A Nonparametric Bayesian Approach to Modeling Overlapping Clusters

Katherine A. Heller
Gatsby Computational Neuroscience Unit
University College London

Zoubin Ghahramani[*]
Engineering Department
University of Cambridge

Abstract

Although clustering data into mutually exclusive partitions has been an extremely successful approach to unsupervised learning, there are many situations in which a richer model is needed to fully represent the data. This is the case in problems where data points actually simultaneously belong to multiple, overlapping clusters. For example a particular gene may have several functions, therefore belonging to several distinct clusters of genes, and a biologist may want to discover these through unsupervised modeling of gene expression data. We present a new nonparametric Bayesian method, the Infinite Overlapping Mixture Model (IOMM), for modeling overlapping clusters. The IOMM uses exponential family distributions to model each cluster and forms an overlapping mixture by taking products of such distributions, much like products of experts (Hinton, 2002). The IOMM allows an unbounded number of clusters, and assignments of points to (multiple) clusters is modeled using an Indian Buffet Process (IBP), (Griffiths and Ghahramani, 2006). The IOMM has the desirable properties of being able to focus in on overlapping regions while maintaining the ability to model a potentially infinite number of clusters which may overlap. We derive MCMC inference algorithms for the IOMM and show that these can be used to cluster movies into multiple genres.

1 Introduction

The problem of clustering data has led to many pivotal methods and models (Duda et al., 2001; Meila and Shi, 2000) in pattern recognition and machine learning which are widely used across many fields. Unfortunately, while clustering methods are wonderful tools for many applications, they are actually quite limited. Clustering models traditionally assume that each data point belongs to one and only one cluster; that is, there are K exhaustive and mutually exclusive clusters explaining the data. In many situations the data being modeled can have a much richer and more complex hidden representation than this single, discrete hidden variable (the cluster or partition assignment) which clustering strives to discover. For example, there may be overlapping regions where data points actually belong to multiple clusters (e.g. the movie "Scream" could belong to both the "horror" movie cluster and the "comedy" cluster of movies). Also, in collaborative filtering one might be interested in predicting which movies someone will like based on previous movies they have liked, and the patterns of movie preferences of others. A common approach is to cluster people; clusters could characterize gender, age, ethnicity, or simply movie taste (e.g. people who like horror movies). However, any particular person can clearly belong to multiple such clusters at the same time, e.g. a female in her 20s who likes horror movies.

In this paper, we develop a new model for overlapping clusters based on a principled statistical framework. Consider the traditional mixture model (Bishop, 2006) for clustering, which can be written

$$p(\mathbf{x}_i|\Theta) = \sum_{j=1}^{K} \pi_j p_j(\mathbf{x}_i|\boldsymbol{\theta}_j)$$

where π_j represents the mixing weight (or mass) of cluster j, $p_j(\mathbf{x}_i|\boldsymbol{\theta}_j)$ is the density for cluster j with parameters $\boldsymbol{\theta}_j$, and $\mathbf{x}_i$ represents data point i. This

[*]ZG is also an Associate Research Professor in the Machine Learning Department at Carnegie Mellon University

mixture model can be rewritten

$$p(\mathbf{x}_i|\Theta) = \sum_{\mathbf{z_i}} p(\mathbf{z_i}) \prod_{j=1}^{K} p_j(\mathbf{x}_i|\boldsymbol{\theta}_j)^{z_{ij}} \qquad (1)$$

where $\mathbf{z_i} = [z_{i1}, \ldots, z_{iK}]$ is a binary vector of length K, $z_{ij} \in \{0,1\} \forall ij$, $\sum_j z_{ij} = 1$, and $P(z_{i1} = 0, \ldots, z_{i,j-1} = 0, z_{ij} = 1, z_{i,j+1} = 0, \ldots, z_{iK} = 0) = \pi_j$. The setting $z_{ij} = 1$ means data point i belongs to cluster j.

To create a model for *overlapping* clusters, two modifications can be made to this representation. First of all, removing the restriction $\sum_j z_{ij} = 1$ allows binary vectors with multiple ones in each row. In other words, instead of K possible binary $\mathbf{z}$ vectors allowed in the mixture model, this allows 2^K possible assignments to overlapping clusters. Removing this restriction will also introduce a normalization constant for the product which for exponential family densities $P_j(x)$ will be easy to compute. Secondly, the number of such overlapping clusters K can be taken to infinity by making use of the Beta-Binomial model underlying the Indian Buffet Process (IBP), (Griffiths and Ghahramani, 2006). This infinite limit means that the model is not restricted a priori to having a fixed number of clusters; and it allows the data to determine how many clusters are required. In the case where the $P_j(\cdot)$ are Gaussian densities, this model will define overlapping clusters in terms of the region where the mass of all Gaussians j, such that $z_{ij} = 1$, overlaps; this region itself will define a Gaussian since the product of Gaussians is Gaussian. Other exponential family models (e.g. multinomials for text data) will work analogously. In sections 2 and 3 we describe this Infinite Overlapping Mixture Model in detail, and in section 4 we outline how to perform inference in the model.

This model for overlapping clusters can be seen as a modern nonparametric generalization of the multiple cause factorial models (Saund, 1994; Ghahramani, 1995). Moreover, it can also be seen as an infinite nonparametric generalization of the influential products-of-experts model (Hinton, 2002). These relationships will be discussed further in section 5. Lastly, we give experimental results for our model in section 6.

2 Overlapping Mixture Models

We are interested in clustering data such that each data point is allowed to belong to multiple clusters, instead of being constrained to a single cluster. In order to do this we need a sensible way of modeling individual data points that belong to many clusters, and which derives from the broad models for each individual cluster. We modify a traditional finite mixture model (1) to achieve this. First we remove the restriction that the binary assignment vector, $\mathbf{z}$, for each data point must sum to 1, and secondly, as we will discuss in the next section, we use a prior that allows a potentially infinite number of clusters, where the actual required number of clusters is inferred automatically from the data. Removing the restiction that $\mathbf{z}$ sums to one in (1), results in a model in which, if a data point belongs simultaneously to several clusters, the distribution of that point is given by the product of component distributions:

$$p(\mathbf{x}_i|\mathbf{z}_i, \Theta) = \frac{1}{c} \prod_k p_k(\mathbf{x}_i|\boldsymbol{\theta}_k)^{z_{ik}} \qquad (2)$$

Here $\mathbf{z}_i = (z_{i1} \ldots z_{iK})$ is a binary vector of cluster assignments for data point i, $\boldsymbol{\theta}_k$ are the parameters of cluster k, and c is the normalizing constant which is needed to ensure that the density integrates to one. Multiplying distributions is a very natural and general way of encoding the idea of overlapping clusters—each cluster provides a soft constraint on the probable region for observing a data point, and overlapping clusters correspond to a conjunction of these constaints.

If the models we are using, $p(\mathbf{x}_i|\boldsymbol{\theta}_k)$, are in the exponential family then:

$$p(\mathbf{x}_i|\boldsymbol{\theta}_k) = g(\mathbf{x}_i) f(\boldsymbol{\theta}_k) e^{s(\mathbf{x}_i)^\top \phi(\boldsymbol{\theta}_k)} \qquad (3)$$

where $s(\mathbf{x}_i)$ are the sufficient statistics, $\phi(\boldsymbol{\theta}_k)$ are the natural parameters, and f and g are non-negative functions. Substituting into equation (2) we get:

$$p(\mathbf{x}_i|\mathbf{z}_i, \Theta) = \frac{g(\mathbf{x}_i)^{\sum_k z_{ik}}}{c} \left[\prod_k f(\boldsymbol{\theta}_k)\right] e^{s(\mathbf{x}_i)^\top (\sum_k z_{ik} \phi(\boldsymbol{\theta}_k))} \qquad (4)$$

From this we see that, conditioned on $\mathbf{z}_i$, the product of exponential family distributions results in a distribution in the same family (3), but with new natural parameters $\tilde{\phi} = \sum_k z_{ik} \phi(\boldsymbol{\theta}_k)$. It follows that normalization constants like c are not problematic when the component densities are in the exponential family.

In the case of Gaussian clusters:

$$p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\mu}, \Sigma) = \frac{1}{c} \exp\left\{ -\frac{1}{2} [\mathbf{x}^\top (\sum_k z_{ik} \Sigma_k^{-1}) \mathbf{x} \right.$$
$$\left. -2\mathbf{x}^\top (\sum_k z_{ik} \Sigma_k^{-1} \boldsymbol{\mu}_k) + \sum_k z_{ik} \boldsymbol{\mu}_k^\top \Sigma_k^{-1} \boldsymbol{\mu}_k] \right\} \qquad (5)$$

Letting $S^{-1} = \sum_k z_{ik} \Sigma_k^{-1}$ and $\mathbf{m} = \sum_k z_{ik} \Sigma_k^{-1} \boldsymbol{\mu}_k$ from within equation (5), we can see that the new parameters for the Gaussian product model are $\tilde{\Sigma} = S$ and $\tilde{\boldsymbol{\mu}} = S\mathbf{m}$.

In the case of binary data and multivariate Bernoulli

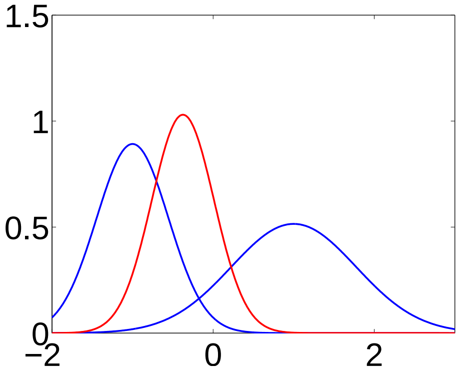

Figure 1: Product of two Gaussians. Here the product of the two blue Gaussians ($\mu_1 = -1$, $\mu_2 = 1$ and $\sigma_1^2 = 0.2$, $\sigma_2^2 = 0.6$) is the red Gaussian.

clusters:

$$p(\mathbf{x}_i|\mathbf{z}_i, \Theta) = \frac{1}{c}\exp\{\sum_{k,d} z_{ik}x_{id}\log(\frac{\theta_{kd}}{1-\theta_{kd}})\} \tag{6}$$

where d indexes the dimensions of $\mathbf{x}_i$. Using equation (6) we can derive that the new parameters for the Bernoulli product model are:

$$\tilde{\Theta}_d = \frac{\prod_k \theta_{kd}^{z_{ik}}}{\prod_k (1-\theta_{kd})^{z_{ik}} + \prod_k \theta_{kd}^{z_{ik}}}. \tag{7}$$

These product models have the desirable property that multiple cluster assignments will help focus the model on a particular overlapping region. See Figure 1 for a simple illustration. The two blue Gaussians each model independent Gaussian clusters ($\mathbf{z}_1 = [1\ 0]$ and $\mathbf{z}_2 = [0\ 1]$); the red Gaussian models the overlap of the two blue Gaussians clusters and defines the overlapping cluster $\mathbf{z}_3 = [1\ 1]$.

3 Infinite Overlapping Mixture Models via the IBP

The model in the previous section defines a generative distribution for overlapping clusters by forming conjunctions of component models. The key component in this model is the binary vector $\mathbf{z}_i$ which indicates which clusters data point i belongs to. We have defined in the previous section how the component models are combined, given the binary assignment vector $\mathbf{z}_i$; we now turn to the distribution over these binary assignment vectors.

A very simple model assigns each element z_{ik} an independent Bernoulli distribution

$$z_{ik}|\pi_k \quad \sim \quad \text{Bernoulli}(\pi_k) \tag{8}$$

where π_k is the mixing proportion, or probability of belonging to cluster k. Note that the π_k need not sum to 1 over k, since belonging to one cluster does not exclude belonging to others. We give each π_k a Beta distribution

$$\pi_k|\alpha \quad \sim \quad \text{Beta}(\frac{\alpha}{K}, 1) \tag{9}$$

which is conjugate to the Bernoulli, where α controls the expected number of clusters a data point will belong to.

A classical problem in clustering, which also occurs in our overlapping clustering model, is how to choose the number of clusters K. While it is possible to perform model comparison for varying K, this is both computationally costly and statistically hard to justify (Neal, 2000). A more elegant solution is to define a nonparametric model which allows an unbounded number of clusters, K.

In order to derive the nonparametric model, we have defined the prior over π_k in (9) to scale so that as K grows larger, the prior probability of each data point belonging to cluster k decreases. Using this scaling it is possible to take the limit $K \to \infty$, integrate out all the mixing proportions π, and still obtain a well-defined distribution over the binary assignment vectors $\mathbf{z}$. This distribution over the assignment vectors results in a process known as the Indian Buffet Process (IBP), (Griffiths and Ghahramani, 2006).

The IBP defines a distribution which can be used to represent a potentially infinite number of hidden features, or in this case cluster assignments, associated with data points. More specifically, it defines a distribution over infinite binary matrices, Z, which can be derived by starting with a distribution over finite $N \times K$ matrices given by (8) and (9), where N is the number of data items, K is the number of features, and the ith row of Z is $\mathbf{z}_i$, and taking the limit as K goes to infinity. Exchangeability of the rows is preserved, and the columns are independent.

The IBP is a simple generative process which results from this distribution, with an analogy to customers eating from Indian Buffets. N customers line up on one side of an Indian Buffet with infinitely many dishes. The first customer serves himself from Poisson(α) dishes (at which point his plate is full). The next customers serve themselves dishes in proportion to their popularity, such that customer i serves herself dish k with probability $\frac{m_k}{i}$, where m_k is the number of previous customer which have served themselves that dish. After passing all previously sampled dishes, customer i then proceeds to try Poisson($\frac{\alpha}{i}$) new dishes. In terms of binary matrices, each of the N customers is a row in the matrix, each dish is a column, and each binary value in the matrix, z_{ik}, indicates whether customer i helped themselves to dish k. A sample of such

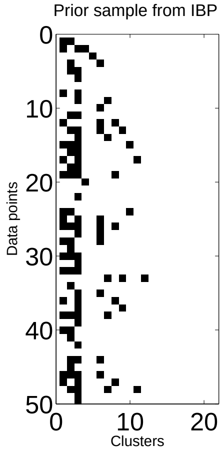

Figure 2: The first 50 rows of the IBP sample (Z matrix) which was used to assign data points to clusters in Figure 5b.

a matrix is shown in Figure 2.

Markov Chain Monte Carlo algorithms have been used to do inference in this model (Griffiths and Ghahramani, 2005; Görür et al., 2006). These algorithms need to compute the full conditional distribution of the assignment variables:

$$P(z_{ik}=1|Z_{-(ik)},X) \propto P(X|Z)P(z_{ik}=1|Z_{-(ik)}) \tag{10}$$

where X is the complete data matrix and Z is the full binary feature matrix, and $Z_{-(ik)}$ is the binary matrix excluding element z_{ik}. In order to compute the last term in equation (10), we can generalize from the finite binary matrix case. Starting from (8) and (9) and integrating out π_k gives:

$$\begin{aligned} P(z_{ik}=1|\mathbf{z}_{-i,k}) &= \int_0^1 P(z_{ik}|\pi_k)P(\pi_k|\mathbf{z}_{-i,k})d\pi_k \\ &= \frac{m_{-i,k}+\frac{\alpha}{K}}{N+\frac{\alpha}{K}} \end{aligned} \tag{11}$$

where $m_{-i,k} = \sum_{j\neq i} z_{jk}$ and $\mathbf{z}_{-i,k}$ is $\mathbf{z}_i$ excluding. Taking the limit as $K \to \infty$ results in:

$$P(z_{ik}=1|\mathbf{z}_{-i,k}) = \frac{m_{-i,k}}{N} \tag{12}$$

for any k in which $m_{-i,k} > 0$. The number of new features associated with i should be drawn from a Poisson($\frac{\alpha}{N}$) distribution. The IBP is described in full detail in (Griffiths and Ghahramani, 2005).

Incorporating this IBP prior over the assignment vectors into the OMM defined in section 2 results in an Infinite Overlapping Mixture Model (IOMM), with all the components required to do inference and learning.

4 IOMM Learning

We use Markov Chain Monte Carlo (MCMC) to do inference in our Infinite Overlapping Mixture Model (IOMM). The MCMC algorithm that we implemented is based on Figure (3). Since the product model is non-conjugate we use Metropolis-Hastings (MH) to resample the model parameters, Θ.

```
Initialize $\Theta$
for $j = 1$ to NumIters do
  for $i = 1$ to $N$ do
    for $k = 1$ to $k_+$ do
      $z_{ik} \sim z_{ik}|z_{-i,k}, \mathbf{x}_i, \Theta$
    end for
    Propose adding new clusters
    Accept or reject proposal
  end for
  Resample $\Theta|Z, X$ using MH proposal
end for
```

Figure 3: MCMC for IOMM, where $k+$ is the number of clusters which data points, excluding i, belong to.

At each iteration we resample the binary matrix, Z, using Gibbs sampling for existing clusters k (i.e. those clusters which have data points other than i as members), where:

$$p(z_{ik}=1|\mathbf{z}_{-i,k},\mathbf{x}_i,\Theta) \propto \frac{m_{-i,k}}{N} p(\mathbf{x}_i|\Theta, z_{ik}=1, \mathbf{z}_{-i,k})$$

and

$$p(z_{ik}=0|\mathbf{z}_{-i,k},\mathbf{x}_i,\Theta) \propto \frac{N-m_{-i,k}}{N} p(\mathbf{x}_i|\Theta, z_{ik}=0, \mathbf{z}_{-i,k})$$

After resampling the existing cluster assignments for a data point i, we then propose adding assignments of i to new clusters using Metropolis-Hastings and following (Meeds et al., 2007). Here the number of new clusters and their parameters are proposed jointly, where the number of new clusters is drawn from a Poisson($\frac{\alpha}{N}$) and the new parameters for those cluster are drawn from the prior.

After resampling the entire Z matrix we resample each θ'_{kd} drawing from the proposal distribution centered around the current value of θ_{kd}. The acceptance ratio for θ'_{kd} is:

$$a = \frac{p(\mathbf{x}_d|Z,\boldsymbol{\theta}'_d)p(\boldsymbol{\theta}'_d)}{p(\mathbf{x}_d|Z,\boldsymbol{\theta}_d)p(\boldsymbol{\theta}_d)} \frac{T(\theta_{kd}|\theta'_{kd},\omega)}{T(\theta'_{kd}|\theta_{kd},\omega)} \tag{13}$$

where $\boldsymbol{\theta}'_d$ is $\boldsymbol{\theta}_d$ substituting θ'_{kd} for θ_{kd}, T is the transition probability between different values of θ_{kd}, and

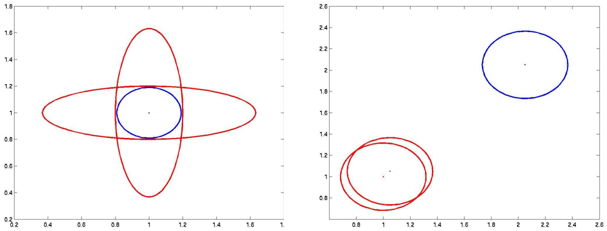

Figure 4: Left: IOMM where the cluster component densities are Gaussian (contours at 1 s.d.). Right: Factorial Model. In each figure, the original Gaussian clusters are shown in red, while the Gaussian cluster modeling membership in both of the original clusters is shown in blue. The IOMM is able to focus in on the area where the original two Gaussians overlap, taking their (unrestricted) covariances into account. The factorial model yields a Gaussian whose mean is the sum of the means of the original two Gaussians, and the (typically axis-aligned) covariance is restricted to be the same for all clusters, since it results from the same additive noise.

ω controls the width of this transition proposal distribution. For example, for binary data we can use multivariate Bernoulli clusters 6, 7. A sensible proposal for θ_{kd} might be $\theta'_{kd} \sim \text{Beta}(\omega\theta_{kd}, \omega(1-\theta_{kd}))$.

5 Related Work

The infinite overlapping mixture model has many interesting relationships to other statistical models. In this section we review some of these relationships, highlighting similarities and differences.

The likelihood function in equation (2) is a product of likelihoods from different component densities, which is highly reminiscent of the *products of experts* (PoE) model (Hinton, 2002). In a PoE, the data model is:

$$p(\mathbf{x}_i|\Theta) = \frac{1}{c}\prod_k p_k(\mathbf{x}_i|\boldsymbol{\theta}_k).$$

Comparing to (2), we see that while in the IOMM, for each data point, $\mathbf{x}_i$ a product of a *subset* of the experts is taken depending on the setting of $\mathbf{z}_i$, in the PoE, each data point is assumed to be generated by the product of *all* experts. This would appear to be a large difference; however we will now show that it is not. Consider the special case of a PoE where each expert is a mixture of a uniform and a Gaussian distribution (a "unigauss" distribution), described in Section 4 of Hinton (2002).[1] For this model, using $\mathbf{1}(x) = 1, \forall x$, to denote the unnormalized uniform distribution (where normalization is subsumed in c above):

$$\begin{aligned} p_k(\mathbf{x}_i|\boldsymbol{\theta}_k) &= (1-\pi_k)\,\mathbf{1}(\mathbf{x}_i) + \pi_k\,\mathcal{N}(\mathbf{x}_i|\mu_k,\Sigma_k) \\ &= \sum_{z_{ik}\in\{0,1\}} p(\mathbf{x}_i|z_{ik},\boldsymbol{\theta}_k)p(z_{ik}|\boldsymbol{\theta}_k) \end{aligned} \quad (14)$$

where $p(z_{ik} = 1|\boldsymbol{\theta}_k) = \pi_k$ and $p(\mathbf{x}_i|z_{ik},\boldsymbol{\theta}_k) = \mathcal{N}(\mathbf{x}_i|\mu_k,\Sigma_k)^{z_{ik}}$. Conditioning on $\mathbf{z}_i$ we now see that

$$p(\mathbf{x}_i|\mathbf{z}_i,\Theta) \propto \prod_k \mathcal{N}(\mathbf{x}_i|\mu_k,\Sigma_k)^{z_{ik}}$$

which is of the same form as in the IOMM (2). More generally, we can therefore view our IOMM as an infinite nonparametric Bayesian Product of Experts, under the assumption that each expert is a mixture of a uniform and an exponential family distribution.

Another line of thought relates the IOMM to multiple cause or factorial models (Saund, 1994; Hinton and Zemel, 1994; Ghahramani, 1995; Sahami et al., 1996). Factorial models are closely related to factor analysis. Each data point $\mathbf{x}_i$ is represented by a latent vector $\mathbf{z}_i = (z_{i1},\ldots,z_{iK})$. In factor analysis, $\mathbf{z}_i$ is assumed to be multivariate Gaussian, and $\mathbf{x}_i$ and $\mathbf{z}_i$ are assumed to be linearly related. A factorial model can be obtained by letting each z_{ik} be discrete, the binary case $z_{ik} \in \{0,1\}$ corresponds to data points having 2^K possible feature vectors. The corresponding distributions over data for each possible feature vector are formed by somehow combining parameters associated with the individual features. In (Hinton and Zemel, 1994; Ghahramani, 1995), the parameters of the individual features are simply added to form the mean of the distribution of $\mathbf{x}_i$ given $\mathbf{z}_i$, with subsequent Gaussian noise added. That is, $E[\mathbf{x}_i] = A\mathbf{z}_i$, where A is some $D \times K$ matrix whose columns are means for the individual features, and the binary vector $\mathbf{z}_i$ picks out which columns to include in the sum for each point. This idea was used to model gene expression data by (Lu et al., 2004); it was also independently re-invented by (Segal et al., 2003; Battle et al., 2005) and also used to discover multiple overlapping processes in gene expression data. Recently, the model of Segal et al. (2003) was extended by Banerjee et al. (2005) from Gaussians to other exponential family distributions.

While all the models we have reviewed in the previous paragraph are clearly useful and share with the IOMM the idea of using a binary latent vector $\mathbf{z}_i$ to model presence or absence of a hidden feature (which could be seen as indicating membership in a particular "cluster"), they do not make reasonable models for *overlapping* clusters. The additive combination rule of the factorial models $E[\mathbf{x}_i] = A\mathbf{z}_i$ does not capture the intuition of overlapping clusters, but rather of multi-

[1] Strictly speaking a "uniform" on the reals is improper, but this can be approximated by a Gaussian with very large variance.

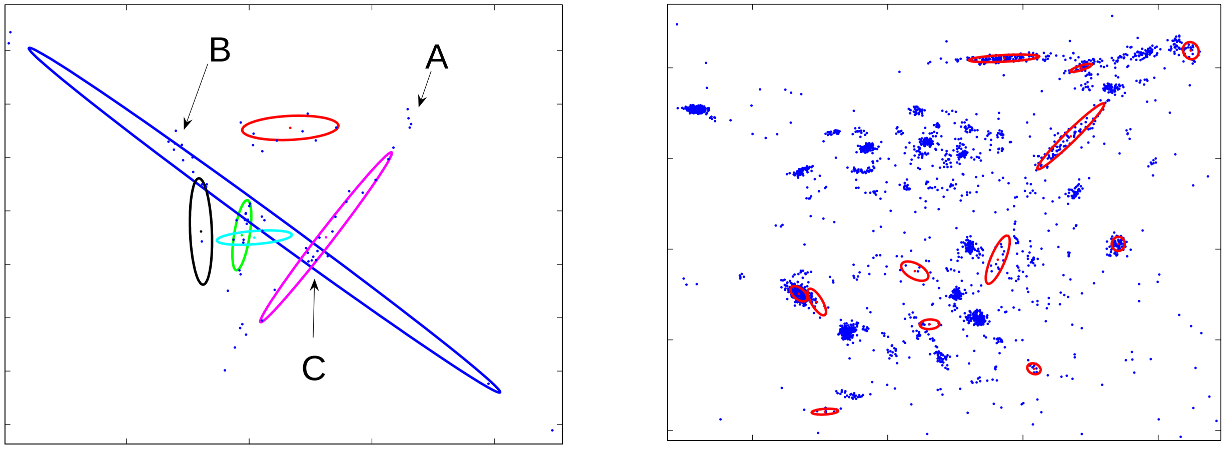

Figure 5: Two draws from the IOMM with Gaussian cluster models. a) left: A draw with 6 independent Gaussian clusters. Label A shows data points which belong to both the red and magenta clusters, label B shows data points which belong to both the red and blue clusters, and label C shows datapoints which belong to both the magenta and blue clusters. b) right: A larger draw from the IOMM with more independent clusters. Part of the IBP sample (Z matrix) used for assignments of data points to clusters is shown in Figure 2

ple processes that add together (Figure 4). For example, if the first and second columns of A are identical ($\mathbf{a}_1 = \mathbf{a}_2$), then one would expect that data points that simultaneously belong to both the first and second cluster ($z_{i1} = 1 = z_{i2}$) should have the same mean as the first cluster ($\mathbf{a}_1$). While this is the case for the IOMM due to the overlapping model we define (2), this is not the case in any of the factorial models described in the previous paragraph.

6 Experiments

Since the IOMM is a generative model, we first tried generating from the model using full covariance Gaussians (5). Figure 5 shows two illustrative datasets that were generated in 2D along with the Gaussians which represent each independent cluster in the model. The IBP sample (or Z matrix) from which Figure 5b was generated is given in Figure 2. The parameters for each independent cluster were drawn from the prior (Normal-Inverse Wishart), and the data points were drawn from the product of Gaussians which corresponded to their cluster assignments from the Z matrix. We can see from these figures that even with a small number of components the IOMM can generate richly structured data sets.

We also generated data from the IOMM with Bernoulli clusters, and then used this synthetic data to test IOMM learning. This synthetic data consisted of $N = 100$ data points in $D = 32$ dimensions, and had $K = 11$ underlying independent clusters. We ran our MCMC sampler for 4000 iterations, burning in the first 1000. Because the clusters that specific columns in the Z matrix correspond to can be permuted, we cannot directly compare the learned Z matrix to the true Z matrix which generated the data. Instead we compute the matrix $U = ZZ^\top$, which is invariant to column permutations. This $N \times N$ matrix computes the number of shared clusters between each pair of data points in the data set, and is therefore a good column invariant way of determining how well the underlying cluster assignment structure is being discovered. Since we have many MCMC samples from which to compute the learned U matrix (which we will call $\hat{U}$), we average all the U matrices together to get $\hat{U}$. The true U matrix, U^*, is constructed from the true Z matrix. Both U^* and $\hat{U}$ are shown in Figure 6. Since U^* is a single sample and $\hat{U}$ is averaged over many samples, $\hat{U}$ is a little lighter (it is reasonable to expect that a few samples will assign a data point to even improbable clusters) and smoother, but the structure of $\hat{U}$ is extremely similar to that of U^*. We then rounded the values in $\hat{U}$ to the nearest integer and compared with U^*. Table 1 provides summary statistics for $\hat{U}$ in terms of the percentage of pairs of data points in $\hat{U}$ which share the exact same number of clusters as the same pair in U^*, differ by at most 1 cluster, and differ by at most 2 clusters. Figure 8 is a box plot showing the distribution of the number of inferred overlaps in $\hat{U}$ for each true number of overlaps in U^*. We can see that the model gives reasonable estimates of the number of overlaps, but is less able to estimate the rare cases of large numbers of overlaps. Lastly, Figure 7 plots the inferred number of clusters at each MCMC iteration, and suggests reasonable mixing of the sampler.

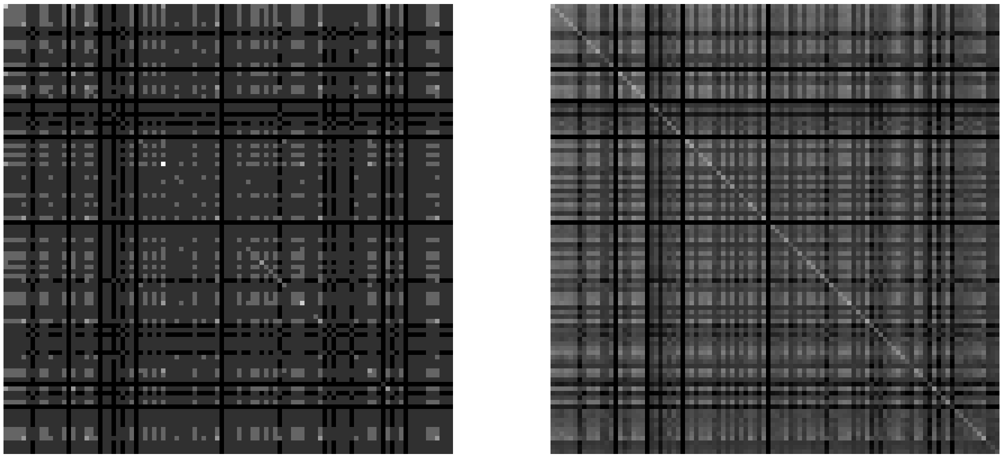

Figure 6: The U^* (left) and learned $\hat{U}$ (right) matrices showing the number of shared clusters for each pair of data points in the synthetic data set.

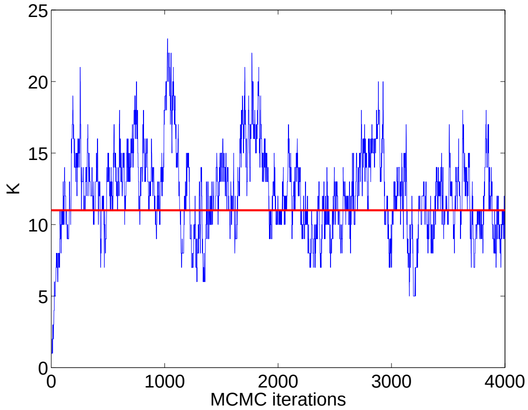

Figure 7: The number of discovered clusters, K, across MCMC iterations. The true number of clusters is marked in red (11).

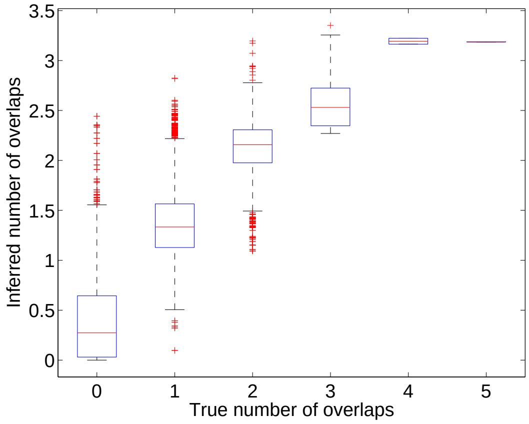

Figure 8: A box plot showing the distribution of inferred number of shared clusters in $\hat{U}$ for each true number of shared cluster in U^*, for every data point pair.

Statistic	Percent
$\lvert(\hat{U} - U^*)\rvert \leq 0$	69.96
$\lvert(\hat{U} - U^*)\rvert \leq 1$	99.12
$\lvert(\hat{U} - U^*)\rvert \leq 2$	100.00

Table 1: Summary statistics for learned $\hat{U}$. Reports the percentage of pairs in $\hat{U}$ which have the same number of shared clusters as the same pair in U^*, or are off by at most 1 or 2 shared clusters.

Lastly, we used the IOMM to cluster movies by genre using the MovieLens data set of people rating movies. We normalized the data over movies such that the ratings for each movie summed to 1 and then binarized the matrix so that a (movie,user) entry was given a value 1 if the new rating value was greater than the mean of the values of all movies that user rated. We then removed users with less than 20 movies given value 1, and movies which less than 10 users assigned a value 1 to. This resulted in a binary matrix of 797 movies by 426 users from which we selected 500 movies at random. These 500 movies belonged to 18 different genres. Unfortunately, an unsupervised learning algorithm does not know what a genre is, and would be very unlikely to cluster movies in accordance with them unless we specify them in advance. In particular people's movie preferences are not simply correlated with genres, and there are many other latent factors which can determine preference (e.g. actors, budget, recency, script, etc.) Instead, we took a semi-supervised approach, randomly selecting 200 movies, fixing the Z matrix for those data points to their correct genres, and trying to learn the remaining 300 movies using the cluster information given by the fixed 200. We ran our IOMM sampler for 3000 iterations, burning in the first 1000 samples. If a movie was assigned to a genre in over half the sampled Z matrices, we said that the movie was assigned to that genre by the IOMM. We compared these IOMM results to two sets of results obtained by using a Dirichlet Process Mixture model (DPM), which can only assign each movie to a single genre. DPM inference was run semi-supervised on the same data set by replicating each of the 200 fixed movies m_i times, once for each of the m_i genres they belong to. We compared the IOMM results to the DPM results using an F1 score, which takes into account both precision and recall, and which can be computed from the true MovieLens assignments of movies to genres. The difference between the two sets of DPM results is that in DPM1 genre membership is decided in the same way as in the IOMM, thus allowing movies to belong to only one genre. In DPM2, we allow movies to belong to multiple genres by saying that a movie belongs to a genre if the movie was assigned to that genre in at least $M/(K+1)$ samples, where M is the total number of samples and K is the known *true* number of genres that movie actually belongs to. These results are presented in table 2, on the 11 genres with at least 10 movie members in the fixed set.

Genre	# Movies	F1 IOMM	F1 DPM1	F1 DPM2
Drama	183	**0.4978**	0.2953	0.3046
Comedy	168	**0.6032**	0.5000	0.4962
Romance	81	**0.3030**	0.2581	0.2581
Action	78	0.5696	**0.6667**	**0.6667**
Thriller	72	**0.2737**	0.1404	0.1333
Adventure	50	**0.3091**	0.0000	0.0000
Children	46	0.3434	0.5714	**0.6047**
Horror	45	**0.7826**	0.6667	0.6780
Sci-Fi	38	**0.3256**	0.1000	0.0952
Crime	34	**0.2745**	0.1818	0.1818
Animation	21	**0.2667**	0.1429	0.1429

Table 2: The F1 scores for the IOMM versus the DPM by genre. The first column is the genre name, the second column is the number of movies in the data set which belong to that genre, the third column is the IOMM F1 score, the fourth column is the DPM1 F1 score, and the last column is the DPM2 F1 score for that genre.

We can see that the IOMM has a better F1 score on 9 of the 11 genres, illustrating that the flexibility of assigning movies to multiple genres leads to better performance even when evaluating single genre membership. It is worth noting that the DPM in this case is fully conjugate and that we took care to integrate out all parameters, resulting in a sampler with much faster mixing. Despite this, the DPM was not able to capture the genre assignments as well as the IOMM.

7 Discussion

We presented a new nonparametric Bayesian method, the Infinite Overlapping Mixture Model, for modeling overlapping clusters. The IOMM extends traditional mixture models to allow data points membership in an unrestricted number of clusters, where the total number of clusters is itself unbounded. The IOMM uses products of models in the exponential family to model overlaps, allowing it to focus in on overlapping regions. We derived MCMC inference algorithms for the IOMM and applied it to the problem of clustering movies into genres, where we showed that its performance is superior to that of Dirichlet Process Mixtures, which restrict movies to a single genre. Our novel approach to discovering overlapping clusters should be applicable to data modeling problems in a wide range of fields.

References

A. Banerjee, C. Krumpelman, S. Basu, R. Mooney, and J. Ghosh. Model based overlapping clustering. In *KDD*, 2005.

A. Battle, E. Segal, and D. Koller. Probabilistic discovery of overlapping cellular processes and their regulation. *Journal of Computational Biology*, 12(7):909–927, 2005.

C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, 2001.

Z. Ghahramani. Factorial learning and the EM algorithm. In *NIPS*, 1995.

D. Görür, Jäkel, and C. Rasmussen. A choice model with inifinitely many latent features. In *ICML*, 2006.

T. Griffiths and Z. Ghahramani. Infinite latent feature models and the indian buffet process. Technical report, Gatsby CNU, 2005.

T. Griffiths and Z. Ghahramani. Infinite latent feature models and the indian buffet process. In *NIPS*, 2006.

G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14, 2002.

G. E. Hinton and R. Zemel. Autoencoders, minimum description length, and helmholtz free energy. In *NIPS*, 1994.

X. Lu, M. Hauskrecht, and R. Day. Modeling cellular processes with variational bayesian cooperative vector quantizer. In *PSB*, 2004.

E. Meeds, Z. Ghahramani, S. Roweis, and R. Neal. Modeling dyadic data with binary latent factors. In *NIPS*, 2007.

M. Meila and J. Shi. Learning segmentation by random walks. In *NIPS*, 2000.

R. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9, 2000.

M. Sahami, M. A. Hearst, and E. Saund. Applying the multiple cause mixture model to text categorization. In *ICML*, 1996.

E. Saund. Unsupervised learning of mixtures of multiple causes in binary data. In *NIPS*, 1994.

E. Segal, A. Battle, and D. Koller. Decomposing gene expression into cellular processes. In *PSB*, 2003.