametric priors are elegant and computationally efficient ways of incorporating prior belief about the distribution over K into our probabilistic mixture models. The Dirichlet process (DP) prior [3] is the most common, and has the appealing property that any sample from a DP is Dirichlet distributed, making it a natural prior for component weights in a mixture model. Component indicators can be sampled directly by first sampling the weights or the weights can be integrated away, and indicators can be sampled marginally using the Polya urn scheme. The Polya urn distribution is exchangeable and has prior mass α allocated to an infinite number of uninstantiated components that do not yet exist in the mixture.This means that new components can be added to the mixture with positive probability. The Pitman-Yor process is a generalization of the DP with an additional parameter d $(0 <= d < 1)$ which *discounts* occupied clusters and has the effect of producing more uniform cluster sizes in general. (For $d = 0$, the prior reduced to a DP.) In our experiments we employ the PY prior[2] because it is more flexible than the DP.

3 General probability model for Bayesian biclustering

We now describe the full probability model for a general parametric model for bicluster densities. In Section 5 we will describe a model with a Gaussian bicluster density.

Let u_i be an indicaor variable such that $u_i = k$ if the ith object belongs to component k. The distribution hierarchy is as follows:

$$\begin{aligned} x_{ij}|u_i, v_j, \phi_{u_i v_j} &\sim \mathrm{F}\left(x_{ij}|\phi_{u_i v_j}\right) \\ u_i|\alpha, d_u, \mathbf{u}^{-i} &\sim \mathrm{PY}\left(u_i|\alpha, d_u, \mathbf{u}^{-i}\right) \\ v_j|\lambda, d_v, \mathbf{v}^{-j} &\sim \mathrm{PY}\left(v_j|\lambda, d_v, \mathbf{v}^{-j}\right) \\ \phi_{u_i v_j}|\Phi &\sim G_0\left(\phi_{u_i v_j}|\Phi\right) \end{aligned}$$

where λ is the concentration parameter for $\mathbf{v}$, and d_u and d_v are discount parameters for $\mathbf{u}$ and $\mathbf{v}$, respectively. Hyperparameters Θ do not depend on cluster parameters. Each entry x_{ij} of a data matrix X is distributed according to a parametric density model with a set of parameters $\phi_{u_i v_j}$, where u_i is an indicator variable which indexes into a set of row clusters and v_j is the equivalent indicator for column clusters. Cluster indicators have independent PY priors with their own concetration and discount parameters. Bicluster parameters have their own base distribution G_0. One could model $\phi_{u_i v_j}$ as $\phi_{u_i} + \phi_{v_j}$, for example, but we have a single set of parameters for each bicluster. When G_0 is conjugate to F, this enables us to integrate the parameters out of the state of the Markov chain. The graphical model of the hierarchy is shown in Figure 3. On the left the weights sampled from a PY process are shown explicitly. On the right we show the model with the weights integrated out.

It is useful to rewrite the data likelihood from a product of univaraite entries as a product of blocks of multivariate data:

$$P\left(\mathbf{X}|\mathbf{u}, \mathbf{v}, \phi\right) \sim \prod_{kl}\prod_{ij} \mathrm{F}\left(x_{ij}|\phi_{kl}\right)^{\delta_k(u_i)\delta_l(v_j)}$$

[2] By putting Gamma and Beta priors over α and d, respectively, we can sample the PY during simulation (see Section 4.3).

$$= \prod_{kl} \mathrm{F}\left(\mathbf{x}_{kl}|\phi_{kl}\right)$$

where $\mathbf{x}_{kl}$ is the vector of all entries in $\mathbf{X}$ which have $u_i = k$ and $v_j = l$. In this paper we will be mostly interested in the case where F is conjugate to G_0, which allows us to intergate out all ϕ_{kl}:

$$\begin{aligned} P\left(\mathbf{X}|\mathbf{u},\mathbf{v},\Phi\right) &\sim \prod_{kl} \int d\phi_{kl} \mathrm{F}\left(\mathbf{x}_{kl}|\phi_{kl}\right) G_0\left(\phi_{kl}|\Phi\right) \\ &= \prod_{kl} \mathrm{H}\left(\mathbf{x}_{kl}|\Phi\right) \end{aligned}$$

Notice how the data likelihood only depends on the entries in each bicluster and the hyperparameters. We will give concrete examples of F, G_0, and H in Section 5.

4 Inference

In this section we describe inference procedures for indicators variables at a fine scale using Gibbs updates and at a coarse scale using split-merge updates. We also briefly show how to sample for PY parameters from their posteriors.

4.1 Gibbs sampling indicators

The procedure for infering row (or column) indicators differs only slightly from typical DP mixture inference (see [9] for examples). When performing Gibbs updates on $\mathbf{u}$ and $\mathbf{v}$, we cycle through the row and column indices, using the exchangeable properties of the PY process to treat each index as the last sample in an exchangeable distribution.

When G_0 is non-conjugate to F, we set $u_i = k$ with probability proportional to the following unnormalized density:

$$\mathrm{PY}\left(u_i = k|\alpha, d_u, \mathbf{u}^{-i}\right) \prod_l \mathrm{F}\left(x_{ij}|\phi_{kl}\right)$$

where, if k is a new row cluster, ϕ_{kl} is sampled from its base distribution (see algorithm 8 in [9]), $\mathbf{x}_{il}$ is the vector of observations from row i that have $v_j = l$, and $\mathbf{x}_{kl}^{-i}$ is the vector observations in bicluster kl, excluding $\mathbf{x}_{il}$. Once the indicators have been sampled for both rows and columns, we then resample bicluster parameters from their posterior:

$$\phi_{kl}|\mathbf{X},\Phi \sim C \cdot G_0\left(\phi_{kl}|\Phi\right) \cdot \mathrm{F}\left(x_{ij}|\phi_{kl}\right)^{\delta_k(u_i)\delta_l(v_j)}$$

When G_0 is conjugate to F, we can *collapse* the Gibbs sampler and set $u_i = k$ with probability proportional to the following unnormalized density:

$$\mathrm{PY}\left(u_i = k|\alpha, d_u, \mathbf{u}^{-i}\right) \prod_l \mathrm{H}\left(x_{ij}|\mathbf{x}_{kl}^{-i}, \Phi\right)$$

Similar, symmetric updates are performed for $\mathbf{v}$. We can understand why the collapsed Gibbs sampler would be much more efficient than the non-conjugate sampler by thinking about how we assess the likelihood of data under new cluster disitributions. For a row cluster to be added, it must sample L new parameters from the prior and have these new parameters model the density of the row better than existing components. Intuitively, we would expect to have a low probability of adding new components with this procedure. Instead, the collapsed sampler integrates over the base distribution's sampling variability.

4.2 Split-merge for cluster indicators

Gibbs updates to single indicator variables can only make small steps in state-space and it is possible that the a Markov chain will remain stuck in poor local minima. For mixtures, splitting and merging clusters can provide the necessary jumps to escape local minima and explore the full state-space. We have applied the conjugate split-merge algorithm of Jain and Neal [6] and describe its essentials below.[3]

The basic idea of split-merge is the following. We are interested in two types of proposals: one proposes *merging* two rows of biclusters into a single row of biclusters; the other proposes *splitting* a row of biclusters into

[3] See [7] for a non-conjugate version of the split-merge algorithm.

two rows of biclusters. Split-merge proposals require three densities: $P(\mathbf{u})$, the joint probability of a partition under the PY prior; $P(\mathbf{X}|\mathbf{u})$, the likelihood of the data given a partition; and $\mathrm{q}\left(\mathbf{u}^{\star}|\mathbf{u}\right)$, the transition probability from the current state $\mathbf{u}$ to the new state $\mathbf{u}^{\star}$. How we compute q is the most important aspect of the split-merge algorithm and we describe it next.

The split-merge algorithm proceeds as follows. Sample uniformly two row indices, f and g. If $f = g$ then propose splitting a row of biclusters currently associated with row cluster u_f, into two rows of biclusters with cluster labels $f^{\text{sp}} = f$ and $g^{\text{sp}} = K + 1$, where K is the current number of occupied clusters. Initially, row indicators $u_i = f$ are randomly assigned f^{sp} and g^{sp}. We then perform several iterations of *restricted* Gibbs to reach the launch state.[4] The transition probability $\mathrm{q}\left(\mathbf{u}^{\text{sp}}|\mathbf{u}\right)$ is the product of probability of reaching the final indicator configuration from the launch state, using a final restricted Gibbs scan.

If $f \neq g$, then we propose merging u_f and u_g into a cluster labeled $f^{\text{mg}} = f$ and removing cluster g. The transition probability $\mathrm{q}\left(\mathbf{u}^{\text{mg}}|\mathbf{u}\right)$ is 1. The reverse transition probability is the product of a *simulated* restricted Gibbs scan from a launch split state to the current split state. The launch split state is found by randomly splitting elements of u_f and u_g, then performing several restricted Gibbs scans.

Once $\mathbf{u}^{\text{sp}}$ or $\mathbf{u}^{\text{mg}}$ is ready to be proposed, we can compute their acceptance probabilties, which for MH is:

$$a\left(\mathbf{u}^{\star}|\mathbf{u}\right) = \min\left[1, \frac{\mathrm{q}\left(\mathbf{u}|\mathbf{u}^{\star}\right)}{\mathrm{q}\left(\mathbf{u}^{\star}|\mathbf{u}\right)} \frac{P\left(\mathbf{u}^{\star}\right)}{P\left(\mathbf{u}\right)} \frac{P\left(\mathbf{x}|\mathbf{u}^{\star}\right)}{P\left(\mathbf{x}|\mathbf{u}\right)}\right]$$

Both the ratios of priors and likelihoods simplify to:

$$\begin{aligned}
\frac{P\left(\mathbf{u}^{\text{sp}}\right)}{P\left(\mathbf{u}\right)} &= \left(\alpha + K^{+}_{\text{sp}} d_u\right) \frac{\Gamma\left(n^{\text{sp}}_{f}\right)\Gamma\left(n^{\text{sp}}_{g}\right)}{\Gamma\left(n_f\right)} \\
\frac{P\left(\mathbf{u}^{\text{mg}}\right)}{P\left(\mathbf{u}\right)} &= \frac{1}{\left(\alpha + K^{+} d_u\right)} \frac{\Gamma\left(n^{\text{mg}}_{f}\right)}{\Gamma\left(n_f\right)\Gamma\left(n_g\right)} \\
\frac{P\left(\mathbf{x}|\mathbf{u}^{\text{sp}}\right)}{P\left(\mathbf{x}|\mathbf{u}\right)} &= \prod_{l} \frac{\mathrm{H}\left(\mathbf{x}_{u^{\text{sp}}_{f} l}|\Phi\right)\mathrm{H}\left(\mathbf{x}_{u^{\text{sp}}_{g} l}|\Phi\right)}{\mathrm{H}\left(\mathbf{x}_{u_f l}|\Phi\right)} \\
\frac{P\left(\mathbf{x}|\mathbf{u}^{\text{mg}}\right)}{P\left(\mathbf{x}|\mathbf{u}\right)} &= \prod_{l} \frac{\mathrm{H}\left(\mathbf{x}_{u^{\text{mg}}_{f} l}|\Phi\right)}{\mathrm{H}\left(\mathbf{x}_{u_f l}|\Phi\right)\mathrm{H}\left(\mathbf{x}_{u_g l}|\Phi\right)}
\end{aligned}$$

In our experiments, we alternate a full Gibbs scan for $\mathbf{u}$ and $\mathbf{v}$, followed by five split-merge proposals. We propose a split-merge move on $\mathbf{u}$ with probability $N/(N + D$, and for v, with probability $D/(N + D)$. We flip a coin to decide whether we propose split or merge moves (otherwise plit proposals are rare events).

4.3 Inferring Pitman-Yor hyperparameters

There is no reason why both PY parameters α and d should not be sampled during simulation. Elaborate sampling schemes do exist for the concentration parameter for DP mixtures [12], but this is unnecessary; we use random walk Metropolis moves. As mentioned earlier, sensible priors for α and d are Gamma and Beta distributions. Our hyperparameters for the discount parameter are such that it favours smaller d.

5 Gaussian model

We now describe the biclustering model we use in our experiments. It is possible to use non-conjugate Gibbs with the following, but in our experiments we integrate out the bicluster parameters.

5.1 A robust bicluster model

We will assume for the moment that there is only a single bicluster to keep the presentation clear. We will follow with the full bicluster model. We assume the following distribution hierarchy:

[4] A restricted Gibbs scan is the same as the collapsed Gibbs scan described in Section 4.1, but only involving the rows in the proposal and those indicators can only choose between f^{sp} and g^{sp}.

$$
\begin{aligned}
x_{ij}|w,a,s &\sim \text{Normal}\left(w,(as)^{-1}\right)\\
w|m,b,s &\sim \text{Normal}\left(m,(bs)^{-1}\right)\\
s|\nu,c &\sim \text{Gamma}\left(\frac{\nu}{2},\frac{\nu}{2}\frac{1}{c}\right)
\end{aligned}
$$

where w is the bicluster centre; s is a precision parameter affecting the noise in the data and centre distributions; a and b are positive scalars which affect the precision parameter; m is the global mean of the centres; the precisions have Gamma priors with shape $\frac{\nu}{2}$ and inverse scale $\frac{\nu}{2}\frac{1}{c}$; ν is a degrees-of-freedom parameter, controlling the variance of precisions; the expected value of the precisions is c.

This distribution hierarchy provides a *robust* model of the bicluster data. When we integrate out the centres and precisions of the bicluster, the result is a Student-t distribution with ν degrees of freedom; the Student-t is considered less sensitive to outliers than Gaussians due to its fatter tail, and thus more robust.

The marginal likelihood of N univariate entries in a bicluster, after integrating out centres and precisions, is a multivariate Student-t of dimension N with mean μ, precision matrix Q, and degrees of freedom v, where

$$
\mu = m\mathbf{1} \qquad Q = ca\left(\mathbf{I} - \frac{a}{a\cdot ND+b}\mathbf{1}\mathbf{1}^{\top}\right)
$$

We are also interested in the predictive distribution of a vector $\mathbf{y}$ of size N_y, conditioned $\mathbf{X}$, a multivariate Student-t of dimension N_y with ν degrees of freedom and

$$
\mu = \frac{a\cdot\sum_{i=1,j=1}^{N,D} x_{ij} + b\cdot m}{a\cdot ND+b}\mathbf{1} \quad Q = ca\left(\mathbf{I} - \frac{a}{a\cdot(ND+N_y)+b}\mathbf{1}\mathbf{1}^{\top}\right)
$$

5.2 Complete Gaussian biclustering model

In the previous section we gave the marginal and predictive distributions of a Gaussian model with a single bicluster. Here we expand the model to include the case of $K\cdot L$ biclusters. Using the same notation from Section **??**, $\phi_{kl}=\{w_{kl},s_{kl}\}$ and $\Phi=\{m,a,b,\nu,c\}$, so that

$$
\begin{aligned}
\text{F}\left(x_{ij}|\phi_{u_i v_j},\Phi\right) &= \text{Normal}\left(w_{kl},(as_{kl})^{-1}\right)\\
G_0\left(\phi_{u_i v_j}|\Phi\right) &= \text{Normal}\left(m,(bs_{kl})^{-1}\right)\text{Gamma}\left(\frac{\nu}{2},\frac{\nu}{2}\frac{1}{c}\right)\\
\text{H}\left(\mathbf{x}_{kl}|\Phi\right) &= \text{Student-t}\left(\mu_{kl},Q_{kl},\nu\right)\\
\text{H}\left(\mathbf{x}_{il}|\mathbf{x}_{kl}^{-i},\Phi\right) &= \text{Student-t}\left(\mu_{il},Q_{il},\nu\right)
\end{aligned}
$$

where

$$
\mu_{kl} = m\mathbf{1} \qquad \mu_{il} = \frac{a\cdot\sum\mathbf{x}_{kl}^{-i} + b\cdot m}{a\cdot N_{kl,-i}+b}\mathbf{1}
$$

and

$$
\begin{aligned}
Q_{kl} &= ca\left(\mathbf{I} - \frac{a}{a\cdot N_{kl}+b}\mathbf{1}\mathbf{1}^{\top}\right)\\
Q_{il} &= ca\left(\mathbf{I} - \frac{a}{a\cdot(N_{kl,-i}+N_{il})+b}\mathbf{1}\mathbf{1}^{\top}\right)
\end{aligned}
$$

6 Experiments

We perform two types of experiments that are not only important to practitioners, but also demonstrate the performance capability of our Bayesian biclustering algorithm. The first is missing value imputation, where we assess the quality of imputed values using root mean-square error (RMSE). The second is cluster analysis, which is much more difficult to assess. For this we merely show some biclusterings and word neighbourhoods using MCMC cluster samples.

6.1 Data sets

We study our method on three different types of data, all subjects of bicluster research: DNA microarray data, document data, and recommendation data.

RNA probes: We first removed all columns with all missing values, then all rows with any missing values, resulting in a matrix of size 828 by 217.[5]

Documents: We have taken a small subset of the original *newsgroup* dataset, using the log of the counts plus one ($x_{ij} = \log(n_{ij} + 1)$).

Recommendation: We have taken a subset of the *Eachmovie* dataset, using the same transformation of ranks as we did for word counts. For this dataset, we consider zeros missing values (in constrast to the other datasets). Thus, a dataset of size 500 by 500 only has 19% of entries observed.

6.2 Missing value imputation

Different application areas have different reasons why they impute missing values. Biologists working with DNA microarray data are often faced with missing values for several reasons, mostly due to artifacts in processing microarray images or by actual missing experimental data. Biologists may be interested in the actual imputed values, but often they require complete data matrices to perform cluster analysis.[6] Predicted values are extremely important for recommendation systems, where the rankings of, say, movies by users, and the relationships between users, are used to recommend movies.

Even though out method integrates over missing values, assessing our methods imputation capability is one way of analysing the quality of the biclusters it discovers. We compare our algorithm with several imputation methods designed for DNA microarray data. Baseline methods fill-in missing values with zeros or the row average of observations (ROW). More advanced methods are based on Singular Value Decomposition (SVD) and k-nearest-neighbours (KNN) [5, 11]. These both iterate imputation until imputed values have converged. The most sophisticated methods we compare with are based on least-squares analysis (LS) [1] and probabilistic PCA (BPCA) [10].

Our main modeling assumption—block constant biclusters—may only be valid in few data sets one may encounter. Many microarray data sets, for example, are obviously not block-constant. Experiments may record experiments which are time-dependent, for instance. In such cases, biclustering with constant values will provide poor descriptions of the data. We can sometimes improve the performance of imputation algorithms by initializing the missing values with the average predicted values from our method (see RNA results). This is another way of demonstrating the quality of the biclustering.

Our imputation experiments are straightforward. For each dataset, there is an original matrix Y, which may or may not contain missing values (this is the case for EACH only). Usind a missing completely at random (MCAR), we perform experiments with $5 - 90\%$ of the observed entries set to missing. For each missing rate, we create 5 versions of the data with different missing entries. As mentioned before, we assess quality by RMSE.

We can see from Figure 4 that our algorithm, despite not being designed for imputation, performed well on data with high missing rates. On the RNA data, we can improve the results of BPCA by initializing it with our expected predictions. Our results indicate that our algorithm is quite robust to high rates of missing data.

6.3 Cluster analysis

In Figures 5 and 6 we show biclusterings from all data sets. Notice how when zeros are treated as missing values (top Figure 6), that the biclustering is more interesting. This is due to the extra noise in this data. Less biclusters are formed when zeros are treated as missing, but these biclusters are still valid for this level of sparsity. We also show word neighbours from NEWSGROUPS in Table 1.

7 Conclusion

We have presented an fully Bayesian biclustering algorithm that is very robust to missing values, precluding any need for imputation before further analysis. This has important consequences for practitioners. If one is actually interested in studying imputed values, our algorithm is capable of imputing values. For data matrices that are very sparse, other imputation methods fail, and therefore biclustering algorithms that rely on complete data will fail as well. Our method shows much more gradual degradation in performance as the missing rate increases. In the future we will work on a version for multinomial data and will extend the biclustering to hierarchies.

[5] Thanks to Tim Hughes and Quaid Morris for providing this unpublished data.

[6] Of course, for these situations, our method obviates imputation before clustering.

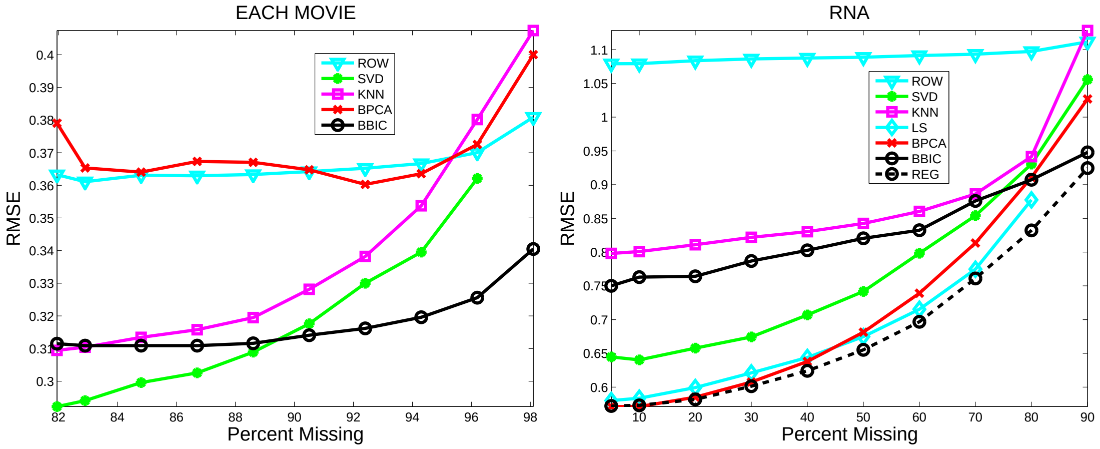

Figure 4: **Top**: Inputation results for the EACH MOVIE dataset. Even at very high sparsity (98% missing rate), our method is still able to perform reasonably well. **Bottom**: Inputation results for the RNA dataset. For this data, even though our imputation performed poorly, initializing BPCA (see REG) with our imputed values improved BPCA significantly.

References

[1] Trond Hellem Bo, Bjarte Dysvik, and Inge Jonassen. LSimpute: accurate estimation of missing values in microarray data with least squares methods. *Nucleic Acids Research*, 32(3), 2004.

[2] Yizong Cheng and George M. Church. Biclustering of expression data. In *ISMB-2000*, volume 8, pages 93–103, 2000.

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

[4] I.E. Givoni, V. Cheung, and B.J. Frey. Matrix tile analysis. In *UAI*, volume 22, 2006.

[5] Trevor Hastie, Robert Tibshirani, Gavin Sherlock, Michael Eisen, Patrick Brown, , and David Botstein. Imputing missing data for gene expression arrays. Technical report, Division of Biostatistics, Stanford University, 1999.

[6] Sonia Jain and Radford M. Neal. A split-merge Markov chain monte carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13:158–182, 2004.

[7] Sonia Jain and Radford M. Neal. Splitting and merging for a nonconjugate Dirichlet process mixture model. Technical Report 0507, Department of Statistics, University of Toronto, 2005.

[8] Laura Lazzeroni and Art Owen. Plaid models for gene expression data. *Statistica Sinica*, 12:61–68, 2002.

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

[10] Shigeyuki Oba, Masa aki Sato, Ichiro Takemasa, Morito Monden, Ken ichi Matsubara, and Shin Ishii. A bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16):2088–2096, 2003.

[11] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[12] Mike West. Hyperparameter estimation in dirichlet process mixture models. Technical report, Duke University, 1992.

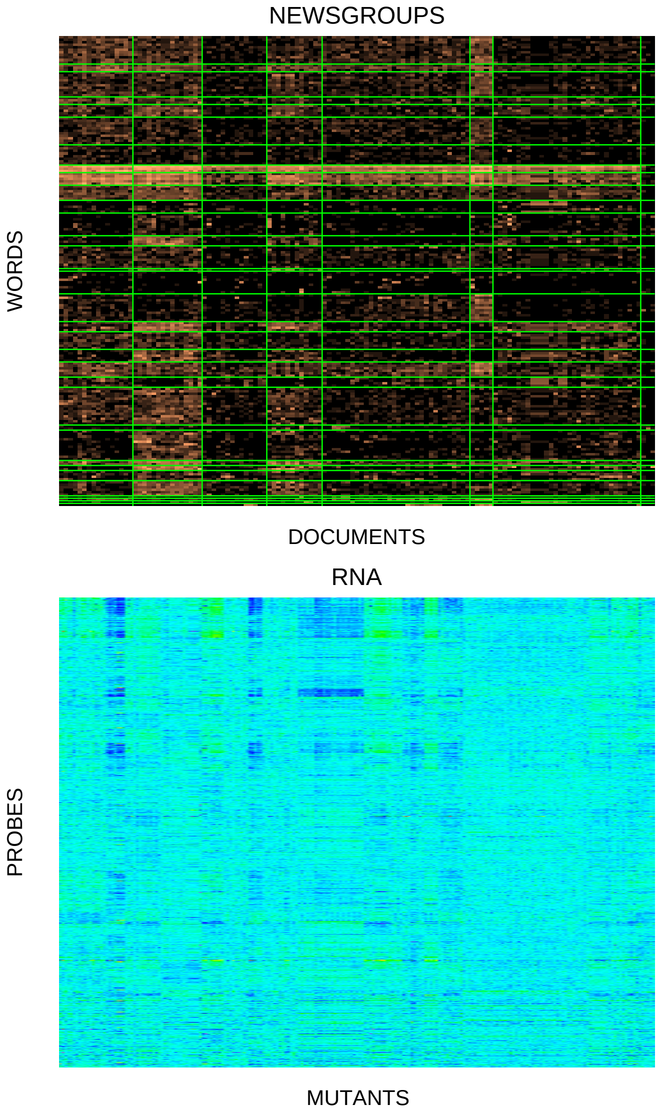

Figure 5: **Top**: NEWSGROUPS biclustering sample. **Bottom**: RNA biclustering sample. Partition lines are left out for clarity.

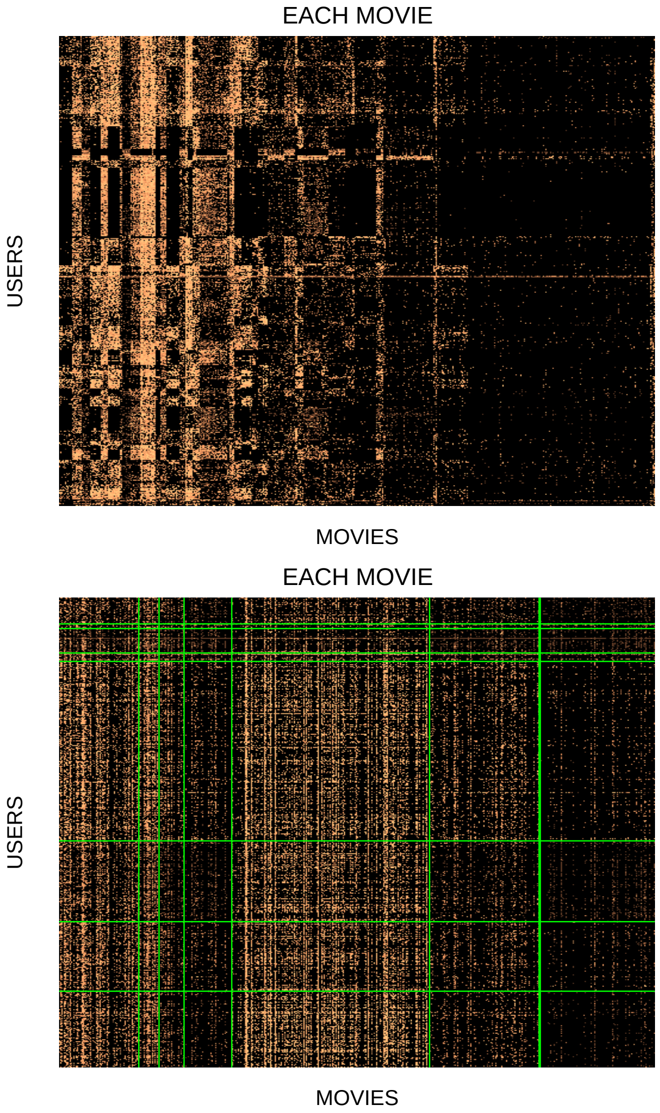

Figure 6: **Top**: EACH MOVIE. A biclustering result when zeros are treated as observed. We have left out partition lines for clarity. **Bottom**: EACH MOVIE A biclustering result when zeros are treated as missing.

doesn	**write**	**world**	**inform**	**discuss**
get	origin	second	system	effect
lot	articl	found	includ	interest
true	free	reason	program	reason
word	check	book	file	real
hand	subject	exist	number	high
show	exampl	respons	gener	refer
great	chang	place		order
idea	exist	give		place
never	open	show		give
suggest	etc	power		power
quit	control	end		respons
hard	new	fact		fact
claim	net	put		nation
turn	manag	interest		person
talk	bit	articl		control
man	found	discuss		articl
wrong	interest	build		sinc
respons	suggest	hand		second
book	type	refer		answer

Table 1: Nearest neighbours of a random set of words in a mini NEWSGROUP dataset. The neighbours were based on the probability of a word being in the same cluster as another word, based on biclustering samples.