Multiplicative Mixture Models for Overlapping Clustering

Qiang Fu
Dept of Computer Science & Engineering
University of Minnesota, Twin Cities
qifu@cs.umn.edu

Arindam Banerjee
Dept of Computer Science & Engineering
University of Minnesota, Twin Cities
banerjee@cs.umn.edu

Abstract

The problem of overlapping clustering, where a point is allowed to belong to multiple clusters, is becoming increasingly important in a variety of applications. In this paper, we present an overlapping clustering algorithm based on multiplicative mixture models. We analyze a general setting where each component of the multiplicative mixture is from an exponential family, and present an efficient alternating maximization algorithm to learn the model and infer overlapping clusters. We also show that when each component is assumed to be a Gaussian, we can apply the kernel trick leading to non-linear cluster separators and obtain better clustering quality. The efficacy of the proposed algorithms is demonstrated using experiments on both UCI benchmark datasets and a microarray gene expression dataset.

1 Introduction

The problem of finding overlapping clusters, where an object can potentially belong to one or more clusters, has been gaining importance in a wide variety of application domains. For example, in social network analysis, since actors can potentially belong to multiple communities, community extraction algorithms should be able to detect overlapping clusters; in computational biology, overlapping clustering is a necessary requirement in the context of microarray analysis and protein function prediction, since a protein can potentially have multiple functions.

In this paper, we present *Multiplicative Mixture Models* (MMMs) as an appropriate framework for overlapping clustering. MMMs are designed to generate overlapping clusters and they can work with a variety of conditional distributions, including all exponential families. We propose an efficient EM-style alternating maximization algorithm for estimation and inference, whereas related models in literature primarily rely on stochastic approximation based on sampling, which can be slow for large scale problems. Further, we show that the proposed model can be kernelized thereby allowing non-linear cluster separators as well as extending its applicability to non-vector data on which a kernel can be suitably defined. We demonstrate that the proposed MMMs can be useful for overlapping clustering and the algorithm scales to large datasets.

The rest of the paper is organized as follows: In Section 2, we present MMMs using exponential family distributions as mixture components. In Section 3, we present an efficient overlapping clustering algorithm that alternates between inference and parameter estimation. In Section 4, we propose a kernelized overlapping clustering algorithm based on Gaussian MMMs. In Section 5, we present experimental results to demonstrate the efficacy of MMMs for overlapping clustering. We present related work in Section 6 and conclude in Section 7.

2 Multiplicative Mixture Models

Consider the traditional additive mixture model with k components whose density function is given by:

$$p(\mathbf{x}|\Theta) = \sum_{j=1}^{k} \pi_j p_j(\mathbf{x}|\theta_j) \ , \tag{1}$$

where π_j is the mixing weight for component j, $p_j(\mathbf{x}|\theta_j)$ is the probability density function for component j parameterized by θ_j and $\mathbf{x}$ is the data point under consideration. From a generative model perspective, one first samples a component j with probability π_j, and then sample $\mathbf{x} \sim p_j(.|\theta_j)$. The model assumes each $\mathbf{x}$ to have been generated from one component, making the model unsuitable for overlapping clustering.

A few alternative approaches to mixture modeling based overlapping clustering have been proposed in recent years [3, 1, 12]. In this paper, we consider a multiplicative mixture model motivated by the product-of-experts model of [9], more recently presented in the context of mixture modeling by [8]. For k mixture components, we assume a (latent) binary vector $\mathbf{z} = [z_1, \ldots, z_k]$ such that the conditional probability

$$p(\mathbf{x}|\mathbf{z},\Theta) = \frac{1}{c(\mathbf{z})} \prod_{j=1}^{k} p_j(\mathbf{x}|\theta_j)^{z_j} \ , \tag{2}$$

where $c(\mathbf{z})$ is a normalization constant. The latent boolean vector $\mathbf{z}$ indicates which components participated in generating $\mathbf{x}$, and $z_j \in \{0,1\}$ without any restrictions. If $\pi(\mathbf{z})$ defines an appropriate prior over $\mathbf{z}$, then we have

$$p(\mathbf{x}|\Theta) = \sum_{\mathbf{z}} \frac{\pi(\mathbf{z})}{c(\mathbf{z})} \prod_{j=1}^{k} p_j(\mathbf{x}|\theta_j)^{z_j} \ . \tag{3}$$

From a generative model perspective, one samples $\mathbf{z}$ with probability $\pi(\mathbf{z})$, and then samples $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z},\Theta)$. Since $\mathbf{z}$ can have multiple components as 1, the model is clearly well suited for overlapping clustering.

There are, however, two issues with the above multiplicative model. First, the model may not be well defined when $\mathbf{z} = \mathbf{0}$, the all zeros vector. We emphasize that this case should not be ignored by setting $\pi(\mathbf{0}) = 0$, or something equivalent. In several real life datasets, there are points which do not naturally belong to any cluster. Rather than forcing them into an existing cluster, it may be more meaningful to have a model which can potentially leave a few points un-clustered. Secondly, since $\mathbf{z}$ is a boolean vector of size k, inference methods may need to go over all 2^k possible states for each data point. Even with $k = 20$, it amounts to considering a million states for each point in each iteration. In practice, we want inference algorithms that are a few orders of magnitude faster, while maintaining reasonable accuracy. We focus on the modeling issues in the rest of this section, and develop efficient algorithms in Section 3.

2.1 Exponential Family Mixtures

To make the discussion concrete, we focus on multiplicative mixture models (MMMs) where the components are exponential family distributions. Recall that a distribution is in the exponential family if the density function with respect to a base measure can be written in the form:

$$p(\mathbf{x}|\theta) = \frac{dP(\mathbf{x}|\theta)}{dP_0(\mathbf{x})} = \exp\{s(\mathbf{x})^T\theta - \psi(\theta)\} \ , \tag{4}$$

where θ is the natural parameter, $s(\mathbf{x})$ is the sufficient statistic, and $\psi(\theta)$ is the cumulant function, which is a convex function of Legendre type [11]. Without loss of generality, we assume $\psi(0) = 0$. With the component distributions being from the same exponential family, the conditional probability in (2) becomes

$$p(\mathbf{x}|\Theta,\mathbf{z}) = \frac{1}{c(\mathbf{z})} \exp\left\{ \sum_{j=1}^{k} z_j s(\mathbf{x})^T\theta_j - z_j\psi(\theta_j) \right\} \ . \tag{5}$$

A direct calculation shows that:

$$c(\mathbf{z}) = \exp\left\{ \psi\left(\sum_{j=1}^{k} z_j\theta_j \right) - \sum_{j=1}^{k} z_j\psi(\theta_j) \right\} \ .$$

Making use of the closed form for $c(\mathbf{z})$, we have

$$p(\mathbf{x}|\Theta,\mathbf{z}) = \exp\left\{ s(\mathbf{x})^T \sum_j z_j\theta_j - \psi\left(\sum_{j=1}^{k} z_j\theta_j \right) \right\} \ . \tag{6}$$

So $p(\mathbf{x}|\Theta,\mathbf{z})$ is in the same exponential family as the component distributions, with natural parameter $\sum_j z_j\theta_j$.

2.2 The Noise Component

When $\mathbf{z} = \mathbf{0}$, from (5) it follows that $p(\mathbf{x}|\mathbf{z},\Theta) = 1$, which may not be a well defined density function depending on the domain of $\mathbf{x}$ as well as the choice of the base measure $P_0(\mathbf{x})$. Intuitively, the points corresponding to $\mathbf{z} = \mathbf{0}$ may be considered as "noise" in that they do not follow the cluster structure implied by the multiplicative mixture model. To incorporate this into the generative model, we introduce another parametric exponential family as the noise component. The noise component does not necessarily come from the same exponential family as other base components.

Introducing another (latent) boolean variable z_{k+1} for the noise component, which is 1 only when $\mathbf{z} = \mathbf{0}$, and 0 otherwise, the conditional probability for the new model is given by

$$p(\mathbf{x}|\mathbf{z},z_{k+1},\Theta) = \frac{1}{c(\mathbf{z})} \prod_{j=1}^{k+1} p_j(\mathbf{x}|\theta_j)^{z_j} \ . \tag{7}$$

2.3 Generative Model

A complete specification of the model requires an appropriate prior $\pi(\mathbf{z})$ over $\mathbf{z}$. Since $\mathbf{z}$ is a boolean vector, we assume each component z_j to be sampled from a Bernoulli distribution ϕ_j, which itself has been drawn from a Beta distribution $\mathrm{Beta}(\alpha_j,\beta_j)$. The generative model for a sample $\mathbf{x}$ can be described as follows:

1. Draw $\phi_j|\{\alpha_j,\beta_j\} \sim \mathrm{Beta}(\alpha_j,\beta_j)$, for $j = 1,\ldots,k$.
2. Draw $z_j|\phi_j \sim \mathrm{Bernoulli}(\phi_j)$, for $j = 1,\ldots,k$.
3. If $\mathbf{z} = \mathbf{0}$, $z_{k+1} = 1$, else $z_{k+1} = 0$.
4. Draw $\mathbf{x}|\{\mathbf{z},z_{k+1},\Theta\} \sim \frac{1}{c(\mathbf{z})}\prod_{j=1}^{k+1} p_j(\mathbf{x}|\theta_j)^{z_j}$.

Based on the above model, the joint distribution

$$p(\mathbf{x},\mathbf{z},\boldsymbol{\phi}|\boldsymbol{\alpha},\boldsymbol{\beta},\Theta) = \frac{1}{c(\mathbf{z})} \left(\prod_{j=1}^{k} p(\phi_j|\alpha_j,\beta_j)p(z_j|\phi_j) \right) \left(\prod_{j=1}^{k+1} p(\mathbf{x}|\theta_j)^{z_j} \right) .$$

The marginal distribution $p(\mathbf{x}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \Theta)$ can be obtained by integrating out the latent variables $(\phi_j, z_j), j = 1, \ldots, k$.

3 Overlapping Clustering Algorithm

Given a set of data points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, the task in overlapping clustering based on MMMs is to simultaneously estimate the set of parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \Theta)$ in the model, as well as infer the latent cluster assignment vector $\mathbf{z}$ for each data point $\mathbf{x}$. In this paper, we formulate the problem as one of finding the mode of the joint distribution of the observable and the corresponding latent cluster assignment $p(\mathbf{x}, \mathbf{z}|\alpha, \beta, \Theta)$. Noting that $(\mathbf{x}, \mathbf{z})$ for different data points are conditionally independent, the problem can be posed as maximizing the following objective function:

$$\begin{aligned} L(\mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \Theta) &= \sum_{i=1}^{n} \log p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\alpha}, \boldsymbol{\beta}, \Theta) \\ &= \sum_{i=1}^{n} \log p(\mathbf{z}_i|\boldsymbol{\alpha}, \boldsymbol{\beta}) + \sum_{i=1}^{n} \log p(\mathbf{x}_i|\mathbf{z}_i, \Theta) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{k} \log \left(\int_{\phi_{i,j}} p(z_{i,j}|\phi_{i,j}) p(\phi_{i,j}|\alpha_j, \beta_j)\, d\phi_{i,j} \right) \\ &\quad + \sum_{i=1}^{n} \left(\sum_{j=1}^{k+1} z_{i,j} \log p(\mathbf{x}_i|\theta_j) - \log c(\mathbf{z}_i) \right) . \end{aligned} \tag{8}$$

Based on the above objective function, we propose an EM-style alternating maximization algorithm to do inference and estimation. In the E- or inference step, given a set of parameter values $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \Theta)$, we optimize L with respect to $\mathbf{z}_i, i = 1, \ldots, n$. In the M- or estimation step, for a given set of overlapping clusterings $\mathbf{z}$, we optimize L over the parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \Theta)$. The alternating iterations are assumed to have converged when either no $\mathbf{z}_i$ changes in the inference step, or when the maximum absolute change over all parameters in the estimation step is below a threshold.

3.1 Inference

First, we focus on the inference step, which maximizes L over $\mathbf{z}$ given the parameters. A naive approach to optimizing over $\mathbf{z}$ is to try every possible value of $\mathbf{z}$, and choose the one which gives the highest log-likelihood. Such an approach has to go over 2^k possibilities for each $\mathbf{x}$ in each iteration. As a result, such an approach will be computationally inefficient and impractical even for moderate k. An alternative approach is to use a fast heuristic which ensures that the log-likelihood is non-decreasing. We follow this strategy by adopting an idea from the literature [1].

For any $\mathbf{x}$, let $\mathbf{z}_0$ be the assignment vector from the previous inference step, let $\mathbf{e}_j, j = 1, \ldots, k$, be the boolean vector with the j^{th} component being 1, and all else zero, and E be the set of all such vectors. The heuristic tries k threads $t_j, j = 1, \ldots, k$, each starting with $\mathbf{z}_{1j} = (\mathbf{z}_0 + \mathbf{e}_j) \bmod 2, j = 1, \ldots, k$. In any thread, the algorithm first computes the log-likelihood for $\mathbf{z}_{1j}$; then, the algorithm finds the best assignment among $\mathbf{z}_{2jj'} = (\mathbf{z}_{1j} + \mathbf{e}_{j'}) \bmod 2$, where $\mathbf{e}_{j'} \in E \setminus \{\mathbf{e}_j\}$; in the next step, the best assignment among $\mathbf{z}_{3jj'j''} = (\mathbf{z}_{2jj'} + \mathbf{e}_{j''}) \bmod 2$, where $\mathbf{e}_{j''} \in E \setminus \{\mathbf{e}_j, \mathbf{e}_{j'}\}$; and so on. If the best $\mathbf{z}$ at any step is better than the best at the next step, the thread terminates setting $\mathbf{z}_{j*} = \mathbf{z}$. Finally, the algorithm picks the best $\mathbf{z}_{j*}$ among $j = 1, \ldots, k$. Since there are k threads, each thread has at most k steps, and each step has at most k evaluations of the log-likelihood, the complexity of the heuristic is $O(k^3)$ (the number of evaluations is at most $k\binom{k}{2}$). Furthermore, it is guaranteed to give an assignment $\mathbf{z}$ that is at least as good as the old assignment, so that the log-likelihood is non-decreasing over iterations. In practice, the heuristic takes much less than $k\binom{k}{2}$ and is very fast, making it appropriate for large datasets with moderate to large k.

3.2 Estimation

In the estimation step, for a given set of overlapping cluster assignments, we optimize L over the parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \Theta)$. The optimization can be broken into two independent parts—one over the parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ of the Beta distributions, and one over the natural parameters Θ of the component exponential family distributions. For a given set of $\mathbf{z}$, let m_j be the total number of $z_{i,j}$ that are 1, so that $(n - m_j)$ is the total number of $z_{i,j}$ that are 0. A direct calculation based on taking derivatives w.r.t. (α_j, β_j) and setting it to 0 shows that the optimal parameters satisfy the following equation:

$$\frac{\alpha_j}{\beta_j} = \frac{m_j}{n - m_j} .$$

Setting $\beta_j = 1$, we only update $\alpha_j = m_j/(n - m_j), j = 1, \ldots, k$ in each iteration.

The dependency on the component model parameters is captured by the second term in (8). We show that for any exponential family distribution, the objective function L is concave in each θ_j given all other parameters are held constant. Using (6) in (8), the second term of the objective function can be written as a function of Θ given by

$$\begin{aligned} f(\Theta) &= \sum_{i=1}^{n} \left(\sum_{j=1}^{k+1} z_{i,j} \log p(\mathbf{x}_i|\theta_j) - \log c(\mathbf{z}_i) \right) \\ &= \sum_{i=1}^{n} \left[s(\mathbf{x}_i)^T \sum_{j=1}^{k+1} z_{i,j}\theta_j - \psi \left(\sum_{j=1}^{k+1} z_{i,j}\theta_j \right) \right] . \end{aligned}$$

Since ψ is the cumulant of an exponential family, it is a convex function of Legendre type [4], implying that it is in C^{∞}. Computing the second derivative of $f(\Theta)$ with respect to θ_j, we have

$$\nabla^2_{\theta_j} f(\Theta) = -\sum_{i=1}^{n} z_{i,j} \nabla^2_{\theta_j} \psi \left(\sum_{h=1}^{k+1} z_{i,h}\theta_h \right) ,$$

which is negative, since ψ is a convex function implying $\nabla^2\psi$ is positive. Hence, $f(\Theta)$ is a concave function of θ_j. In order to find the maximizer θ_j^* given all the other parameters $\theta_h, h \neq j$, taking gradient and setting it to 0, we obtain

$$\theta_j^* = \sum_{i:z_{i,j}=1} \sum_{\substack{h=1 \\ h\neq j}}^{k+1} z_{i,h}\theta_h + (\nabla\psi)^{-1} \left(\sum_{i:z_{i,j}=1} s(\mathbf{x}_i) \right) . \quad (9)$$

Since ψ is a Legendre function, the function $(\nabla\psi)^{-1}$ will be well defined and equal to $\nabla\phi$, where $\phi = \psi^*$, the conjugate of the cumulant function ψ. The actual update equation for any exponential family can be derived by plugging in the specific cumulant function ψ and sufficient statistics $s(\mathbf{x})$.

4 Kernelized Overlapping Clustering

In this section, we show that the proposed multiplicative model can be kernelized, and the overlapping clustering algorithm can be extended to the general case. There are two key advantages to the kernelized extension: (i) Individual base clusters can be separated by non-linear boundaries, making the approach applicable to more complex data, and (ii) Overlapping clustering can be applied to structured data, such as strings, trees, graphs, etc., for which a meaningful kernel can be defined [13]. To make the kernelized extension, we implicitly map the data points to a high dimensional space and assume that in the high dimensional space there are k spherical Gaussian clusters. If $\phi(.)$ is the mapping function, a Gaussian in the high dimensional space can be represented as :

$$p(\phi(\mathbf{x})|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{\exp\left(-\frac{1}{2}(\phi(\mathbf{x})-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\phi(\mathbf{x})-\boldsymbol{\mu})\right)}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}}$$
$$= \frac{\exp\left(-\frac{a}{2}\langle\phi(\mathbf{x}),\phi(\mathbf{x})\rangle + a\langle\phi(\mathbf{x}),\boldsymbol{\mu}\rangle - \frac{a}{2}\langle\boldsymbol{\mu},\boldsymbol{\mu}\rangle\right)}{(2\pi)^{D/2}a^{-D/2}} ,$$

where $a = \frac{1}{\sigma^2}$ is the inverse of the Gaussian variance so that $\boldsymbol{\Sigma}^{-1} = a\mathbb{I}$, $\boldsymbol{\mu}$ is the mean of the Gaussian and D is the feature dimension. Plugging the above expression into (7), the log-likelihood of MMM with respect to a single data point $\mathbf{x}$ becomes :

$$\log p(\phi(\mathbf{x})|\mathbf{z},\bar{\boldsymbol{\mu}},\bar{\boldsymbol{\Sigma}}) = -\frac{D}{2}\log(2\pi) + \frac{D}{2}\log(\bar{a})$$
$$-\frac{\bar{a}}{2}\langle\phi(\mathbf{x}),\phi(\mathbf{x})\rangle + \bar{a}\langle\phi(\mathbf{x}),\bar{\boldsymbol{\mu}}\rangle - \frac{\bar{a}}{2}\langle\bar{\boldsymbol{\mu}},\bar{\boldsymbol{\mu}}\rangle , \quad (10)$$

where $\bar{a} = \sum_{j=1}^{k+1} z_j a_j$, $\bar{\boldsymbol{\mu}} = \frac{\sum_{j=1}^{k+1} z_j a_j \boldsymbol{\mu}_j}{\bar{a}}$ and $\bar{\boldsymbol{\Sigma}}^{-1} = \bar{a}\mathbb{I}$.

A direct calculation for the estimation step shows when each component in MMM is a Gaussian, the mean of each Gaussian $\boldsymbol{\mu}_j, j = 1,\ldots,k$ can be estimated using an appropriate linear combination of all the data points $\phi(\mathbf{x}_i), i = 1,\ldots,n$. Let $\boldsymbol{\mu}_j = \sum_{i=1}^{n} c_{i,j}\phi(\mathbf{x}_i)$ and $c_{i,j} \in \mathbb{R}$, we have:

$$\langle\phi(\mathbf{x}),\bar{\boldsymbol{\mu}}\rangle = \frac{1}{\bar{a}}\sum_{j=1}^{k+1}\langle\phi(\mathbf{x}), z_j a_j \boldsymbol{\mu}_j\rangle = \frac{1}{\bar{a}}\sum_{j=1}^{k+1}\sum_{i=1}^{n} z_j a_j c_{i,j}\langle\phi(\mathbf{x}),\phi(\mathbf{x}_i)\rangle ,$$

$$\langle\bar{\boldsymbol{\mu}},\bar{\boldsymbol{\mu}}\rangle = \frac{1}{\bar{a}^2}\langle\sum_{j=1}^{k+1} z_j a_j \boldsymbol{\mu}_j, \sum_{j'=1}^{k+1} z_{j'} a_{j'} \boldsymbol{\mu}_{j'}\rangle$$
$$= \frac{1}{\bar{a}^2}\sum_{j=1}^{k+1}\sum_{j'=1}^{k+1} z_j z_{j'} a_j a_{j'} \sum_{i=1}^{n}\sum_{i'=1}^{n} c_{i,j}c_{i',j'}\langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_{i'})\rangle .$$

Suppose the kernel similarity matrix is K, replacing the inner product $\langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_{i'})\rangle$ with $K(\mathbf{x}_i,\mathbf{x}_{i'})$, and plugging in the kernelized terms back in (10), we obtain the objective function for kernelized overlapping clustering algorithm. The inference and estimation step remains the same, except that we need to estimate $c_{i,j}, j = 1,\ldots,k+1$, instead of $\boldsymbol{\mu}_j, j = 1,\ldots,k+1$.

5 Experimental Results

5.1 UCI Datasets

We run the overlapping clustering algorithm on 8 UCI datasets (Table 1). For all experiments reported, we set k to be the true number of classes, and use multivariate Gaussian with diagonal covariance matrix to model each cluster. We adopt the semi-supervised seeding approach [2] to do initialization: we randomly select 10% of the data points from each class and each base cluster is initialized using the means and variances of the selected data points from the class. We run the algorithm on each dataset 5 times with different initialization and report the result based on the one which has the highest log-likelihood.

To make comparisons, we use 2 baseline algorithms. The first one is the overlapping clustering algorithm described in [3], which we refer to as BSK algorithm. The second one is the EM algorithm based on Gaussian additive mixture models. To get overlapping clustering, we threshold the posterior probability: for a given threshold t, if for any cluster j the posterior probability $p(j|\mathbf{x}) \geq t$, we consider $\mathbf{x}$ belongs to cluster j. The initialization and convergence criterion are the same for all the algorithms.

Since the UCI datasets do not have overlapping labels, we evaluate the algorithms using predictions based on the overlapping clustering. We study the overlapping data points, which belong to more than one clusters, with the

	Ratio 1	Ratio 2 (Precision)					Ratio 3 (Recall)				
		MMM	BSK	Thresholded EM			MMM	BSK	Thresholded EM		
				0.01	0.1	0.2			0.01	0.1	0.2
Iris	0.1800	**0.6250**	N/A	0.5172	0.6250	**0.6364**	**0.5556**	0	**0.5556**	0.3704	0.2592
Ionosphere	0.7493	**0.9223**	**0.7778**	**0.8462**	0.7143	0.6667	**0.3612**	0.0266	0.0418	0.0190	0.0076
Vowel	0.6913	**0.8537**	N/A	0.6892	0.7609	**0.7778**	0.1918	0	**0.2795**	0.0959	0.0575
Wdbc	0.1002	**0.2857**	N/A	0.6000	0.7000	**0.7778**	**0.6667**	0	0.1579	0.1228	0.1228
Pima	0.5208	**0.6626**	N/A	0.6049	**0.7226**	0.6629	0.2700	0	**0.4975**	0.2475	0.1475
Segment	0.1242	**0.1338**	**0.6667**	**0.2289**	0.2157	0.1818	**0.5958**	0.0139	0.0662	0.0383	0.0209
Landsat	0.2810	**0.3872**	N/A	0.5582	**0.5882**	0.5645	**0.7129**	0	0.0769	0.0387	0.0194
Pendigits	**0.0890**	0.0658	N/A	0.0687	0.0622	0.0388	**0.1779**	0	0.0327	0.0133	0.0051

Table 2. Overlapping points have larger fraction of support vectors, i.e, Ratio 2 $>$ Ratio 1. MMM performs substantially better than BSK. Thresholded EM can have reasonable precision for some (high) thresholds, but gives poor recall on many datasets.

	Iris	Ionosphere	Vowel	Wdbc	Pima	Segment	Landsat	Pendigits
k	3	2	11	2	2	10	6	10
d	4	32	10	30	8	16	36	15
n	150	351	528	569	768	2310	6435	10922

Table 1. Data Sets.

following hypothesis—overlapping points lie close to the boundary of classes, and have higher chance of becoming support vectors in a SVM classifier. We also expect that the set of overlapping data points has a reasonable intersection with that of support vectors. To test the hypothesis, we train a SVM classifier, based on LIBSVM [5], using linear kernel and default parameter settings on each dataset, and obtain the support vectors. Then, for each dataset, we compute the following three ratios: Ratio 1 is the fraction of support vectors in the data set, i.e., $\frac{|\text{Support Vectors}|}{n}$. Ratio 2 (Precision) is the fraction of overlapping points that are support vectors, i.e., $\frac{|\text{Overlapping}\cap\text{Support Vectors}|}{|\text{Overlapping}|}$ and Ratio 3 (Recall) is the fraction of support vectors that are overlapping points, i.e.,$\frac{|\text{Overlapping}\cap\text{Support Vectors}|}{|\text{Support Vectors}|}$.

Based on our hypothesis, we expect Ratio 1 $<$ Ratio 2 and a reasonable Ratio 3. The result is listed in Table 2. For the overlapping clustering algorithm, the hypothesis is valid on 7 datasets. However, BSK algorithm either fails to find any overlapping points on 6 datasets (Ratio 2 is N/A) or finds only few overlapping data points (9 for Ionosphere and 6 for Segment). For EM algorithm, Ratio 2 is larger than Ratio 1 in most cases, but Ratio 3 is usually very small, which indicates that additive mixture model tends to give few overlapping points. This observation can be explained as follows: if one of the posterior probabilities $p(j|\mathbf{x})$ is large, all the other posterior probabilities will become relatively smaller since $\sum_{j=1}^{k} p(j|\mathbf{x}) = 1$. So when we threshold on the posterior probability, we get very few overlapping data points. The phenomenon is obvious for datasets with larger values of k, such as Landsat, Vowel, and Segment.

5.2 Microarray Gene Expresssion Dataset

The microarray gene expression dataset [14] consists of 4062 yeast genes and 215 experimental conditions. Our goal is to cluster the genes into multiple biological processes based on the expression profiles. Since many genes are known to be multi-functional, we would expect that some genes participate in more than one biological processes, thus overlapping clustering is a natural approach for the problem. We report results on 1354 genes that have significant changes in the gene expression, i.e., 1/3 of the genes that have the highest variances of gene expression over the 215 experimental conditions. The number of clusters k is fixed to be 30. We still compare our algorithm with BSK algorithm [3] and EM based on Gaussian additive mixture models. We initialize all the algorithms based on the preliminary clustering result given by kmeans.

Overall, our overlapping clustering algorithm predicts that 556 genes participate in only one process, 552 in two, 219 in three and 27 in four or more, while BSK algorithm discovers that 95 genes do not belong to any process and 397 participate in only one process, 383 in two, 255 in three and 224 in four or more. For EM algorithm, we set the posterior probability threshold to be 0.01. The additive mixture model gives very few overlapping genes even under this low threshold: it predicts 1324 genes participate in only one process, 27 in two and 3 in three or more. This result further illustrates that additive mixture models may not be appropriate for overlapping clustering on complex datasets.

To evaluate whether the cluster assignments for the genes are reasonable from a biological perspective, we check if the genes in each learned biological process show any enrichment for known annotations. We make use of Gene Ontology Term Finder[1] online tool, which searches for shared annotations given a set of genes and computes an associated p-value. The p-value measures the probability of observing a group of genes to be annotated with a certain annotation purely by chance. If a cluster of genes indeed correspond to known biological processes, we would expect a low p-value. We consider an annotation to be significant if the p-value associated with it is less than 10^{-4}. Both the overlapping clustering algorithm and BSK algorithm discover 94 different significant annotations. Among the 62 common significant annotations, the overlapping clustering algorithm performs better in 37 (60%) of them with lower p-values. In case a significant annotation presents in more

[1] http://db.yeastgenome.org/cgi-bin/GO/goTermFinder.pl

Kernels	# Significant Anno.	BSK	Unkernelized Algo.
$\exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{250})$	107	67% (43/64)	67% (48/72)
$\exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{500})$	109	68% (43/63)	63% (54/86)
$\exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{750})$	101	75% (42/56)	60% (49/82)

Table 3. Kernelized overlapping algorithm consistently finds more significant enrichments than both BSK and the unkernelized overlapping clustering algorithm. For the fractions (a/b), b is the number of common significant annotations, and a is the number of times the kernelized algorithm has lower p-values for those common significant annotations.

than one learned processes in any algorithm, we pick the one with the lowest p-value.

We also test the kernelized overlapping clustering algorithm. To satisfy the assumption of using spherical Gaussians, we z-score the dataset. We try three different RBF kernels and specify the feature dimension D to be 1354, the number of genes we use in the experiment[2]. The detailed result is listed in Table 3. As the results show, the kernelized overlapping clustering algorithm performs favorably compared to the baseline overlapping clustering algorithms in terms of enrichment.

6 Related Work

Our MMMs are closely related to the Product of Experts (PoE) model proposed by [9]. The PoE model with k components has $p(\mathbf{x}|\Theta) = \frac{1}{c}\prod_{j=1}^{k} p_j(\mathbf{x}|\theta_j)$. If $\mathbf{z}$ is the all 1 vector in MMMs, then we exactly obtain the PoE model. For general $\mathbf{z}$, $p(\mathbf{x}|\mathbf{z},\Theta)$ is a PoE model over a subset of experts, chosen according to $\mathbf{z}$.

A non-parametric Bayesian model for overlapping clustering, due to [8], is also closely related to the proposed MMMs. The treatment in [8], focuses on the use of nonparametric priors based on the Indian Buffet Process [7], and uses Metropolis-Hastings to sample the model parameters Θ. The analysis for the case when $\mathbf{z}$ is all zero was not explicitly handled.

Another class of overlapping clustering models combines the expectation parameters of component distributions, rather than the natural parameters as in MMMs. For example, [3] use such an idea to discover overlapping processes from gene expression data. Their algorithm works with the observed real gene expression profiles X (genes × experiments), a hidden binary membership matrix Z (genes × processes) containing the membership of each gene in each process, and a hidden real activity matrix A (processes × expriments) containing the activity of each process for each experimental condition. The assumption of their model is $E[\mathbf{x}_i] = A\mathbf{z}_i$, i.e., each $\mathbf{x}_i$ is generated from a Gaussian distribution with mean $A\mathbf{z}_i$, which is sum of the activity levels of the processes that contribute to the generation of $\mathbf{x}_i$. Several related models with a similar generative structure have appeared in the literature in the form of factorial, multi-cause, or overlapping models [6, 12, 10, 1].

7 Conclusions

We have presented an overlapping clustering approach based on multiplicative mixture models (MMMs). The proposed MMMs inherently assume that each point is generated from a product of a subset of the component distributions. When each component distribution in a MMM is from an exponential family, we show that there is an efficient alternating maximization algorithm that converges to a (local) maxima of the joint likelihood of the observations and their assignments. We also show that when each component in a MMM is a multivariate Gaussian, we can use kernel techniques to get non-linear separators and obtain better clustering quality. In practice, the algorithms are accurate, fast, and scale to large datasets.

Acknowledgements: The research was supported by NSF grant IIS-0812183.

References

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

[2] S. Basu, A. Banerjee, and R. Mooney. Semi-supervised clustering by seeding. *ICML*, 2002.

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

[4] A. Banerjee, S. Merugu, I. Dhillon and J. Ghosh. Clustering with Bregman Divergences. *JMLR*, (7):1705–1749, 2005.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[6] Z. Ghahramani. Factorial learning and the EM algorithm. *NIPS*, 1995.

[7] T. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. TR 2005-001, Gatsby Computational Neuroscience Unit, 2005.

[8] K. A. Heller and Z. Ghahramani. A nonparametric Bayesian approach to modeling overlapping clusters. *AISTAT*, 2007.

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

[10] G. Hinton and R. Zemel. Autoencoders, minimum description length, and contrastive divergence. *NIPS*, 1994.

[11] R. T. Rockafellar. *Convex Analysis*. Princeton Landmarks in Mathematics. Princeton University Press, 1970.

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

[13] B. Scholkopf and A. J. Smola. *Learning with Kernels*. MIT Press Cambridge, MA, 2001.

[14] S. Mnaimneh, A. Davierwala, J. Haynes, J. Moffat, W.-T. Peng, W. Zhang, X. Yang, J. Pootoolal, G. Chua, and A. Lopez *Exploration of Essential Gene Functions via Titratable Promoter Alleles*. *Cell*, 118(1):31–44, 2004.

[2] As it is proved in [13], the largest possible D is the size of the dataset.