Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining
Chicago, IL, pp. 532--537, August 2005.

Model-based Overlapping Clustering

Arindam Banerjee Chase Krumpelman
Joydeep Ghosh

Dept. of Electrical and Computer Engineering
University of Texas at Austin
Austin, TX 78705, USA

Sugato Basu Raymond J. Mooney

Dept. of Computer Sciences
University of Texas at Austin
Austin, TX 78705, USA

ABSTRACT

While the vast majority of clustering algorithms are partitional, many real world datasets have inherently overlapping clusters. The recent explosion of analysis on biological datasets, which are frequently overlapping, has led to new clustering models that allow hard assignment of data points to multiple clusters. One particularly appealing model was proposed by Segal et al. [33] in the context of probabilistic relational models (PRMs) applied to the analysis of gene microarray data. In this paper, we start with the basic approach of Segal et al. and provide an alternative interpretation of the model as a generalization of mixture models, which makes it easily interpretable. 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, e.g., KL-divergence, Itakura-Saito distance, I-divergence. The general model is applicable to several domains, including high-dimensional sparse domains, such as text and recommender systems. We additionally offer several algorithmic modifications that improve both the performance and applicability of the model. We demonstrate the effectiveness of our algorithm through experiments 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

Keywords

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

1. INTRODUCTION

Almost all clustering methods assume that each item must be assigned to exactly one cluster and are hence partitional. However, in a variety of important applications, *overlapping clustering*, wherein some items are allowed to be members of two or more discovered clusters, is more appropriate. For example, in biology, genes have more than one function by coding for proteins that participate in multiple metabolic pathways; therefore, when clustering microarray gene expression data, it is appropriate to assign genes to multiple, overlapping clusters [33, 4]. In the popular *20-Newsgroups* benchmark dataset used in text classification and clustering [24], a fair number of the original articles were actually cross-posted to multiple newsgroups; the data was subsequently manipulated to produce disjoint categories. Ideally, a clustering algorithm applied to this data would allow articles to be assigned to multiple newsgroups and would rediscover the original cross-posted articles. In the popular *EachMovie* dataset used to test recommender systems [30], 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 and improve an approach to overlapping clustering introduced by Segal et al. [33], hereafter referred to as the SBK model. The original method was presented as a specialization of a Probabilistic Relational Model (PRM) [18] and was specifically designed for clustering gene expression data. We present an alternative (and we believe simpler) view of their basic approach as a straightforward 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.

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 referenced 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.

The main contributions of the paper can be summarized as:

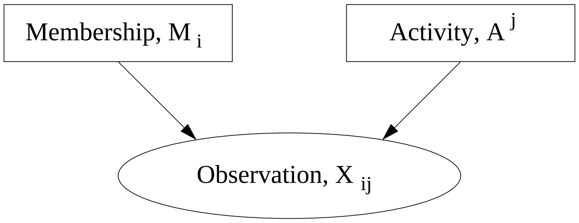

Figure 1: Basic graphical model for overlapping clustering

1. We show that the basic SBK model [33] for overlapping clustering can be (more simply) understood as an extension of the mixture modeling with Gaussian density functions, rather than a simplification of PRMs.

2. We extend the basic SBK model to work with any regular exponential family. Using a connection between exponential families and Bregman divergences [2], we show that the basic computational problem is that of matrix factorization using Bregman divergences to measure loss.

3. We outline an alternating minimization algorithm for the general model that monotonically improves the objective function for overlapping models for any regular exponential family distribution.

4. We present empirical evidence that the proposed overlapping clustering model works better than some alternative approaches to overlapping clustering.

A brief word on notation: $\mathbb{R}^d$ denotes the d-dimensional real vector space; p denotes a probability density function while other lowercase letters like k denote scalars; uppercase letters like 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_{ij} or X_i^j.

2. BACKGROUND

In this section, we give a brief introduction to the PRM-based SBK model. *Probabilistic Relational Models* (PRMs) [18, 23] extend the basic concepts of Bayesian networks into a framework for representing and reasoning with probabilistic relationships between entities in a relational structure. PRMs provide a very general framework, allowing for the learning of graphical models of probabilistic dependencies from arbitrarily complex relational databases.

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 key modeling assumption is as follows: 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. From the model assumption, we have

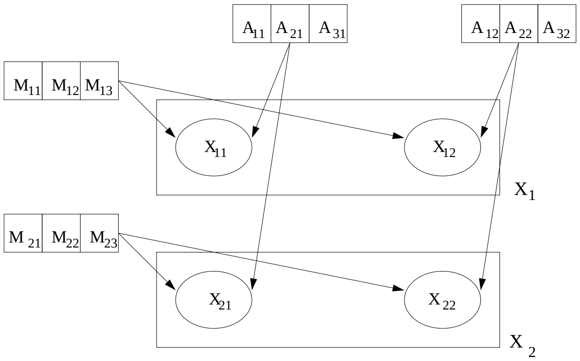

Figure 2: Instantiation of the PRM model to 2 data points (genes), 2 dimensions (experiments) and 3 clusters (processes).

$$p(X_i^j|M_i,A) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(X_i^j - M_iA^j)^2}{2\sigma^2}\right) , \qquad (1)$$

The SBK model assumes that M and A are independent apriori so that $P(M,A) = P(M)p(A)$ and that X_i^j's are conditionally independent given M_i and A^j. Further, M and A are assumed to be component-wise independent as well so that $P(M), P(A)$ can be decomposed into products over each component. All the above assumptions for the SBK model can be represented as a graphical model as shown in Figures 1 and 2. The joint distribution of X, M and A, that the SBK model tries to optimize is given by

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

Assuming that A_j^h are uniformly distributed over a sufficiently large compact set, and noting that the conditional distribution of X_i^j is Gaussian, considering the log-likelihood of the joint distribution, we have

$$\begin{aligned} \max_{M,A} \log p(X,M,A) &\equiv \max_{M,A} \left[\sum_{i,h} \log p(M_i^h) - \frac{1}{2\sigma^2}\sum_{i,j}(X_i^j - M_iA^j)^2\right] \\ &\equiv \min_{M,A} \left[\frac{1}{2\sigma^2}\|X - MA\|^2 - \log p(M)\right] \end{aligned}$$

To find the value of the hidden variables M, A, the SBK model uses an EM approach [15]. 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, ignoring the priors for the time-being. 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. In [33], estimating M and A for a given X proceeds as follows:

1. M is seeded with a first estimate of the clustering in the data, usually the output of a partitional clustering such as hierarchical or k-means run on the rows of X.

2. Next, the least-squares approximation of A for the given X and M is found as $A = M^{\dagger}X$, where $M^{\dagger}$ is the pseudo-inverse of M.

3. Using the A from step 2, the next approximation of M is found by relaxing the requirement that M be binary and solving a bounded least squares optimization for each gene in M. This effectively seeks a solution $\hat{M}_i = [0,1]^k$ for each row such that $||X_i - \hat{M}_i A||^2$ is minimized.

4. A binary solution M is then recovered from the real-valued solution $\hat{M}$ found in step 3 by thresholding. Since thresholding potentially moves the solution away from optimal, a local search is performed over every possible 0-flip of the post-threshold 1's to find the $M_i = \{0,1\}^k$ that minimizes $||X_i - M_i A||^2$.

5. Using the new M calculated in step 4, steps 2-4 are repeated until $||X - MA||^2$ is less than the desired convergence criteria.

The next section describes how the overlapping clustering model we propose generalizes the PRM-based SBK model. We also provide a simple interpretation of our model as a modification to the standard mixture modeling using exponential family distributions, that has been widely used for generative modeling of data.

3. THE MODEL

In this section, we present a quick review of basic mixture modeling, and outline a simplistic way of getting overlaps from the resulting soft-clustering. Then, we propose our model for overlapping clustering, hereafter referred to as MOC, as a generalization of the SBK model, and study the fixed point equations of the proposed model.

3.1 Basic Mixture Model

Given set of n data points, each point being a vector in $\mathbb{R}^d$, let them be represented by a $n \times d$ observation matrix X, such that row X_i denotes the i^{th} data point and X_{ij} represents its j^{th} feature. 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 $\sum_{h=1}^{k} \alpha_h = 1$. To sample a point following the density of this mixture model, first a component density function p_h is chosen with a probability α_h and then a point is sampled from $\mathbb{R}^d$ following p_h.

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. In mixture model estimation, since each point X_i is assumed to be generated from only one underlying mixture component, every row Z_i is a k-dimensional boolean vector constrained to have 1 in only one column and 0 everywhere else. Let z_i be a 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. If the matrix Z is known, one can directly estimate the parameters Θ of the most likely model explaining the data by maximizing the *complete* log-likelihood of the observed data, given by

$$\ln p(X, Z|\Theta) = \sum_{i=1}^{n} \ln(\alpha_{z_i} p_{z_i}(X_i|\theta_{z_i}))$$

However, the Z matrix is typically unknown: the optimum parameters Θ of the log-likelihood function with unknown Z, called the *incomplete* log-likelihood function, can be obtained using the well-known iterative *Expectation Maximization (EM)* algorithm [15].

3.2 Overlapping Clustering with Mixture Model

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. 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.

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 partition $\mathcal{X}_h$ 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 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.

3.3 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. The membership vector M_i for the point X_i effectively encodes 2^k configurations, starting from $[00\ldots0]$, indicating that X_i does not belong to any cluster, to $[11\ldots1]$, indicating that X_i belongs to all k clusters. So, a 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, such that 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

$$p(X|\Theta) = p(X|M, A) = \prod_{i,j} p(X_i^j|M_i, A^j) \qquad (2)$$

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. For example, if p represents a Gaussian density, then its mean would be the sum of the activity levels of the components for which the membership variable M_i^h of the point X_i has a value 1.

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

$$p(X_i^j|M_i, A^j) \propto \exp\{-d_\phi(X_i^j, M_iA^j)\} \qquad (3)$$

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.

3.4 Fixed Point Equations

We now present the fixed point equations of the overlapping clustering model that are satisfied for any Bregman divergence. The equations specify the connection between X, M, A at a fixed point of the model. It further suggests a general gradient descent update technique that we revisit later in Section 4. For notational convenience, let $\phi(X)$ on its own denote $\sum_{i,j}\phi(X_{ij})$ and $X \circ Y$ denote the matrix dot product $\text{tr}(X^TY) = \sum_{i,j} X_{ij}Y_{ij}$.

LEMMA 1. *For any Bregman divergence d_ϕ and any matrix X, the optimal values of M and A that minimize $d_\phi(X, MA)$ must satisfy the fixed point equations*

$$\begin{aligned} M^T\left[(X-MA) \circ \phi''(MA)\right] &= 0 \qquad (4) \\ \left[(X-MA)\circ\phi''(MA)\right]A^T &= 0 . \qquad (5) \end{aligned}$$

Further, $X = MA$ is a sufficient condition for the corresponding M, A to be optimal.

PROOF. The objective function to be optimized is

$$d_\phi(X, MA) = \phi(X) - \phi(MA) - (X-MA)^T\phi'(MA) .$$

Taking gradient with respect to A and setting it to the zero matrix of size $p \times m$, we have

$$\begin{aligned} -M^T\phi'(MA) - (-M^T)\phi'(MA) - M^T\left[(X-MA)\circ\phi''(MA)\right] &= 0 \\ \Rightarrow \qquad M^T\left[(X-MA)\circ\phi''(MA)\right] &= 0 . \end{aligned}$$

An exactly similar calculation with respect to M, with a zero matrix of size $n \times p$, gives

$$\left[(X-MA)\circ\phi''(MA)\right]A^T = 0 .$$

Now, note that any M, A with $X = MA$ satisfies both the fixed point equation. The corresponding loss function

$$d_\phi(X, MA) = d_\phi(X, X) = 0 ,$$

which is the global minimum for the objective function. Hence an exact factorization of X as MA is sufficient for M, A to be globally optimal. □

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 probability 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

$$\alpha_{ih} = \pi_h^{M_i^h}(1-\pi_h)^{1-M_i^h} . \qquad (6)$$

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

$$\pi_h = \frac{1}{n}\sum_i \mathbb{1}_{\{M_i^h=1\}} . \qquad (7)$$

Thus, one can compute the prior matrix α from (6) and (7).

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

$$\min_{\substack{M\\0\leq M_{ih}\leq 1}} ||X-MA||^2\ ,$$

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 [33] or randomized rounding [31]. 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:

$$\binom{k}{1}+\binom{k}{2}+\cdots+\binom{k}{k_0}\leq\left(\frac{ek}{k_0}\right)^{k_0}\ ,$$

where the last inequality holds if $k_0 \leq k/2$. 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 [11] 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 targets to which we want the sum to be close[1]. 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 targets $X_{i1},\ldots,X_{im}$, and for each target 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 [34] and Shapley value computation in co-operative game theory [22, 14]. 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:
   For $h = 1,\ldots,k$, set $[\mathbf{m}_h]_{1\times k}, [\mathbf{w}_h]_{1\times k}$ as all zeros
   Set $[\mathbf{t}]_{1\times k}$ as all ones
   for $h = 1$ to $k$ do
      $\mathbf{w}_h[h] \leftarrow 1$
      $\mathbf{m}_h[h] \leftarrow 1$
      $\ell_h \leftarrow d(\mathbf{x}, \mathbf{m}_h A)$
   for $r = 2$ to $k$ do
      for $h = 1$ to $k$ do
         if $\mathbf{t}_h = 1$ then
            $\ell_h^{\text{old}} \leftarrow \ell_h$
            for $p = 0$ to $(k-1)$ do
               if $(\mathbf{m}_h \vee \mathbf{w}_p \neq \mathbf{m}_h) \cap (d(\mathbf{x}, (\mathbf{m}_h + \mathbf{w}_p)A) < \ell_h)$ then
                  $\mathbf{m}_h \leftarrow \mathbf{m}_h + \mathbf{w}_p$
                  $\ell_h \leftarrow d(\mathbf{x}, \mathbf{m}_h A)$
            if $\ell_h^{\text{old}} = \ell_h$ then
               $\mathbf{t}_h = 0$
   $\mathbf{m} = \mathbf{m}_0$, $\ell = d(\mathbf{x}, \mathbf{m}_0 A)$
   for $h = 1$ to $k$ do
      if $\ell_h < \ell$ then
         $\mathbf{m} \leftarrow \mathbf{m}_h$
         $\ell \leftarrow \ell_h$
   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

[1] The problem is different from the so-called multiple subset sum problem [8].

forming co-operations [22, 14].

Then, in theory, `dynamicM` should consider each one of the $k!$ permutations[2], 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 significantly 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. First, 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. Then, we outline how the update can be done in case of a general Bregman divergence.

In case of the square loss, since the domain of ϕ is $\mathbb{R}$, the problem

$$\min_A \|X - MA\|^2 \tag{8}$$

is just the standard least squares problem that can be exactly solved by

$$A = M^\dagger X \tag{9}$$

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

$$\min_A d_I(X, MA) = \min_A \sum_{i,j} \left(X_{ij} \log \frac{X_{ij}}{(MA)_{ij}} - X_{ij} + (MA)_{ij} \right) , \tag{10}$$

has been studied as a non-negative matrix factorization technique [7, 26]. The optimal update for A for given X, M is multiplicative and is given by

$$A_h^j = A_h^j \frac{\sum_i M_i^h X_i^j / (MA)_i^j}{\sum_i M_i^h} \tag{11}$$

In order to prevent a divide 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 our experiments, we focus on only these two loss functions. In case of a general Bregman divergence, the update steps need not necessarily be as simple. In general, a gradient descent update can be derived using the fixed point equation (4) in Lemma 1. For a learning rate of η, the gradient descent update for A is given by

$$A^{\text{new}} \leftarrow A - \eta\, M^T \left[(X - MA) \circ \phi''(MA) \right] . \tag{12}$$

As in many gradient descent techniques, an appropriate choice of η involves a line search along the gradient direction at every iteration. Note that the simple I-divergence updates in (11) are derived from auxiliary function based methods. Existence of efficient updates based on auxiliary functions for the general case 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 Datasets

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. Clustering a small number of points in a high-dimensional space is a comparatively difficult task, as observed by clustering researchers [16]. 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.

5.1.1 Synthetic data

In [33], Segal et al. demonstrated their approach on gene microarray data and evaluated on standard biology databases. Since these biology databases are generally believed to be lacking in coverage, we elected to create microarray-like synthetic data with a clear ground truth. The synthetic data is generated by sampling points from the MOC model and subsequently adding noise.

To generate n points from MOC, where each point has a dimensionality d and the maximum number of processes it can belong to is k, we first generate a $n \times k$ binary membership matrix M from a Rayleigh distribution using rejection sampling. For each point, we first sample a value from a Rayleigh distribution [32] with a mean of 2. The actual number of processes p for the point is obtained by adding 1 to the sample value, so that the mean number of processes to which a point is assigned is effectively 3. Note that this additive shift assigns each point to at least 1 process since the original Rayleigh distribution has a range $[0, \infty)$, and we also truncate process values of $p > k$ to k. This makes the synthetic data closer to a biological model of gene microarray data, where the average number of processes a gene belongs to has been empirically observed to be close to 3 [33]. The final membership vector for the point is obtained by selecting p processes uniformly at random from the total possible set of k processes and turning on the membership values for those processes, the rest being set to 0. The membership vectors for all n points defines the overall membership matrix M.

We next generate a $k \times d$ activation matrix A, where every point is sampled from a Gaussian N (0,1) distribution. We form the observation X as MA and corrupt it with additive Gaussian noise N (0,0.5): the noise makes the task of recovery of M and A by performing the decomposition on X non-trivial. Three different synthetic datasets of different sizes were generated:

[2] Since the permutations decide clusters to turn "on", certain configurations repeat. A simple check for repeating configurations can bring computations down to 2^k, as one would expect.

- *small-synthetic*: a small dataset with $n = 75$, $d = 30$ and $k = 10$;
- *medium-synthetic*: a medium-sized dataset with $n = 200$, $d = 50$ and $k = 30$;
- *large-synthetic*: a large 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.

5.1.2 Movie Recommendation data

The EachMovie dataset has user ratings for every movie in the collection: users give ratings on a scale of 1-5, with 1 indicating extreme dislike and 5 indicating strong approval. There are 74,424 users in this dataset, but the mean and median number of users voting on any movie are 1732 and 379 respectively. As a result, if each movie in this dataset is represented as a vector of ratings over all the users, the vector is high-dimensional but typically very sparse.

For every movie in the EachMovie dataset, the corresponding genre information is extracted from the Internet Movie Database (IMDB) 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:

- *movie-taa*: 300 movies from the 3 genres – thriller, action and adventure;
- *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, which has been shown to work well on the movie recommendation domain [1].

5.1.3 Text data

Experiments were also run on 3 text datasets derived from the *20-Newsgroups* collection[3], which have the characteristics of being high-dimensional and sparse in the vector-space model. This collection has messages harvested from 20 different Usenet newsgroups, 1000 messages from each newsgroup. This dataset is popular among practitioners for evaluating text clustering or classification algorithms — it has each message annotated by one newsgroup, creating a non-overlapping categorization of messages by newsgroup membership. However, the original dataset had overlapping newsgroup categories — many messages were cross-posted to multiple newsgroups, e.g., multiple messages discussing the David Koresh/FBI standoff were cross-posted to `talk.politics.guns`, `talk.politics.misc` and `alt.atheism` newsgroups. The multiple newsgroup labels on the messages were artificially removed and replaced by one label; so, interestingly, the *20-Newsgroups* dataset had natural category overlaps, but was artificially converted into a dataset with non-overlapping categories. We parsed 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 datasets were created:

- *news-similar-3*: consists of 300 messages posted to 3 reduced newsgroups on similar topics (`comp.graphics`, `comp-.os.ms-windows`, and `comp.windows.x`), which had significant overlap between clusters due to cross-posting;
- *news-related-3*: consists of 300 messages posted to 3 reduced newsgroups on related topics (`talk.politics.misc`, `talk.politics.guns`, and `talk.politics.mideast`);
- *news-different-3*: consists of 300 messages posted to 3 reduced newsgroups that cover different topics (`alt.atheism`, `rec.sport.baseball`, `sci.space`).

The vector-space model of *news-similar-3* has 300 points in 1864 dimensions, *news-related-3* has 300 points in 3225 dimensions, while *news-different-3* had 300 points in 3251 dimensions. All the datasets were pre-processed by stop-word removal and removal of very high-frequency and low-frequency words, following the methodology of Dhillon et al. [17]. The raw counts of the remaining words were used in the vector-space model, and in this case too I-divergence was used as the Bregman divergence for overlapping clustering, with suitable Laplace smoothing.

5.2 Methodology

We used an experimental methodology similar to the one used to demonstrate the effectiveness of the SBK model [33]. 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:

$$\text{Precision} = \frac{\text{Number of Correctly Identified Linked Pairs}}{\text{Number of Identified Linked Pairs}}$$

$$\text{Recall} = \frac{\text{Number of Correctly Identified Linked Pairs}}{\text{Number of True Linked Pairs}}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision + Recall}}$$

5.3 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

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

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

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

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

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

MOC consistently outperforms the thresholded mixture model in terms of overall F-measure, by a large margin in most cases.

Figure 3 plots the improvements of MOC compared to the thresholded mixture model on the synthetic data, which 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, as shown in Figure 4. Moreover, BLS is only applicable for Gaussian models, whereas `dynamicM` can be applied for M estimation with any regular exponential model, by using the corresponding Bregman divergence to estimate the loss of approximating X by MA.

Figure 5 shows normalized reconstruction error, F-measure, precision, and recall for a run on the large synthetic data set, where the normalized reconstruction error is defined to be $||X - MA||^2/nd$. This graph demonstrates evidence validating the central assumption of the model: finding the MA decomposition that minimizes reconstruction error corresponds to finding a good estimate of the true cluster memberships.

Detailed inspection at the results revealed that MOC gets overlapping clustering that is 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. In the text domain, *news-related-3* has each article posted to 1.21 clusters on an average, and MOC assigns every posting to an mean number of 1.16 clusters. In both these cases, 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. For example, the movie "Toy Story" in the *movie-afc* dataset belongs to all the three genres of animation, fam-

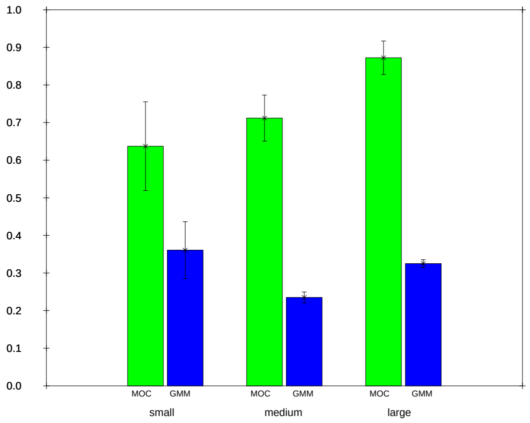

Figure 3: Average F-measure of the proposed model of overlapping clustering (MOC) and the thresholded Gaussian mixture model (GMM) on the synthetic datasets.

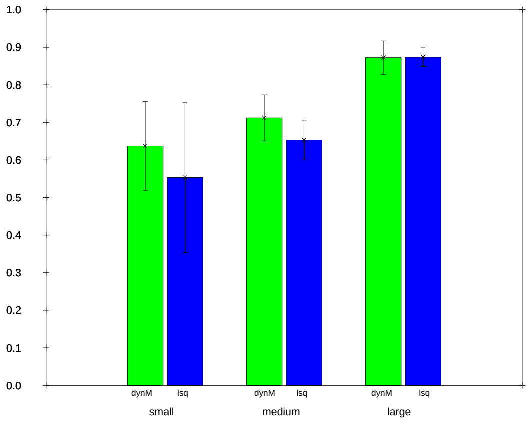

Figure 4: Comparison of the performances of dynamicM (dynM) and bounded least squares followed by search (lsq) on the synthetic data sets.

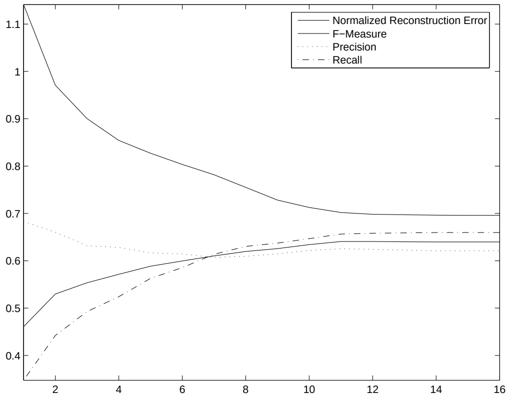

Figure 5: Plots of F-measure and Normalized Reconstruction Error vs. Iteration for a run on the large synthetic data. Note that decreasing error corresponds to increasing F-measure.

ily and comedy in this dataset, and MOC correctly put it in all 3 clusters. Similarly in the newsgroup dataset, message ID "76129" (which has a discussion on the topics of Israel, Judaism and Islam) is cross-posted to 2 newsgroups (`talk.politics.mideast` and `talk.politics.misc`), and MOC correctly put it in 2 clusters out of the possible 3.

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. The prototypical clustering algorithm in this community is fuzzy *c*-means [5], which is qualitatively very similar to a soft *k*-means algorithm obtained by applying EM to a mixture of isotropic Gaussians model. Moreover, assigning an object to multiple clusters using fuzzy *c*-means is again very similar to applying a threshold to the posterior probability $p(h|x)$ obtained through soft *k*-means.

In classification, there are several applications where an object may belong to multiple classes or categories. Typically this is achieved by assigning that object to all classes for which the corresponding (estimate of) *aposteriori* class probability is greater than a threshold, rather that choosing only the class with the highest *aposteriori* probability. For example, when classifying documents from the Reuters data set version 3 using k-nearest neighbor, a relatively high value of k=45 was chosen in [35]. A document was assigned to every class for which the weighted sum of the neighbors belonging to that class exceeded an empirically determined threshold. Note that the weighted sum is proportional to a local estimate of the corresponding *aposteriori* probability, with the weights determining the effective nature of the Parzen window that is used.

One of the earlier works on overlapping clustering techniques with the possibility of not clustering all points was presented in [28]. The more recent interest is due to the fact that overlapping clusters occur naturally in microarray data. Researchers soon realized that bi-clustering or co-clustering, i.e., simultaneous clustering of rows and columns, was suitable for such data sets since only certain groups of genes are co-expressed given a corresponding subset of conditions[27]. Several methods for obtaining overlapping gene clusters, including gene shaving [20] and mean square residue bi-clustering [10] have been proposed. Before the PRM based SBK model was proposed, one of the most notable effort in adapting bi-clustering to overlapped clustering was through the plaid model [25], wherein the gene-expression matrix was modeled as a superposition of several layers of plaids (subsets of genes and conditions). An element of the matrix can belong to multiple plaids while another may not belong to any plaid. The algorithm proceeds recursively by finding the most prominent plaid, removing it from the matrix, and then applying the plaid finding method to the residual.

Bregman divergences were conceived and have been extensively studied in the convex optimization community [9]. Over the past few years, they have been successfully applied to a variety of machine learning issues, for example to unify seemingly disparate concepts of boosting and logistic regression [13]. More recently, they have been studied in the context of clustering [2].

Our formulation has some similarities to but a few very important differences with a large class of models studied in the context of generalized linear models (GLMs) [29, 12, 19, 21]. In GLMs [29], 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 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. [12] while extending PCA to the exponential families. Although several extensions [19] of the basic GLM model to matrix factorization have been studied, expect for the well known instance of non-negative matrix factorization (NMF) using I-divergence [26, 7], all formulations use the canonical link function and hence cannot provide solutions to our problem. 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 [33]. 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 demonstrate the generality and effectiveness of our approach. In particular, we have shown that the approach produces more accurate overlapping clusters 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].

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 *Proc. of Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, 2004.

[2] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman divergences. In *Proc. of the 4th SIAM Intl. Conf. on Data Mining (SDM-04)*, 2004.

[3] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proc. of 10th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD-2004)*, pages 59–68, 2004.

[4] A. Battle, E. Segal, and D. Koller. Probabilistic discovery of overlapping cellular processes and their regulation using gene expression data. In *Proc. of 8th Intl. Conf. on Research in Computational Molecular Biology (RECOMB-2004)*, 2004.

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

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

[7] C. Byrne and Y. Censor. Proximity function minimization using multiple Bregman projections, with applications to split feasibility and Kullback-Leibler distance minimization. *Annals of Operations Research*, 105:77–98, 2001.

[8] A. Caprara, H. Kellerer, and U. Pferschy. The multiple subset sum problem. *SIAM Journal on Optimization*, 11(2):308–319, 2000.

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

[10] Y. Cheng and G. M. Church. Biclustering of expression data. In *Proc. 8th Intl. Conf. on Intelligent Systems for Molecular Biology (ICMB)*, pages 93–103, 2000.

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

[12] M. Collins, S. Dasgupta, and R. Schapire. A generalization of principal component analysis to the exponential family. In *Proc. of 14th Annual Conf. on Neural Information Processing Systems (NIPS)*, 2001.

[13] M. Collins, R. E. Schapire, and Y. Singer. Logistic regression, AdaBoost and Bregman distances. In *Proc. of 13th Annual Conf. on Computational Learing Theory (COLT)*, pages 158–169, 2000.

[14] V. Conitzer and T. Sandholm. Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In *Proc. of 19th Natnl. Conf. on Artificial Intelligence*, pages 219–225, 2004.

[15] 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.

[16] I. S. Dhillon and Y. Guan. Information theoretic clustering of sparse co-occurrence data. In *Proc. of 3rd IEEE Intl. Conf. on Data Mining (ICDM-03)*, pages 517–521, 2003.

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

[18] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proc. of 16th Intl. Joint Conf. on Artificial Intelligence (IJCAI-99)*, 1999.

[19] G. Gordon. Generalized^2 linear^2 models. In *Proc. of 14th Annual Conf. on Neural Information Processing Systems (NIPS)*, 2001.

[20] 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.

[21] J. Kivinen and M. K. Warmuth. Relative loss bounds for multidimensional regression problems. *Machine Learning*, 45:301–329, 2001.

[22] 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.

[23] D. Koller and A. Pfeffer. Probabilistic frame-based systems. In *Proc. of 15th Natl. Conf. on Artificial Intelligence (AAAI-98)*, pages 580–587, 1998.

[24] K. Lang. NewsWeeder: Learning to filter netnews. In *Proc. of 12th Intl. Conf. on Machine Learning (ICML-95)*, pages 331–339, San Francisco, CA, 1995.

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

[26] D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 13, pages 556–562, 2001.

[27] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE Trans. Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[28] 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.

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

[30] P. McJonese. Eachmovie collaborative filtering dataset. DEC Systems Research Center.

[31] P. Raghavan and C. D. Thompson. Randomized rounding. *Combinatorica*, 7:365–374, 1987.

[32] S. M. Ross. *Introduction to Probability Models*. Academic Press, 2000.

[33] E. Segal, A. Battle, and D. Koller. Decomposing gene expression into cellular processes. In *Proc. of the 8th Pacific Symposium on Biocomputing (PSB)*, 2003.

[34] R. Srikant and R. Agrawal. Mining generalized association rules. In *Proc. of the 21st Intl. Conf. on Very Large Databases (VLDB-95)*, pages 407–419, 1995.

[35] Y. Yang and X. Liu. A re-examination of text cateogrization methods. In *Proc. of 22nd Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1999.