Redefining Class Definitions using Constraint-Based Clustering

An Application to Remote Sensing of the Earth's Surface

Dan R. Preston
Tufts University
Department of Computer Science
161 College Ave, Medford, MA
dan.preston@tufts.edu

Carla E. Brodley
Tufts University
Department of Computer Science
161 College Ave, Medford, MA
brodley@cs.tufts.edu

Roni Khardon
Tufts University
Department of Computer Science
161 College Ave, Medford, MA
roni@cs.tufts.edu

Damien Sulla-Menashe
Boston University
Department of Geography and Environment
675 Commonwealth Ave, Boston, MA
dsm@bu.edu

Mark Friedl
Boston University
Department of Geography and Environment
675 Commonwealth Ave, Boston, MA
friedl@bu.edu

ABSTRACT

Two aspects are crucial when constructing any real world supervised classification task: the set of classes whose distinction might be useful for the domain expert, and the set of classifications that can actually be distinguished by the data. Often a set of labels is defined with some initial intuition but these are not the best match for the task. For example, labels have been assigned for land cover classification of the Earth but it has been suspected that these labels are not ideal and some classes may be best split into subclasses whereas others should be merged. This paper formalizes this problem using three ingredients: the existing class labels, the underlying separability in the data, and a special type of input from the domain expert. We require a domain expert to specify an $L \times L$ matrix of pairwise probabilistic constraints expressing their beliefs as to whether the L classes should be kept separate, merged, or split. This type of input is intuitive and easy for experts to supply. We then show that the problem can be solved by casting it as an instance of penalized probabilistic clustering (PPC). Our method, Class-Level PPC (CPPC) extends PPC showing how its time complexity can be reduced from $O(N^2)$ to $O(NL)$ for the problem of class re-definition. We further extend the algorithm by presenting a heuristic to measure adherence to constraints, and providing a criterion for determining the model complexity (number of classes) for constraint-based clustering. We demonstrate and evaluate CPPC on artificial data and on our motivating domain of land cover classification. For the latter, an evaluation by domain experts shows that the algorithm discovers novel class definitions that are better suited to land cover classification than the original set of labels.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
KDD'10, July 25–28, 2010, Washington, DC, USA.
Copyright 2010 ACM 978-1-4503-0055-110/07 ...$10.00.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms; Similarity Measures*

General Terms

Algorithms, Experimentation, Measurement

Keywords

KDD-Process, mining scientific data, constraint-based clustering, class discovery, remote sensing

1. INTRODUCTION

The first steps in the iterative process of KDD are to understand the domain and collect the relevant data [6]. In the case of supervised classification learning tasks, it is assumed that this step provides the classes of interest. For example, in the domain of classifying high resolution CT (HRCT) scans of the lung, one collects examples of scans of patients from the n pulmonary disease classes [1]. The next steps are to clean and transform the data, perform feature extraction/selection, and then apply the chosen data mining algorithm. In cases where the desired performance is not achieved, practitioners can revisit any of the steps in the process to improve either the quality of the training data, or change the data mining algorithm [6]. In this paper, we address the following data collection issue: what if the features cannot discriminate the classes of interest? The most obvious solution is to ask the domain expert for additional features. However, for some domains this is infeasible because the instruments that generated the data may not have the ability to provide new measurements. The second option, and the one addressed in this paper, is to reconsider the class definitions.

Before discussing our solution, we first examine how class definitions arise. For many datasets, they are provided by human experts as the categories or concepts that people find useful. For others one can apply clustering methods to automatically find the homogeneous groups in the data. Both approaches have drawbacks.

The categories and concepts that people find are useful may not be supported by the features (i.e., the features may be inadequate for making the class distinctions of interest). Applying clustering to find the homogeneous groups in the data may find a class structure that is not of use to the human. For example, applying clustering to the HRCT data may group two pulmonary diseases together in one cluster that have radically different treatments or, conversely, it may split one disease class into multiple clusters that have no meaning with respect to diagnosis or treatment of the disease.

In this paper, we present a method for redefining class definitions that leverages both the class definitions found useful by the human experts and the structure found in the data via clustering. Our approach is based on the observation that for supervised training data sets, significant effort went into defining the class labels of the training data, but that these distinctions may not be supported by the features. Our method, named Class-Level Penalized Probabilistic Clustering (CPPC), is designed to find the natural number of clusters in a data set, when given constraints in the form of class labels. We require a domain expert to specify an $L \times L$ matrix of pairwise probabilistic constraints, where L is the number of classes in the data. This matrix makes use of the idea that the expert will have different levels of confidence for each class definition, and likely preferences as to which class labels may be similar. Thus, each element of the matrix defines the expert's belief that a pair of classes should be kept separate or should be merged. Elements on the diagonal reflect the expert's opinion of whether a particular class is multimodal. Using the instances' class labels and the probabilities in the matrix provides us with pairwise instance-level constraints. In Section 2 we describe the definition and use of this matrix in more detail. Our framework for discovering the natural clustering of the data given this prior knowledge uses a method based primarily on the Penalized Probability Clustering (PPC) algorithm [15]. A straightforward implementation of the PPC algorithm using mean field approximation [10] results in $O(N^2)$ computational complexity due to the number of constraints (i.e., one constraint for each pair of instances), where N is the total number of instances. In CPPC, we reduce this time complexity to $O(NL)$ by taking advantage of the repetitions in the set of instance pairwise constraints that are induced from the class pairwise constraints.[1]

Previous work on constraint-based clustering has not explicitly addressed the issue of finding an optimal value for K, the number of clusters. We present a new method for calculating adherence to the constraints that can be applied in conjunction with any model selection criterion. The result is a heuristic criterion that maximizes the fit to the data and adherence to the constraints, while trying to minimize model complexity.

Our research was motivated by the domain of land cover classification of the Earth's surface using remotely sensed data. For land cover data sets, it is traditional to partition data into L land cover classes, where the specific classes depend on the end use (e.g., tracking global climate change). Most importantly in the context of this paper, the classes are defined based on the user requirements without considering what classes can be discriminated by the features generated from the remotely sensed spectral data. Before we began this research, we speculated that 1) large, complex classes such as agriculture may contain several distinguishable sub-classes and 2) geographic regions classified as one of the mixed classes, such as "mixed forest" should most likely be merged with either of its two subclasses: "deciduous broadleaf forest" or "evergreen broadleaf forest." In the later case, the spectral remotely sensed data of some sensors cannot distinguish well between these types of forest. In Section 4.2 we describe the results of our methods on a land cover data set for North America, in which we find areas of confusion where classes should be merged or where separation within a class provides additional useful information, such as corn and wheat clusters within the agriculture class.

The remainder of this paper is organized as follows. In Section 2, we review constraint-based clustering, and define the CPPC framework. In particular we describe how the constraints are generated from the class labels using the matrix of constraints provided by the domain expert. Building on the PPC model for incorporating constraints into the EM algorithm, we formulate a set of equations for reducing the complexity from $O(N^2)$ to $O(NL)$. In Section 3, we present a heuristic to measure adherence to constraints, and provide a criterion for determining the correct model complexity for constraint-based clustering algorithms. In Section 4, we first present results for a synthetic data set to examine different aspects of the algorithm, and the affects of changes in the choice of values for the parameters. We then present a detailed analysis of our results on the land cover data sets done by the geographers on our team. CPPC was able to discover a better feature-supported set of classes, which in many cases provided more useful class distinctions than the original labels. The new cluster definitions' accuracy is supported by subjective analyses of several map products, and other quantitative summary analyses.

2. CLASS-LEVEL PAIRWISE CONSTRAINT CLUSTERING

In this section we address how to use the prior knowledge encoded in the class labels to generate pair-wise constraints. Our approach provides an intuitive method for specifying constraints for the human expert. In particular they need only provide the constraints for each pair of *classes* rather than for each pair of instances – a process which we describe in more detail in Section 2.2. Before describing our approach, we first review the existing work in constraint-based clustering.

2.1 Constraint-Based Clustering

For domains in which one has knowledge as to which instances should and should not be clustered together, one can apply constraint-based clustering methods [4]. Constraint-based clustering was originally introduced by Wagstaff, et al [24] using a modification of K-means [16] that takes into account must-link constraints, where two points must be in the same cluster, and cannot-link constraints, where two points cannot be in the same cluster. These are "hard" constraints, and thus any clustering of the data points must satisfy the constraints. In essence, the algorithm reassigns examples to their nearest cluster at each iteration, but only if it does not violate any constraints. Because the EM algorithm often provides better results than K-means, due to its ability to model specific distributions, solutions have been introduced that address the use of "hard" constraints in the EM algorithm. In [20], both must-link and cannot-link constraints are added by forcing the expectation to zero if the constraints are not met. Hard constraints are not always available or desirable. Thus, several algorithms appear in the literature that present the ability to incorporate probabilistic (or "soft") constraints in some form [13, 25, 15]. In this approach, a soft clustering algorithm is applied and constraints are expressed as a prior probability that pairs of instances should (or should not) be assigned to the same cluster. These probabilistic constraints are combined with the probability of generating the data by the mixture model, to get a combined likelihood function. Clustering with pairwise constraints in this setting can be formulated as inference

[1] In general, $L \ll N$, as the number of classes is generally much smaller than the number of instances.

in a corresponding Bayesian network [13]. Due to the intractability of this construction for large datasets, Lu and Leen [15] applied a mean field approximation to produce the PPC algorithm. We describe PPC and our modifications in more detail in Section 2.3.

Constraint based clustering is related but distinct from semi supervised clustering [3] where the labels (cluster membership) of some of the examples are known in advance, and therefore the induced constraints are more explicit. Another closely related approach includes spectral clustering and other graph based clustering algorithms [23] where pairwise similarities between examples provide the only information available to induce the clustering. Recent work (see [12]) has shown that all three approaches can be recast into a single framework by defining a suitable kernel function for each approach capturing the constraints or similarities. Then the problem can be seen as optimizing some trace equation in the kernel space, and the kernel K-means algorithm can be used to find a local maximum of the objective function. Our work can be distinguished in that we optimize the likelihood function directly and develop an efficient algorithm for the optimization by using the structure of the constraints.

2.2 Specifying the Constraints

To incorporate prior knowledge into the clustering algorithm in the form of constraints, we need a method for assigning a constraint for each instance pair X and Y. In our approach, this constraint is a function of their labels ℓ_X and ℓ_Y. If the expert specifies one constraint for each pair of classes, then we can generate instance pair constraints by assigning the constraint C(ℓ_i, ℓ_j) to each pair of instances where one is a labeled as ℓ_i and the other as ℓ_j.

More specifically, given L original classes, the domain expert specifies an $L \times L$ matrix C, in which each element $C(\ell_i, \ell_j)$ represents the belief that instances in class ℓ_i should be clustered with those in class ℓ_j. In this work, we assume C is symmetric, leaving an extension to asymmetric constraints to future work. A negative value, from the interval $[-1, 0)$, indicates a belief that the two classes should not be clustered together (e.g., a value of $-.75$ means that the expert believes with .75 probability that ℓ_i should not be confused as ℓ_j). A value of 0 indicates a lack of prior knowledge, and a value from the interval $(0, 1]$ indicates the belief that two classes should be merged. Thus, each value in the matrix is a belief, and is negated depending on the type of constraint (positive for must-link and negative for cannot-link). The value of a diagonal element $C(\ell_i, \ell_i)$ represents the belief that instances from class ℓ_i should be kept together, where a low probability indicates that it's likely that the class may be multimodal and a high probability would force the class to adhere to its original definition. An example of the C-matrix is provided in Table 1.

It is important to understand what the values in the C matrix actually represent, and what the effect of these values will be. The belief probability is *not* the expert's opinion on how the clustering would perform if there were no constraints. In other words, the expert does not need to be concerned with his/her belief in how separable the classes may be, as this is reflected in the unsupervised aspect of the algorithm. Instead, the expert should define the values in terms of his/her preferences. If the expert believes that a class definition is too broad for any reason, he/she should define the value on the diagonal (e.g., $C(\ell_i, \ell_i)$ for class ℓ_i) as a negative value, such that it will be biased toward splitting the class. Or, if the expert believes that two classes ℓ_i and ℓ_j should be merged for any reason, $C(\ell_i, \ell_j)$ should be a positive value. For example, if treatment of pulmonary diseases ℓ_i and ℓ_j are the same then we may not care if we can discriminate these two diseases, and thus we would set $C(\ell_i, \ell_j) > 0$. Finally, if the expert's goal is to maintain as much of the original class definitions as allowed by the chosen model (e.g., Gaussian Mixture Model), the constraint values should be set to 1 on the diagonal, and -1 for all other entries.

2.3 Integrating Expert Information into EM

CPPC is based on the EM algorithm, but incorporates the expert knowledge in the E- and M-steps. In particular, the PPC algorithm [15] can be extended to make use of the C matrix, by noting that the constraint for a pair of instances X and Y is their labels' class-wise constraint value $C(\ell_X, \ell_Y)$. The complexity of the E-step is $O(N^2)$ when using PPC. In the next section, we show that one can reduce this to $O(NL)$ by taking advantage of the redundancy in the pairwise constraints. Note that in this section we assume the data comes from a mixture model of multivariate Gaussians [17], but our method can be applied to other mixture probability distributions. Further we assume that K is given, addressing how to find the optimal K in the following section. We begin with the initialization of the model.

Initializing the clusters: For non-constraint based EM, the clusters are typically initialized by selecting K points randomly as the cluster centers or by using K-means [16]. In our case we have class label constraints, which allow us to use the original class model to initialize the clusters. Each cluster θ_u consists of a prior probability P_u mean vector $\vec{\mu_u}$ and a covariance matrix $\vec{\Sigma_u}$. There are three cases, $K = L$, $K > L$, and $K < L$, where L is the number of original class labels and K is the number of clusters. When $K = L$ we can create one cluster for each class and assign instances to the clusters by their class labels. This is done by calculating L clusters $\theta_1, \ldots, \theta_L$ and calculating the prior P_u, the mean vector $\vec{\mu_u}$, and the covariance matrix $\vec{\Sigma_u}$ for θ_u, corresponding to the data with class label u.

In the situation where $K > L$ we begin by assigning each of the L classes to its own cluster $\theta_1, \ldots, \theta_L$, in the same way we defined the clusters for the $K = L$ situation. We then repeat the following three steps for $L+1, \ldots, K$: 1) we choose the cluster θ_u with the largest average variance (averaged over each feature); 2) we split the cluster θ_u by defining two new clusters $\theta_{u'}, \theta_{v'}$. The mean and covariance vectors, $\vec{\mu_{u'}}, \vec{\mu_{v'}}, \vec{\Sigma_{u'}}, \vec{\Sigma_{v'}}$, are defined by performing K-means on the instances that are members of cluster θ_u (where K = 2). The new priors P'_u, P'_v are defined by dividing the total number of instances in each cluster by the total number of instances overall; and 3) we remove θ_u. This method is similar to the Gaussian split method defined in [22].

When $K < L$ we initialize the clusters $\theta_1, \ldots, \theta_L$ with the original L class model definitions. We then repeat the following three steps until there are a total of K clusters: 1) We choose the two clusters θ_u and θ_v for which the Euclidean distance of the cluster means is the smallest; 2) we then form a new cluster $\theta_{u'}$ by merging θ_u and θ_v into a single cluster, and re-computing $P_{u'}$, $\vec{\mu_{u'}}$ and $\vec{\Sigma_{u'}}$; and 3) we remove both θ_u and θ_v.

Performing EM with Class Pairwise Constraints: Our CPPC algorithm is based on the constrained clustering algorithm of Lu and Leen [15] who defined a generative model capturing the constraints and clustering objective. In this model, an MRF-like equation defines the probabilities of cluster memberships. Given cluster membership, a Gaussian mixture model defines the probability of the data. The complete data log likelihood of this model is

$$\ln \mathcal{L} = \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik}\alpha_{ik} + \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\sum_{k=1}^{K} z_{ik}z_{jk}\lambda C(\ell_i, \ell_j) - \ln \Omega \quad (1)$$

where $z_{i,k} \in \{0, 1\}$ is 1 if example i is generated by cluster k, $\alpha_{i,k} = \ln(P_k P(\vec{y_i}|\theta_k))$, $\lambda > 0$ is a weight factor allowing us

to scale all entries in C, and Ω, the normalizing constant for the distribution is defined as:

$$\Omega = \sum_{Z \in \{0,1\}^{N \times K}} \left[\prod_{i=1}^{N} \prod_{k=1}^{K} P_k^{z_{ik}} \right] \times \left[\prod_{i=1}^{N} \prod_{j=1, j \neq i}^{N} e^{\sum_{k=1}^{K} z_{ik} z_{jk} \lambda C(\ell_i, \ell_j)} \right] \quad (2)$$

Inference for this probabilistic mode is intractable and therefore Lu and Leen [15] developed a variational EM (mean field) algorithm to learn the parameters. Defining $Pr(z_i) = \prod_k q_{ik}^{z_{ik}}$, such that $q_{ik} = E[z_{ik}]$, the E-step estimates q_{ik} using the following update formula to bias the clusters toward the original class structure

$$q_{ik} \sim P_k P(\vec{y_i}|\theta_k) \exp\left(2 \sum_{j=1, j \neq i} \lambda C(\ell_i, \ell_j) q_{jk} \right) \quad (3)$$

As noted in [15], the update formula in Equation 3 tends to converge after approximately twenty iterations, which was confirmed during our experiments. During the M-step of the PPC algorithm, we must calculate the new cluster parameters for each cluster θ_k:

$$\vec{\mu_k} = \frac{\sum_{i=1}^{N} \vec{y_i} q_{ik}}{\sum_{i=1}^{N} q_{ik}} \quad (4)$$

$$\vec{\Sigma_k} = \frac{\sum_{i=1}^{N} q_{ik} (\vec{y_i} - \vec{\mu_k})(\vec{y_i} - \vec{\mu_k})^T}{\sum_{i=1}^{N} q_{ik}} \quad (5)$$

Currently, there is no closed-form solution to find P_k. Thus, we must use a method for approximating the value. Lu and Leen [15] proposed a method for calculating P_k that requires an approximation of Ω and search over a discretized space of the probabilities. However, the discretized search is exponential in the number of clusters and not feasible in general. A more tractable alternative approximates Ω empirically but uses an optimization method to optimize P_k (we used conjugate gradients in our experiments). To approximate Ω we observe that (2) captures an expectation under a product of multinomials, and therefore one can approximate Ω empirically by sampling. However, this too is expensive due to the approximation of Ω which is required at every step. Instead, for large datasets, we propose to optimize the pseudo likelihood which essentially ignores the normalizing term Ω of the distribution, an approximation which has been widely used before in various models including constrained clustering [25]. This gives the update equation:

$$P_k = \frac{1}{N} \sum_{i=1}^{N} q_{ik}. \quad (6)$$

In our experiments the results of this approach are indistinguishable from the more expensive alternative.

Complexity Analysis and Algorithm Speedup: Using this model, the complexity of the update step in Equation 3 requires a calculation of the q_{ik} values for each clusters k and instance i. In addition, to calculate each q_{ik}, we need to sum over all instances $j = 1, \ldots, N$ where $j \neq i$. Thus, to compute the E-step, our algorithm would have complexity $O(KN^2)$. We next show that one can reduce this to $O(KNL)$ by taking advantage of the redundancy in the pairwise constraints.

The main observation is that the summations over all instances for all q_{ik} have many repeated values. This is because the label of the current instance ℓ_i remains constant, and only changes for the L possible labels of the other instance. Thus, we can perform a preprocessing step that is only $O(KNL)$,

$$S^*(\ell, k) = \sum_{i=1}^{N} \lambda C(\ell, \ell_i) q_{ik} \quad (7)$$

which is calculated in N steps over $L \times K$ possible values. Using this insight we can rewrite Equation 3 to be:

$$q_{ik} \sim P_k P(\vec{y_i}|\theta_k) \exp(2(S^*(\ell_i, k) - \lambda C(\ell_i, \ell_i) q_{ik})) \quad (8)$$

which is calculated in one step over $N \times K$ possible values. Our construction takes advantage of the repetitions in the set of induced pairwise constraints to perform a less expensive preprocessing step, allowing each step of the q_{ik} calculation to take constant time.

3. EVALUATION

Determining whether a clustering result is desirable is difficult. In unsupervised clustering, the compactness of clusters and the distance of cluster means can be used to determine whether a clustering was successful. This becomes increasingly challenging in constraint-based clustering, because one must also consider how well the clustering adheres to the constraints. We address this issue in this section, and in addition, use our evaluative metrics to determine an optimal number of clusters K. We begin by presenting a non-summary criterion to evaluate constraint adherence, followed by a review of clustering evaluation, and finally combining these two ideas to form a heuristic for determining model complexity.

3.1 Measuring Constraint Adherence

One non-summary method for measuring the fit between a class partition and clustering partition could be to look at a confusion matrix between the clusters and classes. In other words, each value (a, b) in the matrix would represent the frequency that instances appear in cluster a and have original class label b. This matrix can be useful for an expert to analyze if they wish to observe how instances were clustered together. Unfortunately, this can quickly become difficult to understand as the number of classes and/or clusters grows. Furthermore, there is no clear way to calculate a summary value that would have any statistically significant meaning, nor is there a way to compare among different numbers of clusters K (as the dimensions differ for each value of K).

To alleviate the problem of understanding the meaning of the confusion matrix, we develop a new "Separability" matrix S that will provide measurements in an easily understandable form and has the same dimensions for any value of K. We define S to be an $L \times L$ matrix, where $S(a, b) \in [-1, 1]$ with -1.0 denoting that instances from that pair of classes are never clustered together and a value of 1.0 denoting that all instances from that pair of classes are clustered together in a single cluster. To find the value of $S(a, b)$ we first define two frequency counts, ϕ_{ab} and φ_{ab}. ϕ_{ab} represents the frequency of unordered instance pairs $\{x_i, x_j\}$ with original class labels a, b that share the same cluster z_i:

$$\phi_{ab} = |\{x_i, x_j | \ell_i = a, \ell_j = b, z_i = z_j\}| \quad (9)$$

φ_{ab} is the frequency of instance pairs that appear in different clusters:

$$\varphi_{ab} = |\{x_i, x_j | \ell_i = a, \ell_j = b, z_i \neq z_j\}| \quad (10)$$

Note that ϕ_{aa} represents the number of times pairs of instances from the same class are clustered together and φ_{aa} represents the number of pairs of instances from the same class that are found in different clusters. Thus we define $S(a, b)$ as:

$$S(a, b) = \frac{\phi_{ab}}{\phi_{ab} + \varphi_{ab}} - \frac{\varphi_{ab}}{\phi_{ab} + \varphi_{ab}} \quad (11)$$

$S(a,b)$ represents how often instances with class labels a and b appear together. When $a = b$, this represents how often a class is kept together (positive values) or split apart (negative values). When $a \neq b$, $S(a,b)$ represents how often two classes are merged (positive values) or kept separate (negative values).

The domain expert may be interested in examining S, as each element in the matrix provides a granular perspective of how each original class was split or merged. Ultimately, it would be desirable to look at these values in aggregate to form a criterion that balances constraint adherence with traditional model evaluations (compactness and distance of cluster means). Thus, in the remainder of this section, we examine traditional methods for determining K and how to modify them using the constraint adherence criterion presented above.

3.2 Review of Clustering Evaluation

Identifying the correct value of K, the number of clusters in the data, has been a topic of significant research [2, 19]. Heuristic criteria consist of two terms: a model fit term and a complexity penalty term. The complexity penalty term is needed because as the number of clusters grows typically the model fit will increase. Indeed, for EM, the likelihood is maximized when each instance is assigned to its own cluster. A commonly used criterion is the Bayesian Information Criterion [19]:

$$BIC = N \log \frac{RSS}{N} + K \log N \tag{12}$$

where N is the number of instances, K is the number of clusters and RSS, the residual sum of squares, is the fit term and decreases as the number of clusters grows. Specifically, it is defined as

$$RSS = \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik}[1 - P(\vec{y_i}|\theta_k)]^2 \tag{13}$$

where $z_{ik} = 1$ if instance $\vec{y_i}$ has cluster label k after clustering. The second term in Equation 12, $K \log N$ is the complexity penalty term. There are many other criteria used to select K. Among the most popular are the Akaike Information Criterion [2], the Deviance Information Criterion [21], and the Hannan-Quinn Information Criterion [9]. Applications of these criteria choose a value for K at the "knee" of a curve where the y-axis is the criterion and the x-axis is the number of clusters K.

3.3 Determining Model Complexity

A simple extension of BIC would add a penalty term to the log likelihood of Equation 1. However, as discussed above calculating Ω and hence the log likelihood is computationally hard. In the following we propose an alternative measure that similarly captures a compromise between the likelihood of the Gaussian portion of the model and the penalty imposed by the constraints. We believe that this gives a better tool to evaluate the quality of the resulting clustering, as compared to using an approximation of Ω or simply using the pseudo likelihood.

To evaluate how closely a clustering result $\{z_i\}$ adheres to the constraints, C we create a measure $G(C, \{z_i\})$, based on the fraction of instances that are clustered together versus those that are split apart. This value can be used to compare the effectiveness of two clustering results, and additionally, used as the adherence term for our new criterion. To compute $G(C, \{z_i\})$, we first compute the $L \times L$ matrix S, defined in Section 3.1 from the clustering result $\{z_i\}$. We now have a fully defined S matrix in the same dimensions as the original C-matrix. To obtain a summary value that can be used to evaluate the full clustering result for a *symmetric* C matrix, one can use the square of the difference between C and S to evaluate how well the clustering adhered to the constraints:

$$G(C, \{z_i\}) = \sum_{a=1}^{L} \sum_{b=1}^{L} (C(a,b) - S(a,b))^2 \tag{14}$$

where Eq. 14 is minimized when it adheres to the constraints in C.

We can use $G(C, \{z_i\})$ to help select the correct model complexity – i.e., the number of clusters K. Here, we show its application in conjunction with the BIC criterion, but similar applications are straightforward for other criteria described in Section 3.2. We define the *constrained Bayesian Information Criterion* (cBIC) to be:

$$\text{cBIC} = (1-\lambda) N \log \frac{RSS}{N} + \lambda N \log \frac{G(C, \{z_i\})}{4L^2} + K \log N \tag{15}$$

where the term $G(C, \{z_i\})$ is normalized to $[0, 1]$ by dividing by $4L^2$ (the largest value that $G(C, \{z_i\})$ can take), and scaled by N, such that it grows at the same rate as the fit term $N \log \frac{RSS}{N}$. Furthermore, λ is the expert's probabilistic belief in the constraint matrix and it balances the unsupervised (BIC) and supervised ($G(C, \{z_i\})$) metrics that determine optimal settings for each. As with BIC, we wish to minimize cBIC.

To use this new criterion, we run the constraint-based EM algorithm for a range of K deemed reasonable by the domain expert. At this point we can show the expert a graph of cBIC, for which the x-axis are values of K and the y-axis are values of cBIC. Ideally our criterion would provide a minimum, but more realistically provides a range of acceptable values (where the curve flattens, see Figure 3 for the cBIC curve for the land cover data set) which the expert can then use to examine the clustering solutions.

4. EXPERIMENTAL EVALUATION

Before describing the results for the land cover domain, we present results for a synthetic data set. It is designed to explore three aspects of the algorithm: 1) how the constraints are used to make decisions between classes (i.e., when they are merged or split), 2) to analyze the efficacy of cBIC, and 3) to examine the sensitivity of the results to λ. Our motivating domain of geography is examined in detail in Section 4.2. Our methodology for each experiment is as follows. For each experiment where we examine BIC and cBIC, we run the experiment ten times and average the criterion over all runs.[2] Furthermore, when a clustering is examined in detail, we use the run with the criterion value nearest the average.

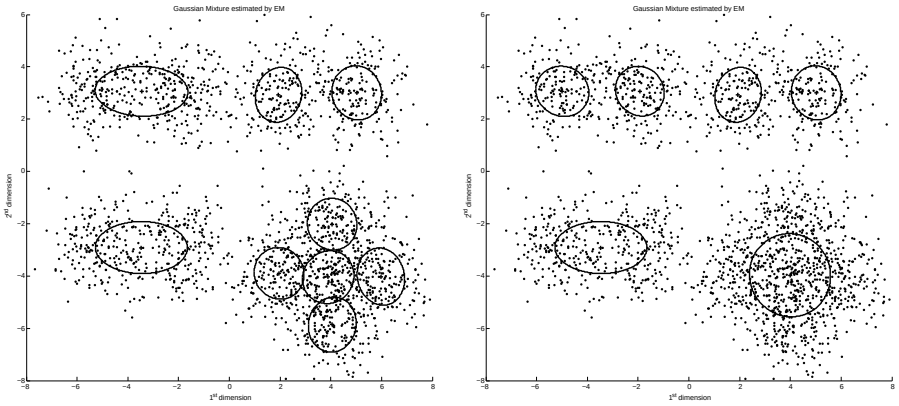
4.1 Synthetic Data Set

The synthetic data set examines four situations we may face in constraint-based clustering. Our data set is generated by eleven Gaussian distributions (each having the identity covariance matrix) and originally defined as nine classes. The constraint matrix is provided in Table 1 and we set $\lambda = 0.2$. We randomly generate 200 instances from each of the eleven distributions.

The original classification of the eleven Gaussian mixtures (each with two features) can be seen in Figure 1(a) as a two-dimensional plot (shown as the boldface ovals) and the resulting CPPC cluster definitions using Table 1 for $K = 6$, at which CBIC is minimized, are shown in Figure 1(b). Note that the values of -1.0 in Table 1 help in preventing each group of clusters from biasing the expectation for the others. The four potential situations are:

1. *One class generated by two Gaussians, seen in the top left of the plot in Figure 1(a), with Gaussian mean vectors* $[-5, 3]$ *and* $[-2, 3]$. This demonstrates a situation with weak constraints for

[2]Variance in clustering results can exist due to the random initialization when $K > L$.



(a) Original classification (b) CPPC clustering, using $K = 6$

Figure 1: Analysis for synthetic data set

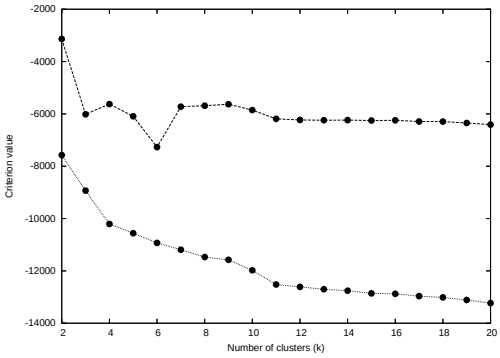


Figure 2: Comparison of cBIC (upper line) and BIC (lower line) for constraints in Table 1

class 1 (see Table 1), and the class is multimodal. Thus, the CPPC algorithm will not bias the expectation significantly, and splits this class into two clusters, as seen in Figure 1(b).

2. *Two classes generated by one Gaussian each, seen in the top right of the plot, with Gaussian mean vectors* $[5, 3]$ *and* $[2, 3]$. In this case, we are provided mid-range constraints: our belief that classes 2 and 3 will be confused is 0.5. The two classes remain as two clusters after the CPPC clustering, due to their separability in the feature space. Furthermore, because the constraints did not provide a strong enough penalty to cBIC, a larger K was chosen and thus the clusters did not merge. As we see later, increasing the confidence value λ in our C matrix will increase the penalty for the constraints, and thus choose a smaller K that will bias the expectation such that these two clusters are merged.

3. *One class (class* 4*) generated by two Gaussians, seen in the bottom left of the plot, with Gaussian mean vectors* $[-5, -3]$ *and* $[-2, -3]$. By providing a strong constraint of 1.0 in the C matrix for class 4, the expectation is heavily biased and thus keeps the class from being split, even though the two Gaussians are easily separable in feature space.

4. *Five classes generated by five Gaussians, seen in the bottom right of the plot, with Gaussian mean vectors* $[4, -2]$, $[4, -4]$, $[6, -4]$, $[2, -4]$, *and* $[4, -6]$. In the final case, we are provided with five Gaussians that are somewhat separable in the feature space, yet our domain expert has provided us with entries in the C matrix that claims each of these original class definitions are easily confusable

Table 1: Synthetic Data Set: C-matrix Constraints

	1	2	3	4	5	6	7	8	9
1	.1	-1	-1	-1	-1	-1	-1	-1	-1
2	-1	.5	.5	-1	-1	-1	-1	-1	-1
3	-1	.5	.5	-1	-1	-1	-1	-1	-1
4	-1	-1	-1	1	-1	-1	-1	-1	-1
5	-1	-1	-1	-1	.9	.9	.9	.9	.9
6	-1	-1	-1	-1	.9	.9	.9	.9	.9
7	-1	-1	-1	-1	.9	.9	.9	.9	.9
8	-1	-1	-1	-1	.9	.9	.9	.9	.9
9	-1	-1	-1	-1	.9	.9	.9	.9	.9

with one another, but not with class 1, 2, 3, and 4. Because of these strong constraints, the cBIC is heavily penalized and thus a lower K is chosen. It should be noted that if a larger K were chosen, then this situation would appear as five clusters, as the original clustering in Figure 1(a) shows. This is because the constraints are the same (all equal to 0.9). The important point to be made is that cBIC chooses a smaller K due to these high-probability constraints.

In Figure 2, we report the criterion values for different values of K.[3] For our experiment, we use $\lambda = 0.2$ and thus we notice that cBIC is minimized at $K = 6$. One can see the "knee" of the criterion for BIC around $K = 12$, which is approximately the number of Gaussian distributions (eleven) used to generate the data set. Furthermore, no error bars are shown in the figure due to minimal variance in the results.

Finally, we note that when one sets $\lambda = 0.75$, the cBIC curve indicates that K should be 5 (not shown). This is due to the second situation (see above) in which we had mid-range constraint probabilities. If $K = 5$ then the two classes in the second situation (top right of the plot in Figure 1(b)) are merged. This occurs because the confidence in the constraints provided in the C matrix are weighted more heavily due to a stronger value of λ. Choosing an appropriate value λ involves striking a balance between the unsupervised portion of the algorithm and the supervised portion. Thus, if one wishes to have a strong tendency toward the original labels, one would provide a strong λ (close to 1). Otherwise, if the analyst is less confident in their labels and/or constraint matrix C, a lower λ (closer to 0) should be chosen.

[3]Although we display cBIC and BIC on the same graph, we are not suggesting that we directly compare them as they are functions with different ranges.

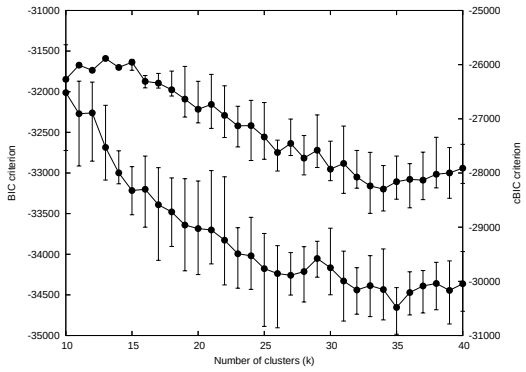


Figure 3: Land Cover data set: cBIC for CPPC for (upper line), and BIC for original EM (lower line).

4.2 Land Cover Data Set

In the domain of remote sensing, it is often important to produce a classified map of the Earth to make inferences on trends and properties of the earth's surface (for example, global climate change). These classifications include agriculture, forests, urban areas, water, etc., which can generally be identified by a set of features measured from advanced instruments. The IGBP class definitions [14] were developed by international community consensus through a process sponsored by the International Geosphere-Biosphere Program. The classification scheme is designed to capture the first order global variation in biome and land use properties. The set of seventeen IGBP classes has become the standard, and elaborate methods have been produced to create the most accurate representation of the Earth's surface [7].

Although widely used over the last decade, the IGBP scheme is now recognized to be limited in terms of its characterization of land cover. In particular, class definitions based on vegetation height and arbitrary per cover fractions that may or may not be identifiable from remote sensing are specific weaknesses. And on the other hand, the instruments used to generate the data have greatly improved. Because of this increased amount of information, there has been a resurgence in techniques to identify subclasses from these original classes [18]. Significant progress has been made in the field to extract the most amount of information from these features (i.e., pixels), such as feature selection using domain knowledge and spatial information [5], and reducing the problem to better representations (e.g., superpixels) [8].

The 2004 version of the V5 MCD12Q1 Land Cover product [7] was used to derive the training data for this exercise. The original class labels for the IGBP data set were obtained by expert labeling and the interpretation of Landsat TM imagery, and recently from Google Earth. We chose to examine only North America, which covers only 15 of the 17 IGBP classes. There are 7329 instances in the hand-labeled training data each described by 139 features.

In the following analysis, our goal is to obtain a better set of classes by leveraging the original IGBP class labels and additional information we have gained in the feature space from improved instruments and sensors. In cases where a single class is split into one or more separate classes, the constrained clustering presents distinctions that may be useful and ecologically important but were not considered when the legend was first developed. In the cases where classes are merged, the constrained clustering indicates that the distinction between the two classes are not present in the data and that either the class needs to be redefined or new features need to be considered to create an actual distinction.

In the following analysis, we perform CPPC and original EM clustering using our framework on the data to discover new class structure, given the expert defined 15×15 C matrix.[4] Furthermore, we are provided with an overall confidence value of $\lambda = 0.25$, which completes the set of inputs for our algorithm. Our experts determined the value of λ by acknowledging that their overall confidence in the C-matrix was not particularly high, due to significant uncertainty in the original class labels.

The methodology for our analysis of the land cover data set is as follows. We first needed to reduce the dimensionality of the data set, which was done by performing principle component analysis [11]. We selected $M = 19$ features, which covered 95% of the variance. Using a PCA is standard practice in remote sensing, and we adhere to this practice rather than examining other dimensionality reduction methods to be consistent with other published results for datasets created by the same sensor.

Figure 3 shows cBIC values for CPPC for $K = 2, \ldots, 40$, using the pseudo likelihood approximation for P_k. Results (not shown) for the more expensive method for estimating P_k, are indistinguishable from the ones shown here. To achieve accurate estimates of the cBIC we ran CPPC ten times for each K and averaged the value. One can see that the cBIC value is minimized at $K = 34$ for CPPC, and thus we provided our domain expert with the new classification of the data at that value. In addition, we needed to compare our framework with the original EM algorithm (i.e., without constraints). We performed K-means [16] to seed EM with initialization parameters, and ran EM until convergence. The BIC values for $K = 2$ to $K = 40$ can be seen in Figure 3, which represents the average BIC value of ten EM runs for each K.

In order to compare the two algorithms and the split and merge decisions each made, we chose to compare classifications using the same K. Thus, we decided to analyze $K = 34$ for both EM and CPPC, where cBIC is minimized in Figure 3.To produce a map from the class definitions resulting from CPPC or EM we first train a boosted C4.5 tree[5] and then apply the tree to the remainder of the pixels in North America.

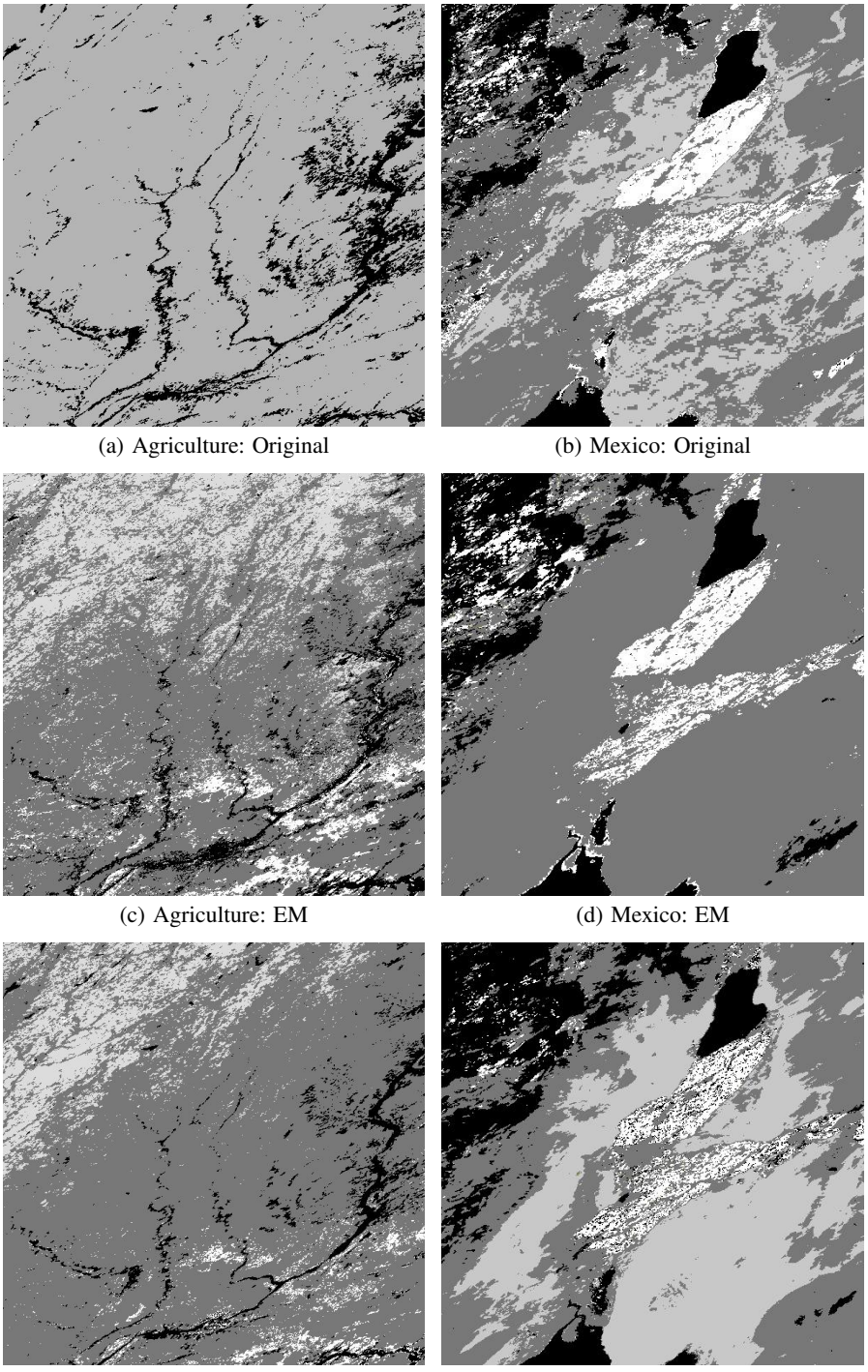
The geographers on our team spent several days comparing the map produced by CPPC to that produced by EM and to one produced with the original IGBP classification. *They were not informed which set of labels was the product of CPPC or EM.*[6] They concluded that the constrained clustering resulted in more realistic distinctions of land cover types than either the original class definitions or unconstrained EM. Figure 4 provides a detailed analysis of two map areas of the Earth.[7] The left side of the figure shows a case where CPPC and EM are able to split a class into meaningful subcategories. The right side of the figure shows a case where CPPC retains class definitions of the original classification when EM is not able to do so. Thus CPPC is able to mix existing labels, domain experts biases, and the geometry of the data to make meaningful class definitions.

[4]Due to space limitations, the constraint matrix is not provided here. It can be found at http://www.cs.tufts.edu/research/ml/projects/igbp/index.html.

[5]The geographers on our team are highly familiar with C4.5 and have code that takes the classified data to produce a global land cover map, a non-trivial process.

[6]They were highly familiar with the maps produced using the original labels, thus we did not include it in the "blind" comparison.

[7]Due to print requirements, we are limited to black and white maps that do not accurately represent class separation. A full color analysis can be found at http://www.cs.tufts.edu/research/ml/projects/igbp/index.html.



(a) Agriculture: Original (b) Mexico: Original

(c) Agriculture: EM (d) Mexico: EM

(e) Agriculture: CPPC (f) Mexico: CPPC

Figure 4: Maps using original IGBP classification (top), EM clustering (middle), and CPPC clustering (bottom). The left side (a, c and e) shows a map tile centered on the intensively farmed border of Wisconsin, Minnesota, Iowa, and Illinois. In each map on the left dark grey represents natural vegetation and black represents water. In the top map, light grey is agriculture (IGBP class 12). Comparing to the agriculture zone in the original map the EM and CPPC results show three agriculture variants. The medium grey is expected to be broadleaf crops (e.g., corn and soybeans). The light grey is expected to be cereal crops or a mix of cereals and broadleaf, and the white group is irrigated corn. This demonstrates a prime example of a class being split into different zones, representing differing crop types or irrigation status. The right side (b, d, f) shows the town of Mexicala, Mexico on the US border, with San Diego to the west. In each map on the right, black represents natural woody vegetation and water, white represents agriculture, light grey represents barren, and dark grey represents sparse vegetation, grassland and open shrubland. The agriculture in the center of the scene is heavily irrigated and occurs in the center of a desert. In contrast with CPPC, the EM clustering does not differentiate between sparsely vegetated/open shrubland and barren desert. This demonstrates the ability of CPPC to retain information from the original IGBP labels. In addition to this desirable outcome, the CPPC and EM clusterings identify a transition between the desert vegetation to grassland, which is not represented by the IGBP classification.

5. CONCLUSIONS

In this paper, we have presented a framework designed to discover the natural classification for a data set, given its prior class labels using constraint-based clustering. As inputs, CPPC requires a domain expert to provide a matrix C of confidence values between each pair of original class definitions, and an overall confidence value λ. CPPC is an approach to address the issue of existing classes that may be unsupported by their feature space. This is done by performing constraint-based clustering, based on EM using a redefined likelihood function using mean field approximation. In addition, we provided a framework to evaluate a class clustering model using cBIC, which also allows us to choose the appropriate model size. Using a synthetic data set, our experimental results demonstrated how decisions were made by the algorithm, the utility of the cBIC criterion, its differences from the original BIC criterion, and the sensitivity of the results to the confidence parameters. In addition, our motivating domain provided interesting examples of merging, splits, and information retainment.

For future work, it would be instructive to examine asymmetric constraints. These situations can arise when there is no cost to classifying one class as another, but not in the other direction. As an extreme example, consider two treatments, where one is not harmful (penicillin) and the other may be quite harmful for all cases except the one that it cures (chemotherapy). This idea of directionality in constraints only exists because of the ability to use existing class labels. Finally, our geographers are interested in using this tool in diagnosing the strength of the IGBP classification scheme. By observing the results of the constraints, one can better assess what sorts of actual divisions exist in the data and whether or not our desired class descriptions are feasible distinctions (i.e., can actually be distinguished by the features). In the past, an unsupervised clustering method could have attempted to provide some insight, but would be unable to maintain the required elements of the basic original class structure.

Acknowledgments

Mark Friedl and Damien Sulla-Menashe were supported by NASA cooperative agreement number NNX08AE61A. Carla Brodley, Dan Preston and Roni Khardon were supported by NSF IIS-0803409.

6. REFERENCES

[1] A. M. Aisen et al. Automated storage and retrieval of medical images to assist diagnosis: Implementation and preliminary assessment. *Radiology*, 228:265–270, July 2003.

[2] H. Akaike. A new look at the statistical identification model. *IEEE Trans. Auto Control*, AC-19:716–723, 1974.

[3] S. Basu, A. Banerjee, and R. J. Mooney. Semi-supervised clustering by seeding. In *ICML*, pages 27–34, 2002.

[4] S. Basu, I. Davidson, and K. Wagstaff. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC, 2008.

[5] C.-C. Chen and D. Landgrebe. A spectral feature design system for the hiris/modis era. *Geoscience and Remote Sensing, IEEE Transactions on*, 27(6):681–686, Nov 1989.

[6] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data-mining to knowledge discovery: An overview. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurasamy, editors, *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[7] M. Friedl et al. Global land cover mapping from MODIS: Algorithms and early results. *Remote Sensing of Environment*, 83:287–302, 2002.

[8] H. Ghassemian and D. Landgrebe. Object-oriented feature extraction method for image data compaction. *Control Systems Magazine, IEEE*, 8(3):42–48, Jun 1988.

[9] E. J. Hannan and B. G. Quinn. The determination of the order of an autoregression. *Journal of the Royal Statistical Society, Series B*, 41(2):190–195, 1979.

[10] T. S. Jaakkola. Tutorial on variational approximation methods. In *Advanced Mean Field Methods: Theory and Practice*, pages 129–159. MIT Press, 2000.

[11] I. T. Jolliffe. *Principal component analysis*. Springer Series in Statistics, 1986.

[12] B. Kulis, S. Basu, I. S. Dhillon, and R. J. Mooney. Semi-supervised graph clustering: a kernel approach. *Machine Learning*, 74(1):1–22, 2009.

[13] M. H. C. Law, E. Topchy, and A. K. Jain. Clustering with soft and group constraints. *Proc. Joint IAPR International Workshops on Structural, Syntactic, And Statistical Pattern Recognition*, pages 662–670, 2004.

[14] T. Loveland et al. Development of a global land cover characteristics database and IGBP DISCover from 1-km AVHRR data. *Remote Sensing of Environment*, 83:287–302, 2002.

[15] Z. Lu and T. K. Leen. Penalized probabilistic clustering. *Neural Comput.*, 19(6):1528–1567, 2007.

[16] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the Fifth Sym. on Math, Statistics, and Probability*, pages 281–297, 1967.

[17] G. J. McLachlan and K. E. Basford. *Mixture models. Inference and applications to clustering*. Statistics: Textbooks and Monographs, 1988.

[18] M. Pugh and A. Waxman. Classification of spectrally-similar land cover using multi-spectral neural image fusion and the fuzzy artmap neural classifier. In *IGARSS 2006*, pages 1808–1811, 31 2006-Aug. 4 2006.

[19] G. Schwartz. Estimating the dimension of a model. *The Annals of Statistics*, 5(2):461–464, 1978.

[20] N. Shental, A. Bar-hillel, and D. Weinshall. Computing gaussian mixture models with EM using equivalence constraints. In *In Advances in Neural Information Processing Systems 16*. MIT Press, 2003.

[21] S. D. Spiegelhalter et al. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B*, 64(4):583–639, 2002.

[22] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. Split and merge EM algorithm for improving gaussian mixture density estimates. *J. VLSI Signal Process. Syst.*, 26(1-2):133–140, 2000.

[23] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[24] K. Wagstaff, C. Cardie, and S. Schroedl. Constrained k-means clustering with background knowledge. In *ICML*, pages 577–584, 2001.

[25] Q. Zhao and D. J. Miller. Mixture modeling with pairwise, instance-level class constraints. *Neural Computation*, 17(11):2482–2507, 2005.