Model-based Overlapping Clustering

Arindam Banerjee Chase Krumpelman Joydeep Ghosh

Dept. of Electrical and Computer Engineering
University of Texas at Austin
Austin, TX 78712, USA

Sugato Basu Raymond J. Mooney

Dept. of Computer Sciences
University of Texas at Austin
Austin, TX 78712, USA

ABSTRACT

While the vast majority of clustering algorithms are partitional, many real world datasets have inherently overlapping clusters. Several approaches to finding overlapping clusters have come from work on analysis of biological datasets. In this paper, we interpret an overlapping clustering model proposed by Segal et al. [23] as a generalization of Gaussian mixture models, and we extend it to an overlapping clustering model based on mixtures of any regular exponential family distribution and the corresponding Bregman divergence. We provide the necessary algorithm modifications for this extension, and present results on synthetic data as well as subsets of 20-Newsgroups and EachMovie datasets.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - Data Mining; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms

Keywords

Overlapping clustering, exponential model, Bregman divergences, high-dimensional clustering, graphical model.

1. INTRODUCTION

Most clustering methods partition the data into non-overlapping regions, where each point belongs to only one cluster. In a variety of important applications, though, *overlapping clustering*, wherein some items are allowed to be members of two or more discovered clusters, is more appropriate. For example, in biology, genes often simultaneously participate in multiple processes; therefore, when clustering micro-array gene expression data, it is appropriate to assign genes to multiple, overlapping clusters [23, 4]. Similarly, when clustering documents into topic categories, documents may contain multiple relevant topics and an overlapping clustering might be more relevant [22]. In the *20-Newsgroups* benchmark dataset, articles with multiple topics are cross posted to multiple newsgroups. Ideally, a clustering algorithm applied to this data would allow articles to be assigned to multiple topic labels and would rediscover the original cross-posted articles. In the *EachMovie* dataset used to test recommender systems, many movies belong to more than one genre, such as "Aliens", which is listed in the action, horror and science fiction genres. An overlapping clustering algorithm applied to this data should automatically discover such multi-genre movies.

In this paper, we generalize an approach to overlapping clustering introduced by Segal et al. [23], hereafter referred to as the SBK model. The original method was presented as a specialization of a Probabilistic Relational Model (PRM) [14] and was specifically designed for clustering gene expression data. We present an alternative view of their basic approach as a generalization of standard mixture models. While the original model maximized likelihood over constant variance Gaussians, we generalize it to work with any regular exponential family distribution, and corresponding Bregman divergences, thereby making the model applicable for a wide variety of clustering distance functions [2]. This generalization is critical to the effective application of the approach to high-dimensional sparse data, such as typically those encountered in text mining and recommender systems, where Gaussian models and Euclidean distance are known to perform poorly. Further, we propose a novel algorithm `dynamicM` that assigns instances to multiple clusters for the general model. We also outline an alternating minimization algorithm that monotonically improves the objective function for overlapping models for any regular exponential family distribution.

In order to demonstrate the generality and effectiveness of our approach, we present experiments in which we produced and evaluated overlapping clusterings for subsets of the *20-Newsgroups* and *EachMovie* data sets mentioned above. An alternative "straw man" algorithm for overlapping clustering is to produce a standard probabilistic "soft" clustering by mixture modeling and then make a hard assignment of each item to one or more clusters using a threshold on the cluster membership probability. The ability of thresholded soft clustering to produce good overlapping clusterings is an open question. Consequently, we experimentally compare our approach to an appropriate thresholded soft clustering and show that the proposed overlapping clustering model produces groupings that are more similar to the original overlapping categories in the *20-Newsgroups* and *EachMovie* data.

A brief word on notation: uppercase letters such as X signify a matrix, whose i^{th} row vector is represented as X_i, j^{th} column vector is represented as X^j, and whose entry in row i and column j is represented as X_i^j as well as X_{ij}.

2. BACKGROUND

In this section, we give a brief introduction to the PRM-based

KDD'05, August 21–24, 2005, Chicago, Illinois, USA.

SBK model. *Probabilistic Relational Models* (PRMs) [14] extend the basic concepts of Bayesian networks into a framework for representing and reasoning with probabilistic relationships between entities in a relational structure. The SBK model is an instantiation of a PRM for capturing the relationships between genes, processes, and measured expression values on DNA microarrays. The structure of the instantiated model succinctly captures the underlying biological understanding of the mechanism generating the observed microarray values — namely, that genes participate in processes, experimental conditions cause the invocation of processes at varying levels, and the observed expression value in any particular microarray spot is due to the combined contributions of several different processes. The SBK model places no constraints on the number of processes in which any gene might participate, and thus gene membership in multiple processes, i.e., overlapping clustering, naturally follows.

The SBK model works with three matrices: the observed real expression matrix X *(genes* $\times$ *experiments)*, a hidden binary membership matrix M *(genes* $\times$ *processes)* containing the membership of each gene in each process, and a hidden real activity matrix A *(processes* $\times$ *conditions)* containing the activity of each process for each experimental condition. The SBK modeling assumes that the expression value X_i^j corresponding to gene i in experiment j has a Gaussian distribution with constant variance. The mean of the distribution is equal to the sum of the activity levels A_h^j of the processes h in which gene i participates so that $p(X_i^j|M_i,A) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2\sigma^2}(X_i^j - M_iA^j)^2)$. The SBK model further assumes that M and A are independent so that $p(M,A) = p(M)p(A)$ and that X_i^j's are conditionally independent given M_i and A^j. M and A are assumed to be component-wise independent as well. Assuming that elements of A are uniformly distributed, considering the log-likelihood of the joint distribution, we have

$$\max_{M,A} \log p(X,M,A) \equiv \min_{M,A} \left[\frac{1}{2\sigma^2}\|X - MA\|^2 - \log p(M)\right].$$

To find the value of the hidden variables M and A, the SBK model uses an EM approach [12]. The E step involves finding the best estimates of the binary genes-process memberships M. The M step involves computing the prior probability of gene membership in each process $p(M)$ and the process-condition activations A.

The core parameter estimation problem is much easier to understand if we recast it as a matrix decomposition problem, initially ignoring the priors. With the knowledge that there are k relevant processes in the observations, we want to find a decomposition of the observed expression matrix $X \in \mathbb{R}^{n\times d}$ into a binary membership matrix $M \in \{0,1\}^{n\times k}$ and a real valued activation matrix $A \in \mathbb{R}^{k\times d}$ such that $||X - MA||^2$ is minimized. Hence, the problem is one of matrix factorization, and the difficulty arises from the fact that M is a binary matrix.

3. THE MODEL

In this section, we first outline a simplistic way of getting overlaps from soft-clustering based on mixture models. Then, we propose our model for overlapping clustering, hereafter referred to as MOC, as a generalization of the SBK model.

3.1 Overlapping Clustering with Mixture Model

Given a set of n data points $\{X_i\}_{i=1}^n$ in $\mathbb{R}^d$, represented by a $n \times d$ matrix X, fitting a mixture model to X is equivalent to assuming that each data point X_i is drawn independently from a probability density $p(X_i|\Theta) = \sum_{h=1}^k \alpha_h p_h(X_i|\theta_h)$, where $\Theta = \{\theta_h\}_{h=1}^k$, k is the number of mixture components, p_h is the probability density function of the h^{th} mixture component with parameters θ_h, and α_h are the component mixing coefficients such that $\alpha_h \geq 0$ and $\sum_{h=1}^k \alpha_h = 1$. In mixture model estimation, each point X_i is assumed to be generated from only one underlying mixture component. Let Z be a $n \times k$ boolean matrix such that Z_{ij} is 1 if the j^{th} component density was selected to generate X_i, and 0 otherwise. Let z_i be a hidden random variable corresponding to the index of the 1 in each row Z_i: every z_i is therefore a multinomial random variable, since it can take one of k discrete values. Since the Z matrix is unknown, the optimum parameters Θ of the mixture model can be obtained using the well-known iterative *Expectation Maximization (EM)* algorithm [12]. The probability value $p(z_i = h|X_i,\Theta)$ after convergence of the EM algorithm gives the probability of the point X_i being generated from the h^{th} mixture component. Using these probabilities, mixture models are often used to generate a partitional clustering of the data, where the points estimated to be most probably generated from the h^{th} mixture model component are considered to constitute the h^{th} partition.

In order to use the mixture model to get overlapping clustering, where a point can deterministically belong to multiple clusters, one can choose a threshold value λ such that X_i belongs to the h^{th} partition if $p(z_i = h|X_i,\Theta) > \lambda$. Such a thresholding technique can enable X_i to belong to multiple clusters. However, there are two problems with this method. One is that the choice of the parameter λ, which is difficult to learn given only X. Secondly, this is not a natural generative model for overlapping clustering. In the mixture model, the underlying model assumption is that a point is generated from only one mixture component, and $p(z_i = h|X_i,\Theta)$ simply gives the probability of X_i being generated from the h^{th} mixture component. However, an overlapping clustering model should generate X_i by simultaneously activating multiple mixture components. We describe one such model in the next section.

3.2 Proposed Overlapping Clustering Model

The overlapping clustering model that we present here is a generalization of the SBK model described in Section 2. The SBK model minimizes the squared loss between X and MA, and their proposed algorithms is not applicable for estimating the optimal M and A corresponding to other loss functions. In MOC, we generalize the SBK model to work with a broad class of probability distributions, instead of just Gaussians, and propose an alternate minimization algorithm for the general model.

The most important difference between MOC and the mixture model is that we remove the multinomial constraint on the matrix Z, so that it can now be an arbitrary boolean matrix. To distinguish from the constrained matrix Z, we denote this unconstrained boolean matrix as the membership matrix M. Every point X_i now has a corresponding k-dimensional boolean membership vector M_i: the h^{th} component M_i^h of this membership vector is a Bernoulli random variable indicating whether X_i belongs to the h^{th} cluster. So, a membership vector M_i with multiple 1's directly encodes the fact that the point X_i belongs to multiple clusters.

Let us now consider the probability of generating the observed data points in MOC. A is the activity matrix of this model, where A_h^j represents the activity of cluster h while generating the j^{th} feature of the data. The probability of generating all the data points is

$$p(X|\Theta) = p(X|M,A) = \prod_{i,j} p(X_i^j|M_i,A^j) \quad (1)$$

where $\Theta = \{M,A\}$ are the parameters of p, and X_i^j's are conditionally independent given M_i and A^j. In MOC, we assume p to be the density function of any regular exponential family distribution, and also assume that the expectation parameter corresponding to X_i is of the form M_iA, so that $E[X_i] = M_iA$. In other words, using

vector notation, we assume that each X_i is generated from an exponential family density whose mean M_iA is determined by taking the sum of the activity levels of the components that contribute to the generation of X_i, i.e., M_i^h is 1 for the active components.

Using the above assumptions and the bijection between regular exponential distributions and regular Bregman divergences [2], the conditional density can be represented as:

$$p(X_i^j|M_i,A^j) \propto \exp\{-d_\phi(X_i^j,M_iA^j)\} \quad (2)$$

where d_ϕ is the Bregman divergence corresponding to the chosen exponential density p. For example, if p is the Poisson density, d_ϕ is the I-divergence; if p is the Gaussian density, d_ϕ is the squared Euclidean distance [2].

Similar to the SBK model, the overlapping clustering model tries to optimize the following joint distribution of X, M and A:

$$\begin{aligned} p(X,M,A) &= p(M,A)p(X|M,A) = p(M)p(A)p(X|M,A) \\ &= \left(\prod_{i,h} p(M_i^h)\right)\left(\prod_{h,j} p(A_h^j)\right)\left(\prod_{i,j} p(X_i^j|M_i,A^j)\right). \end{aligned}$$

Making similar model assumptions as in Section 2, we assume that M and A are independent of each other apriori and A is distributed uniformly over a sufficiently large compact set, implying that $p(M,A) = p(M)p(A) \propto p(M)$. Then, maximizing the log-likelihood of the joint distribution gives

$$\begin{aligned} \max_{M,A} \log p(X,M,A) &\equiv \max_{M,A} \left[\sum_{i,h} \log p(M_i^h) - \sum_{i,j} d_\phi(X_i^j,M_iA^j)\right] \\ &\equiv \min_{M,A} \left[\sum_{i,j} d_\phi(X_{ij},(MA)_{ij}) - \sum_{i,h} \log \alpha_{ih}\right]. \end{aligned}$$

where $\alpha_{ih} = p(M_i^h)$ is the (Bernoulli) prior probability of the i-th point having a membership M_{ih} to the h-th cluster.

4. ALGORITHMS AND ANALYSIS

In this section, we propose and analyze algorithms for estimating the overlapping clustering model given an observation matrix X. In particular, from a given observation matrix X, we want to estimate the prior matrix α, the membership matrix M and the activity matrix A so as to maximize $p(M,A,X)$, the joint distribution of (X,M,A). The key idea behind the estimation is an alternating minimization technique that alternates between updating α, M and A.

4.1 Updating α

The prior matrix α can be directly calculated from the current estimate of M. If π_h denotes the prior probability of any point belonging to cluster h, then, for a particular point i, we have $\alpha_{ih} = \pi_h^{M_i^h}(1-\pi_h)^{1-M_i^h}$. Since π_h is the probability of a Bernoulli random variable, and the Bernoulli distribution is a member of the exponential family, the maximum likelihood estimate is just the sample mean of the sufficient statistic [2]. Since the sufficient statistic for Bernoulli is just the indicator of the event, the maximum likelihood estimate of the prior π_h of cluster h is just $\pi_h = \frac{1}{n}\sum_i \mathbb{1}_{\{M_i^h=1\}}$. Thus, one can compute the prior matrix α using these update equations.

4.2 Updating M

In the main alternating minimization technique, for a given X,A, the update for M has to minimize

$$\sum_{i,j} d_\phi(X_{ij},(MA)_{ij}).$$

Since M is a binary matrix, this is integer optimization problem and there is no known polynomial time algorithm to exactly solve the problem. The explicit enumeration method involves evaluating all 2^k possibilities for every data point, which can be prohibitive for even moderate values of k. So, we investigate simple techniques of updating M so that the loss function is minimized.

There can be two ways of coming up with an algorithm for updating M. The first one is to consider a real relaxation of the problem and allow M to take real values in $[0,1]$. For particular choices of the Bregman divergence, specific algorithms can be devised to solve the real relaxed version of the problem. For example, when the Bregman divergence is the squared loss, the corresponding problem is just the bounded least squares (BLS) problem given by $\min_{M:0\le M_{ih}\le 1} \|X-MA\|^2$, for which there are well studied algorithms [6]. Now, from the real bounded matrix M, one can get the cluster membership by rounding M_{ih} values either by proper thresholding [23] or randomized rounding. If k_0 clusters get turned "on" for a particular data point, the SBK model performs an explicit 2^{k_0} search over the "on" clusters in order get improved results. Another alternative could be to keep M in its real relaxed version till the overall alternating minimization method has converged, and round it at the very end. The update equation of the priors π_h and α_{ih} has to be appropriately changed in this case.

Although the real relaxation approach seems simple enough for the squared loss case, it is not necessarily so for all Bregman divergences. In the general case, one may have to solve an optimization problem (not necessarily convex) with inequality constraints, before applying the heuristics outlined above. In order to avoid that, we outline a second approach that directly tries to solve the integer optimization problem without doing real relaxation.

We begin by making two observations regarding the problem of estimating M: (1) In a realistic setting, a data point is more likely to be in very few clusters rather than most of them; and (2) For each data point i, estimating M_i is a variant of the *subset sum problem* that uses a Bregman divergence to measure loss. Taking the first observation a step further, for a domain if it is well understood (or desirable) that each data point can belong to at most k_0 clusters, for some k_0 possibly significantly smaller than k, then it may be computationally feasible to perform an explicit search over all the possibilities: $\binom{k}{1}+\binom{k}{2}+\cdots+\binom{k}{k_0} \le \left(\frac{ek}{k_0}\right)^{k_0}$, where the last inequality holds if $k_0 \le k/2$. Note that for $k_0 = 1$, the overlapping clustering model essentially reduces to the regular mixture model. However, in general, such a brute-force search may only be feasible for very small value of k_0. Further, it is perhaps not easy to decide on such a k_0 apriori for a given problem. So, we focus on designing an efficient way of searching through the relevant possibilities using the second observation.

The subset sum problem is one of the hard knapsack problems [9] that tries to solve the following: Given a set of k natural numbers $a_1,\ldots,a_k$ and a target number x, find a subset S of the numbers such that $\sum_{a_h\in S} a_h = x$. In a more realistic setting, one works with a set of real numbers, and tries to find a subset such that the sum over the subset is the *closest* possible to x. In our case, we measure closeness using a Bregman divergence and we have multiple target numbers to which we want the sum to be close. In particular, then the problem is to find M_i^* such that

$$M_i^* = \operatorname*{argmin}_{M_i\in\{0,1\}^k} d_\phi(X_i,M_iA) = \operatorname*{argmin}_{M_i\in\{0,1\}^k} \sum_{j=1}^{m} d_\phi(X_{ij},\sum_{h=1}^{k} M_i^h A_h^j).$$

Thus, there are m target numbers $X_{i1},\ldots,X_{im}$, and for each target number X_{ij} the subset is to be chosen from $A_j^1,\ldots,A_j^k$. The total loss is the sum of the individual losses, and the problem is to find a single M_i that minimizes the total loss.

Using the inherent bias of natural overlapping problems to put each point in low number of clusters, and the similarities of our formulation to the subset sum problem, we propose the algorithm dynamicM (Algorithm 1). The algorithm is motivated by the Apriori class of algorithms in data mining and Shapley value computation in co-operative game theory [17]. It is important to note that no theoretical claim is being made regarding the optimality of dynamicM. The belief is that such an efficient algorithm will work well in practice, as the empirical evidence in Section 5 suggests.

Algorithm 1 dynamicM

Input: Row vector $[\mathbf{x}]_{1\times d}$, distance function d, activity matrix $[A]_{k\times d}$, initial guess $[\mathbf{m}_0]_{1\times k}$
Output: Boolean membership vector $[\mathbf{m}]_{1\times k}$ that gives a low value for $d(\mathbf{x}, mA)$
Method:
Initialize assignment vector $[\mathbf{m}]_{1\times k}$ to all zeros
{Separate search thread for each initial cluster turned "on"}
for $h = 1$ to k **do**
  Turn "on" only the h-th cluster, i.e., set $\mathbf{m}(h) = 1, \mathbf{m}(i) = 0$, if $i \neq h$
  Set the h-th thread $\mathbf{t}_h$ to be 'active'
  Compute objective function $\ell_h = d(\mathbf{x}, \mathbf{m}A)$
  {Run over all possible sizes (> 1) of clusters turned "on"}
  for $r = 2$ to k **do**
    if thread $\mathbf{t}_h$ is still 'active' **then**
      Set $\ell_h^{\text{old}} = \ell_h$
      From the rest $(k - r + 1)$ clusters, find best cluster to turn "on"
      if best cluster to turn "on" is p **then**
        Turn "on" the p-th cluster, i.e., $\mathbf{m}(p) = 1$
        Compute objective function $\ell_h \leftarrow d(\mathbf{x}, \mathbf{m}A)$
      if $\ell_h^{\text{old}} \leq \ell_h$ **then**
        Set $\ell_h = \ell_h^{\text{old}}$
        Set the h-th thread $\mathbf{t}_h$ to be 'inactive'
Set $\mathbf{m} = \mathbf{m}_0$, $\ell = d(\mathbf{x}, \mathbf{m}_0 A)$
Find the best $\mathbf{m}$ over all threads using $\ell_h, h = 1, \ldots, k$
If best $\mathbf{m}$ over threads is worse than $\mathbf{m}_0$, set $\mathbf{m} = \mathbf{m}_0$
Output $[\mathbf{m}]_{1\times k}$

The algorithm dynamicM starts with 1 cluster turned "on" and greedily looks for the next best cluster to turn "on" so as to minimize the loss function. If such a cluster is found, then it has 2 clusters turned "on". Then, it repeats the process with the 2 clusters turned "on". In general, if h clusters are turned "on", dynamicM considers turning each one of the remaining $(k-h)$ clusters "on", one at a time, and computes loss corresponding to the membership vector with $(h+1)$ clusters turned "on". If, at any stage, turning "on" each one of the remaining $(k-h)$ clusters increases the loss function, the search process is terminated. Otherwise, it picks the best $(h+1)^{th}$ cluster to turn "on", and repeats the search for the next best on the remaining $(k-h-1)$ clusters.

Such a procedure will of course depend on the order in which clusters are considered to be turned "on". In particular, the choice of the first cluster to be turned "on" will partly determine which other clusters will get turned "on". The permutation dependency of the problem is somewhat similar in flavor to that of pay-off computation in a co-operative game. If h players are already in co-operation, the value-add of the $(h+1)^{th}$ partner will depend on the permutation following which the first h were chosen. In order to design a fair pay-off strategy, one computes the average value-add of a player, better known as Shapley value, over all permutations of forming co-operations [17].

Then, in theory, dynamicM should consider each all possible permutations, keep turning clusters "on" following each permutation to figure out the lowest loss achieved along that particular permutation, and finally compute the best membership vector among all permutations. Clearly, such an approach would be infeasible in practice. Instead, dynamicM starts with k threads, one corresponding to each one of the k clusters turned "on". Then, in each thread, it performs the search outlined above for adding the next "on" cluster, till no such clusters are found, or all of them have been turned "on". The search is similar in flavor to the Apriori algorithms, or, dynamic programming algorithms in general, where an optimal substructure property is assumed to hold so that the search for the best membership vector with $(h+1)$ clusters turned "on" starts from that with h clusters turned "on". Effectively, dynamicM searches over k permutations, each starting with a different cluster turned "on". The other entries of the permutation are obtained greedily on the fly. Since dynamicM runs k threads to achieve partial permutation independence, the best membership vector over all the threads is selected at the end. The algorithm has a worst case running time of $O(k^3)$ and is capable of running with any distance function.

4.3 Updating A

We now focus on updating the activity matrix A. Since there are no restrictions on A as such, the update step is simpler than that for M. Note that the only constraint that such an update needs to satisfy is that MA stays in the domain of ϕ. We give exact updates for particular choices of Bregman divergences: the squared loss and the I-divergence, since we use only these in section 5.

In case of the square loss, since the domain of ϕ is $\mathbb{R}$, the problem $\min_A \|X - MA\|^2$ is just the standard least squares problem that can be exactly solved by $A = M^\dagger X$, where $M^\dagger$ is the pseudo-inverse of M, and is equal to $(M^T M)^{-1} M^T$ in case $M^T M$ is invertible. In case of I-divergence or un-normalized relative entropy, the problem

$$\min_A d_I(X, MA) = \min_A \sum_{i,j} \left(X_{ij} \log \frac{X_{ij}}{(MA)_{ij}} - X_{ij} + (MA)_{ij} \right) , \quad (3)$$

has been studied as a non-negative matrix factorization technique [19]. The optimal update for A for given X, M multiplicative and is given by

$$A_h^j = A_h^j \frac{\sum_i M_i^h X_i^j / (MA)_i^j}{\sum_i M_i^h} \quad (4)$$

In order to prevent a division by 0, it makes sense to use $\max((MA)_i^j, \varepsilon)$ and $\max(\sum_i M_i^h, \varepsilon)$ as the denominators for some small constant $\varepsilon > 0$. With the above updates, the respective loss functions are provably non-increasing. In the case of a general Bregman divergence, the update steps need not necessarily be as simple and will be investigated as a future work.

5. EXPERIMENTS

This section describes the details of our experiments that demonstrate the superior performance of MOC on real-world data sets, compared to the thresholded mixture model.

5.1 Methodology

We run experiments on three types of datasets: synthetic data, movie recommendation data, and text documents. For the high-dimensional movie and text data, we create subsets from the original datasets, which have the characteristics of having a small number of points compared to the dimensionality of the space. The purpose of performing experiments on these subsets is to scale down the sizes of the datasets for computational reasons but at the same time not scale down the difficulty of the tasks, since clustering a small number of points in a high-dimensional space is a comparatively difficult task.

Synthetic data: In [23], apart from demonstrating their approach on gene microarray data and evaluating on standard biology databases,

Segal et al. also showed results on microarray-like synthetic data with a clear ground truth since the biology databases are generally believed to be lacking in coverage. The synthetic data was generated by sampling points from the overlapping clustering model and subsequently adding noise. We used a similar technique to create three synthetic datasets of different sizes: (1) *small-synthetic*: a dataset with $n = 75$, $d = 30$ and $k = 10$; (2) *medium-synthetic*: a dataset with $n = 200$, $d = 50$ and $k = 30$; and (3) *large-synthetic*: a dataset with $n = 1000$, $d = 150$ and $k = 30$. For the synthetic datasets we used squared Euclidean distance as the cluster distortion measure in the overlapping clustering algorithm, since Gaussian densities were used to generate the noise-free datasets.

Movie Recommendation data: The EachMovie[1] dataset has 5-point user ratings for the 74,424 movies in the collection. The corresponding movie genre information is extracted from the Internet Movie Database (IMDB)[2] collection. If each genre is considered as a separate category or cluster, then this dataset also has naturally overlapping clusters since many movies are annotated in IMDB as belonging to multiple genres, e.g., Aliens belongs to 3 genre categories: action, horror and science fiction. We created 2 subsets from the EachMovie dataset: (1) *movie-taa*: 300 movies from the 3 genres – thriller, action and adventure; and (2) *movie-afc*: 300 movies from the 3 genres – animation, family, and comedy. We clustered the movies based on the user recommendations to rediscover genres, based on the belief that similarity in recommendation profiles of movies gives an indication about whether they are in related genres. For this domain we use I-divergence with Laplace smoothing as the cluster distortion measure.

Text data: Experiments were also run on 3 text datasets derived from the *20-Newsgroups* collection[3], which has 20,000 documents from 20 Usenet newsgroups. We processed the original newsgroup articles to recover the multiple newsgroup labels on each message posting. From the full dataset, a subset was created having 100 postings in each of the 20 newsgroups, from which the following three data subsets were created with varying levels of overlap in the topics: (1) *news-similar-3*; (2) *news-related-3*; and (3) *news-different-3*. Details of these datasets are outlined in [3]. The vector-space model of each data subset was created using standard text pre-processing methods [13], and each data subset has 300 points in high-dimensional space (> 1000 words). In this case, I-divergence was again used as the Bregman divergence for overlapping clustering, with suitable Laplace smoothing.

We used an experimental methodology similar to the one used to demonstrate the effectiveness of the SBK model [23]. For each dataset, we initialized the overlapping clustering by running k-means clustering, where the additive inverse of the corresponding Bregman divergence was used as the similarity measure and the number of clusters was set by the number of underlying categories in the dataset. The resulting clustering was used to initialize our overlapping clustering algorithm.

To evaluate the clustering results, precision, recall, and F-measure were calculated over pairs of points. For each pair of points that share at least one cluster in the overlapping clustering results, these measures try to estimate whether the prediction of this pair as being in the same cluster was correct with respect to the underlying true categories in the data. Precision is calculated as the fraction of pairs correctly put in the same cluster, recall is the fraction of actual pairs that were identified, and F-measure is the harmonic mean of precision and recall.

[1]http://research.compaq.com/SRC/eachmovie

[2]http://www.imdb.com

[3]http://www.ai.mit.edu/people/jrennie/20Newsgroups

5.2 Results

Table 1 presents the results of MOC versus the standard mixture model for the datasets described in Section 5.1. Each reported result is an average over ten trials. For the synthetic data sets, we compared our approach to thresholded Gaussian mixture models; for the text and movie data sets, the baselines were thresholded multinomial mixture models. Table 1 shows that for all domains, even though the thresholded mixture model has slightly better precision in most cases, it has significantly worse recall: therefore MOC consistently outperforms the thresholded mixture model in terms of overall F-measure, by a large margin in most cases. Table 1 also shows that the performance of MOC improves empirically as the ratio of the data set size to the number of processes increases.

Table 2 compares the performance of using the `dynamicM` algorithm versus the bounded least squares (BLS) algorithm followed by local search, in the *M* estimation step in MOC. BLS/search gets better results on precision, which is expected since BLS is the optimal solution for the real relaxation of the *M* estimation problem for the Gaussian model. However `dynamicM` outperforms BLS/search on the overall F-measure. Moreover, BLS is only applicable for squared Euclidean distance, whereas `dynamicM` can be applied for *M* estimation with any distance function.

Detailed inspection of the results revealed that MOC gets overlapping clusterings that are closer to the ground truths for the text and the movie data. For example, for *movie-afc*, the average number of clusters a movie is assigned to is 1.19, whereas MOC clustering has an average of 1.13 clusters per movie. The thresholded mixture model got posterior probability values very close to 0 or 1, as is very common in mixture model estimation for high-dimensional data: as a result there was almost no cluster overlap for various choices of the threshold value, and points were assigned to 1.00 clusters on an average in the thresholded mixture models. MOC was also able to recover the correct underlying multiple genres in many cases, e.g., the movie "Toy Story" in the *movie-afc* dataset belongs to all the three genres of animation, family and comedy in this dataset, and MOC correctly put it in all 3 clusters.

The main purpose of the experiments in this section is to illustrate that the overlapping clustering model can be generalized to work for exponential models beyond Gaussians. We have not provided results on the biological datasets in this section due lack of space. However, note that if we run our algorithm on the biological data using BLS/search and a Gaussian model, then we will get exactly the same results as the SBK model [23].

6. RELATED WORK

Possibility theory, developed in the fuzzy logic community, allows an object to "belong" to multiple sets in the sense of having high membership values to more than one set [5]. In particular, unlike probabilities, the sum of membership values may be more than one [22]. One of the earlier works on overlapping clustering techniques with the possibility of not clustering all points was presented in [20]. Most recent work in overlapping clustering has been primarily driven by the needs of microarray analysis. Several methods for obtaining overlapping gene clusters, including gene shaving [16] and mean square residue bi-clustering [8] have been proposed. Before the PRM based SBK model was proposed, one of the most notable efforts was the the plaid model [18], wherein the gene-expression matrix was modeled as a superposition of several layers of plaids (subsets of genes and conditions).

Bregman divergences were conceived and have been extensively studied in the convex optimization community [7]. Over the past few years, they have been successfully applied to a variety of ma-

Data	F-measure		Precision		Recall	
	MOC	Mixture	MOC	Mixture	MOC	Mixture
small-synthetic	**0.64** $\pm$ 0.12	0.36 $\pm$ 0.08	**0.83** $\pm$ 0.07	0.80 $\pm$ 0.07	**0.53** $\pm$ 0.14	0.24 $\pm$ 0.07
medium-synthetic	**0.71** $\pm$ 0.06	0.24 $\pm$ 0.01	**0.73** $\pm$ 0.05	0.60 $\pm$ 0.03	**0.70** $\pm$ 0.09	0.15 $\pm$ 0.01
large-synthetic	**0.87** $\pm$ 0.04	0.33 $\pm$ 0.01	0.85 $\pm$ 0.06	**0.87** $\pm$ 0.04	**0.89** $\pm$ 0.05	0.20 $\pm$ 0.01
movie-taa	**0.62** $\pm$ 0.03	0.50 $\pm$ 0.04	0.55 $\pm$ 0.01	**0.56** $\pm$ 0.01	**0.71** $\pm$ 0.07	0.46 $\pm$ 0.08
movie-afc	**0.76** $\pm$ 0.03	0.61 $\pm$ 0.07	0.80 $\pm$ 0.01	**0.81** $\pm$ 0.02	**0.72** $\pm$ 0.06	0.50 $\pm$ 0.09
news-different-3	**0.45** $\pm$ 0.01	0.41 $\pm$ 0.05	0.34 $\pm$ 0.01	**0.40** $\pm$ 0.05	**0.68** $\pm$ 0.05	0.41 $\pm$ 0.06
news-related-3	**0.54** $\pm$ 0.02	0.39 $\pm$ 0.02	0.42 $\pm$ 0.01	**0.44** $\pm$ 0.02	**0.76** $\pm$ 0.08	0.35 $\pm$ 0.01
news-similar-3	**0.35** $\pm$ 0.02	0.28 $\pm$ 0.01	0.23 $\pm$ 0.01	**0.24** $\pm$ 0.01	**0.69** $\pm$ 0.06	0.34 $\pm$ 0.01

Table 1: Comparison of results of MOC and thresholded mixture models on all datasets

Data	F-measure		Precision		Recall	
	dynamicM	BLS/search	dynamicM	BLS/search	dynamicM	BLS/search
small-synthetic	**0.64** $\pm$ 0.12	0.55 $\pm$ 0.20	0.83 $\pm$ 0.07	**0.98** $\pm$ 0.03	**0.52** $\pm$ 0.14	0.41 $\pm$ 0.19
medium-synthetic	**0.71** $\pm$ 0.06	0.65 $\pm$ 0.05	0.73 $\pm$ 0.05	**0.91** $\pm$ 0.06	**0.70** $\pm$ 0.09	0.51 $\pm$ 0.06
large-synthetic	**0.87** $\pm$ 0.04	**0.87** $\pm$ 0.02	0.85 $\pm$ 0.06	**0.92** $\pm$ 0.02	**0.89** $\pm$ 0.05	0.83 $\pm$ 0.04

Table 2: Results: `dynamicM` **vs Bounded Least Squares (with search) for synthetic data**

chine learning issues, for example to unify seemingly disparate concepts of boosting and logistic regression [11]. More recently, they have been studied in the context of clustering [2].

Our formulation has some similarities to generalized linear models (GLMs) [21, 10]. However, there are a few very important differences. In GLMs [21], a multidimensional regression problem of the form $d_\phi(Y, f(BZ))$ is solved where Z is the (known) input variable, Y is the (known) response and f is the so-called canonical link function derived from ϕ. The problem is to find B and can be solved using iteratively re-weighted least squares (IRLS) in the general case. Extension to the case where both B and Z are unknown and one alternates between updating B and Z has been studied by Collins et al. [10] while extending PCA to the exponential families. Although several extensions [15] of the basic GLM model to matrix factorization have been studied, except for the well known instance of non-negative matrix factorization (NMF) using I-divergence [19], all formulations use the canonical link function and hence are different our formulation. Moreover, our model constraints M to be a binary matrix, which is never a standard constraint in GLMs.

7. CONCLUSIONS

In contrast to traditional partitional clustering, overlapping clustering allows items to belong to multiple clusters. In several important applications in bioinformatics, text management, and other areas, overlapping clustering provides a more natural way to discover interesting and useful classes in data. This paper has introduced a broad generative model for overlapping clustering, MOC, based on generalizing the SBK model presented in [23]. It has also provided a generic alternating minimization algorithm for efficiently and effectively fitting this model to empirical data. Finally, we have presented experimental results on both artificial data and real newsgroup and movie data, which is more general and effective than an alternative "naive" method based on thresholding the results of a traditional mixture model.

A few issues regarding practical applicability of MOC needs further investigation. It maybe often desirable to use different exponential family models for different subsets of features. MOC allows such modeling in theory, as long as the total divergence is a convex combination of the individual ones. Further, MOC can potentially benefit from semi-supervision [3] as well as be extended to a co-clustering framework [1].

Acknowledgements: The research was supported in part by NSF grants IIS 0325116, IIS 0307792, and an IBM PhD fellowship.

8. REFERENCES

[1] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. Modha. A generalized maximum entropy approach to Bregman co-clustering and matrix approximation. In *KDD*, 2004.

[2] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman divergences. In *SDM*, 2004.

[3] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *KDD*, 2004.

[4] A. Battle, E. Segal, and D. Koller. Probabilistic discovery of overlapping cellular processes and their regulation using gene expression data. In *RECOMB*, 2004.

[5] J. C. Bezdek and S. K. Pal *Fuzzy Models for Pattern Recognition* . IEEE Press, 1992.

[6] A. Bjorck. *Numerical Methods for Least Squares Problems*. Society for Industrial & Applied Math (SIAM), 1996.

[7] Y. Censor and S. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, 1998.

[8] Y. Cheng and G. M. Church. Biclustering of expression data. In *ISMB*, 2000.

[9] V. Chvátal. Hard knapsack problems. *Operations Research*, 28(6):1402–1412, 1980.

[10] M. Collins, S. Dasgupta, and R. Schapire. A generalization of principal component analysis to the exponential family. In *NIPS*, 2001.

[11] M. Collins, R. E. Schapire, and Y. Singer. Logistic regression, AdaBoost and Bregman distances. In *COLT*, 2000.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[13] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42:143–175, 2001.

[14] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI*, 1999.

[15] G. Gordon. Generalized^2 linear^2 models. In *NIPS*, 2001.

[16] T. Hastie, R. Tibshirani, M. B. Eisen, A. Alizadeh, R. Levy, L. Staudt, W. C. Chan, D. Botstein, and P. Brown. Gene shaving as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, 2000.

[17] J. Kleinberg, C. H. Papadimitriou, and P. Raghavan. On the value of private information. In *Proc. 8th Conf. on Theoretical Aspects of Rationality and Knowledge*, 2001.

[18] L. Lazzeroni and A. B. Owen. Plaid models for gene expression data. *Statistica Sinica*, 12(1):61–86, 2002.

[19] D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2001.

[20] W. T. McCormick, P. J. Schweitzer, and T. W. White. Problem decomposition and data reorganization by a clustering technique. *Operations Research*, 20:993–1009, 1972.

[21] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman & Hall/CRC, 1989.

[22] M. Sahami, M. Hearst, and E. Saund. Applying the Multiple Cause Mixture Model to Text Categorization. In *ICML*, 1996.

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