SEMISUPERVISED LEARNING OF MIXTURE MODELS WITH CLASS CONSTRAINTS

Qi Zhao and David J. Miller

Department of Electrical Engineering
The Pennsylvania State University
e-mail:millerdj@ee.psu.edu

ABSTRACT

Most prior work on semisupervised clustering/mixture modeling with given class constraints assumes the number of classes is *known*, with each learned cluster assumed to be a class and, hence, subject to the given instance-level constraints. When the number of classes is incorrectly assumed and/or when the "one-cluster-per-class" assumption is not valid, the use of constraint information in these methods may actually be *deleterious* to learning the ground-truth data groups. In this work we extend semisupervised learning with constraints 1) to allow allocation of *multiple* mixture components to individual classes and 2) to estimate both the number of components/clusters *and*, leveraging the constraint information, the number of *classes* present in the data. For several real-world data sets, our method is shown to correctly estimate the number of classes and to give a favorable comparison with the recent mixture modeling approach of Shental et al.

1. INTRODUCTION

The objective of unsupervised clustering is to extract the hidden structure in a given data set. Among various clustering methods, mixture modeling is a very important one, often performing better than the hard-partitional clustering algorithms such as K-means. However, there are significant challenges associated with mixture modeling, including (poor) local maxima of learning, how to choose the parametric form of the mixture component densities, and how to choose the *number* of components/clusters in the solution[1]. While there are many model selection approaches, there is as yet no consensus on the proper choice for the case of limited data.

In recent years, the utility of "side information" in clustering – to help avoid poor local optima and to learn the clustering distortion metric [1] or, equivalently, the form of the mixture component densities [2] – has been investigated. Early work in this area, known broadly as *semisupervised learning* e.g. [3], assumed the existence of class labels for some of the data samples. A less restrictive form of "side information", applicable in some situations where label information is inaccessible or inappropriate, is *pairwise sample constraints* – i.e., an indication that a given pair of samples *does* belong to the same group (a *must-link* (ML) constraint) or does *not* belong to the same group (a *cannot-link* (CL) constraint). Note that this form of side information does not explicitly specify the class label for any data samples, nor does it necessarily even indicate how *many* classes are involved in the problem[2]. Moreover, whereas a labeled set of samples *entails* class constraints between all pairs of samples in the set, a set of samples, each possessing some constraints, does *not*, in general, determine labels for *any* of the samples. Non-exhaustively, we can identify two general sources/scenarios where constraints may be the side information of choice:

1. *Domain knowledge*, including spatial information in images, temporal continuity in video, and some other prior knowledge. For example, pixels near an image border are likely to represent image background. This can be expressed as must-link constraints for pairs of pixels near the border and cannot-links for (border,center-of-field) pixel pairs.
2. *Interactive, on-line databases:* here, users/experts are solicited to provide supervision information for records in the database. Even supposing that such users provide category labels (as opposed to constraints) for a subset of examples, individual users may not conform to a common convention for class names/labels or even agree on the *number* of classes. In this situation, the individual users' labeled examples cannot be pooled in a simple way to form an aggregate semisupervised data set; however, each user's labeled instances entail must-link and cannot-link constraints that *are* reasonably pooled across all users. Moreover, instead of inferring constraints from user-supplied labels, constraints on pairs of examples may be *directly* elicited from users.

Prior Work on Learning with Instance-Level Constraints

Both ML and CL constraints are considered in the modified K-means algorithm in [4]. This method minimizes a hard clustering distortion, but while ensuring that no constraints are violated. In this approach, the data is first partitioned into *chunklets*, obtained by applying transitive closure to the ML constraints, i.e. chunklets consist of disjoint data subsets constrained to belong to the same class, as entailed by the specified ML constraints. Then a variant of K-means is applied, one satisfying the ML and CL constraints, with each individual cluster treated as a distinct class. The "nearest neighbor" step in this method involves iterative sequential assignment of each chunklet to the nearest cluster (class) consistent with the chunklet's CL constraints. In [5], pairwise constraints were introduced within a *graph-based* clustering framework, applied to image segmentation. First, constraints were obtained based

This work was supported by National Science Foundation grant NSF IIS-0082214.

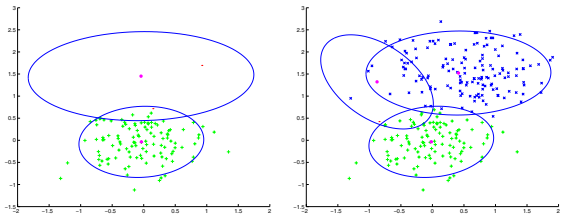
[1] In the sequel we use the terminology 'components' and 'clusters' interchangeably.

[2] At the same time, constraints do provide information about the number of classes that may be present – this information will be leveraged for estimating the number of classes in our work.

on grouping cues from the image. Next, these constraints were smoothed in order to propagate them spatially. The smoothed constraints were incorporated into a normalized cuts clustering objective. The use of constraints was found to substantially improve segmentation results even though only ML constraints were considered. Klein et al. [6] proposed a hierarchical, complete-linkage agglomerative clustering algorithm with integrated metric learning. Here, the distance measure was modified based on ML constraints, with CL constraints enforced during the cluster merging. Metric learning has been considered in, e.g.,[1] and [7]. Our approach, through its use of multiple components per class, effectively performs a type of "local" metric learning. Constraint information has also been integrated within the learning of "soft" clustering solutions, i.e. Gaussian mixture models [2]. While this work is closely related to ours, there are key differences which will be shortly discussed. The approach in [2] was extended in [8] to consider soft constraints. As indicated later, our approach also incorporates constraints in a soft fashion.

Clusters and Classes

Our method learns a mixture model for the data with individual clusters capturing homogeneous groups of points. However, the clusters are not necessarily treated as classes, individually subject to the given instance-level constraints. Rather, we allow classes to be composed of one or *more* clusters, with the allocation of clusters to classes chosen to best satisfy the given (class) constraints. All previous works have assumed one cluster per class. The potential difficulties stemming from this assumption are illustrated in Fig. 1 for the method from [2]. For this data set, one of the two classes consists of two ground-truth clusters, with the other class a singleton cluster. [2] assumes one cluster per class and requires specifying the number of classes. This method learns general covariances for individual clusters so that the cluster shape can be adapted to better satisfy the given constraints. If this method assumes 2 classes (Fig. 1a), it has difficulty capturing 2 ground truth components within one of its learned classes/clusters. On the other hand, if 3 classes are assumed (Fig. 1b), ML constraints within one of the ground truth classes (between 2 ground truth components) make it difficult to capture the true cluster structure. The method

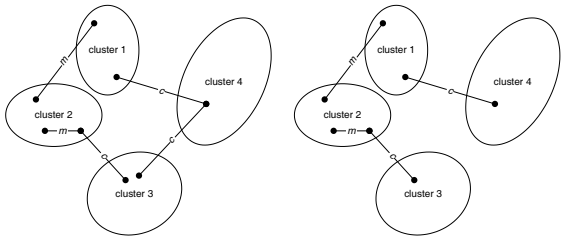


(a) (b)

Fig. 1. 2-D data from 3 components but only 2 classes, with given ML and CL constraints. Mixture model solutions for [2] assuming 2 classes (a)) and 3 classes (b)).

we propose here accurately learns both the clusters and classes for this example, as well as for more complex mixture distributions.

In addition to assuming one cluster per class, most prior works assume the number of *classes* (same as clusters, in these approaches) is known. In our method we do not assume either the cluster number or class number are known. To estimate the number of clusters, we use a model selection criterion, i.e. the Bayesian Information Criterion [9]. For a given number of clusters, the number of *classes* is automatically estimated as a byproduct of the minimization of our learning objective function. This will be seen shortly. There are several factors which affect the ability of our method to accurately estimate the number of classes. An incorrect class number estimate may stem from inaccuracy in model assumptions (a mixture with known parametric density forms), local optima of learning, limited data, or inaccuracy in the estimation of the number of mixture components. Somewhat unrealistically, let us ignore any inaccuracy attributable to the above factors, supposing that the learning and model selection capture the ground truth mixture used to generate the given data. In this case, identifying the *classes* (and their number) boils down to identifying to which class each mixture component belongs. Whether or not this can be uniquely determined depends on both the *consistency* and *sufficiency* of the supplied constraint information. We will suppose the given constraints are logically consistent. Several different "constraint sufficiency" cases are show in Fig. 2. In Fig. 2a, the constraints are sufficient to uniquely discern the classes, while in Fig. 2b they are not. Even if the constraint information is *in principle* sufficient,



(a) Case 1 (b) Case 2

Fig. 2. Constraint Illustration. A *must-link* is denoted by *m*, while a *cannot-link* is denoted by *c*.

in practice there is a difficult learning problem to address in estimating these classes and their component/cluster constituents. We next develop our learning framework.

2. LEARNING FRAMEWORK

Assume a K-component mixture, the K components each belonging to one of a *maximum* of L_{m} classes, where $L_{\mathrm{m}} \leq K$. Extra parameters are needed to describe these relations. Let $\beta_{l|k}$ be the probability that component k is assigned to class l. Here we have $\sum_{l=1}^{L_{\mathrm{m}}} \beta_{l|k} = 1, k = 1, \cdots, K$. Note that if $\beta_{l|k} = 0, \forall k$, then class l is *not used* and the number of classes, as estimated by $\{\beta_{l|k}\}$, is *smaller* than L_{m}. The joint data likelihood of sample x_i and class label l, given generation by component k, is $\alpha_k \beta_{l|k} f(x_i|\theta_k)$. Based on this, the *complete data log-likelihood* can now be written:

$$U(M, V, \Theta) = -\sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{l=1}^{L} M_{ik} V_{kl} \log \left[\alpha_k \beta_{l|k} f(x_i|\theta_k)\right], \quad (1)$$

where $M = [M_{ik}]$ is the data assignment matrix with $M_{ik} = 1$ if sample x_i is assigned to component k; else $M_{ik} = 0$, and where V is the component assignment matrix with $V_{kl} = 1$ if component k is assigned to class l; else $V_{kl} = 0$. Our approach incorporates constraints by adding a *constraint penalty function* to the complete data log-likelihood. In particular, we consider the potential

$$U(M,V,\Theta) = -\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{l=1}^{L} M_{ik}V_{kl}\log\left[\alpha_k\beta_{l|k}f(x_i|\theta_k)\right] + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij}\sum_{l=1}^{L}\Big(\sum_{k=1}^{K} M_{ik}V_{kl}\Big)\Big(\sum_{k'=1}^{K} M_{jk'}V_{k'l}\Big)D, \tag{2}$$

where

$$C_{ij} = \begin{cases} 1 & : \text{ } \textit{cannot-link} \text{ between samples } x_i \text{ and } x_j \\ -1 & : \text{ } \textit{must-link} \text{ between samples } x_i \text{ and } x_j \\ 0 & : \text{ otherwise} \end{cases}, \tag{3}$$

and with D a positive value. Note that the penalty "softly" encourages constraint satisfaction, with the degree of enforcement determined by the value of D. While hard constraint satisfaction, as considered in [2], may be desirable in some cases, soft constraints are more appropriate when some constraints are unreliable and/or when the models for individual classes are not sufficiently powerful to satisfy all the constraints (see Fig. 1).

If one applies the Expectation-Maximization formalism to this "penalized" complete-data log likelihood, treating M and V as the missing data, one finds that it is *intractable* to compute the E-step – this is due to the coupling of missing variables introduced by the penalty term. Thus, in order to have a feasible algorithm for learning the mixture model parameters, we need to invoke an approximation.

2.1. Mean Field Approximation and Learning

It is well-known that the learning objective of the EM algorithm for mixture models can be stated as the minimization of the *free energy* $F = \langle U\rangle - H$, with $\langle U\rangle$ the expected complete data log-likelihood and H the entropy of the probabilistic assignments of data to components (which are the expected missing values). For the "penalized" likelihood in (2), this minimization can be stated as

$$\min_{\Theta, P(M,V|\mathcal{X})} F(U,P) = \langle U(M,V,\Theta)\rangle - H(P(M,V|\mathcal{X})) \tag{4}$$

where $\langle U(M,V,\Theta)\rangle = \sum_{\{M,V\}} P(M,V|\mathcal{X})U(M,V,\Theta)$. When there are no constraints, V vanishes in (4) and the optimal joint pmf $P(M|\mathcal{X})$ satisfies *statistical independence* of individual data assignments, i.e., $P(M|\mathcal{X}) = \prod_{i=1}^{N}\big(\sum_k M_{ik}P(M_{ik}|x_i)\big)$. Unfortunately, when constraints are introduced, one can verify that the optimal joint pmf $\tilde{P}(M,V|\mathcal{X})$, minimizing (4), does *not* simplify to a tractable form unless some approximation is applied. We invoke a *mean-field approximation* as applied, e.g., in [10]. In particular, we *approximate* this optimal joint pmf $\tilde{P}(M,V|\mathcal{X})$ by a pmf that *does* have a tractable, *factorized* form, which we denote by $P^0(M,V|\mathcal{X})$. The form of $P^0(M,V|\mathcal{X})$ is chosen to minimize the new free energy:

$$F(U^0,P) = \langle U^0(M,V)\rangle - H(P(M,V|\mathcal{X})) \tag{5}$$

where $U^0(M,V) = -\sum_{i=1}^{N}\sum_k\sum_l M_{ik}V_{kl}\mathcal{E}_{ikl}$. The *form* of the solution is a *tractable* Gibbs distribution due to the definition of $U^0(M,V)$:

$$\begin{aligned} P^0(M,V|\mathcal{X}) &= \frac{e^{-U^0(M,V)}}{\sum_{M',V'} e^{-U^0(M',V')}} \\ &= \prod_{i=1}^{N}\Big(\sum_{k,l} M_{ik}V_{kl}\frac{e^{-\mathcal{E}_{ikl}}}{\sum_{k',l'} e^{-\mathcal{E}_{ik'l'}}}\Big) \\ &= \prod_{i=1}^{N}\Big(\sum_{k,l} M_{ik}V_{kl}\cdot P(M_{ik},V_{kl}|x_i)\Big) \end{aligned} \tag{6}$$

The *parameters* $\mathcal{E}_{ikl}$ approximate the average interaction of $M_{ik}V_{kl}$ with other assignment variables (seen from (2)). We wish to choose the field parameters $\mathcal{E} = \{\mathcal{E}_{ikl}, \forall i,k,l\}$ so as to make the tractable $P^0(M,V|\mathcal{X})$ as close as possible to the optimal, intractable joint pmf $\tilde{P}(M,V|\mathcal{X})$. For concision of expression, we subsequently drop $\mathcal{X}$ from these pmfs. Choosing relative entropy as the criterion with $\tilde{P}(M,V)$ treated as the prior, we pose

$$\begin{aligned} &\min_{\mathcal{E}} I(P^0(M,V)||\tilde{P}(M,V)) \\ &= \min_{\mathcal{E}}\Big\{\sum_{M,V} P^0(M,V)U(M,V,\Theta) - H(P^0(M,V))\Big\}. \end{aligned} \tag{7}$$

After taking the derivative with respect to $\mathcal{E}$, setting to zero, and after some manipulation, we obtain the *mean-field equations*:

$$\begin{aligned} \mathcal{E}_{ikl} &= -\log[\alpha_k\beta_{l|k}f(x_i|\theta_k)] \\ &\quad + D\sum_{j=1,j\neq i}^{N} C_{ij}[\langle M_{jk}\rangle + \sum_{k'\neq k}\langle M_{jk'}V_{k'l}\rangle]. \end{aligned} \tag{8}$$

Then, the marginal posterior probabilities (factors) of $P^0(M,V|\mathcal{X})$ satisfy

$$\begin{aligned} \langle M_{ik}V_{kl}\rangle &\equiv \text{Prob}(M_{ik}=1, V_{kl}=1|x_i) \quad \propto \quad e^{-\mathcal{E}_{ikl}} \\ &= \alpha_k\beta_{l|k}f(x_i|\theta_k)\prod_{j=1,j\neq i}^{N} e^{-C_{ij}D[\langle M_{jk}\rangle + \sum_{k'\neq k}\langle M_{jk'}V_{k'l}\rangle]} \end{aligned} \tag{9}$$

Since relative entropy is non-negative, we have that

$$\begin{aligned} F(U,\tilde{P}) &\leq F(U^0,P^0) + \sum_{M,V} P^0(M,V)[U(M,V,\Theta) - U^0(M,V)] \\ &= \sum_{M,V} P^0(M,V)\cdot U(M,V,\Theta) - H(P^0(M,V)). \end{aligned} \tag{10}$$

From (10) and (7), we see that $P^0(M,V|\mathcal{X})$ is chosen to minimize an upper bound on the original free energy $F(U,\tilde{P})$.

We also minimize this upper bound with respect to the *model parameters*. Taking the derivative of (10) with respect to the parameters and setting to zero, we obtain the necessary optimality conditions:

$$\alpha_k = \frac{\sum_i\sum_l\langle M_{ik}V_{kl}\rangle}{N} = \frac{\sum_i\langle M_{ik}\rangle}{N} \tag{11}$$

$$\beta_{l|k} = \frac{\sum_i\langle M_{ik}V_{kl}\rangle}{\sum_{l'}\sum_i\langle M_{ik}V_{kl'}\rangle} = \frac{\sum_i\langle M_{ik}V_{kl}\rangle}{\sum_i\langle M_{ik}\rangle} \tag{12}$$

$$\mu_k = \frac{\sum_i \langle M_{ik} \rangle x_i}{\sum_i \langle M_{ik} \rangle} \tag{13}$$

$$\Sigma_k = \frac{\sum_i \langle M_{ik} \rangle (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i \langle M_{ik} \rangle} \tag{14}$$

Here we have assumed Gaussian components, i.e., $\{\theta_k\} = \{\mu_k, \Sigma_k\}$.

(9) and (11)-(14) form the basis for an iterative algorithm minimizing the free energy (10). Given fixed associations $\{\langle M_{ik} V_{kl} \rangle\}$ (and $\langle M_{ik} \rangle = \sum_{l=1}^{L_{\mathrm{m}}} \langle M_{ik} V_{kl} \rangle$), (11)-(14) directly specify M-step parameter updates. Given fixed parameters, (9) specifies a fixed point update equation for $\{\langle M_{ik} V_{kl} \rangle\}$. One point of caution concerns how the updates of $\langle M_{ik} V_{kl} \rangle$ are carried out. Updating these associations *sequentially*, i.e. one association at a step, is guaranteed to descend in the free energy (10) [10]. Updating $\langle M_{ik} V_{kl} \rangle \ \forall i, k, l$ *in parallel* according to (9) is *not* guaranteed to descend. In practice, oscillations may occur if D is made too large.

3. EXPERIMENTAL RESULTS

We compared our method against [2] on several real-world data sets from the UC Irvine repository. As in [2], the performance of the methods was evaluated via a combined measure of purity P and accuracy A scores, defined as follows: $\rho = \frac{2PA}{P+A}$, where purity (P) measures the homogeneity of estimated classes, i.e., how many of the estimated class points belong to a single true class, and accuracy (A) measures how many of the true class points reside in a single estimated class (rather than being spread over several estimated classes). About 30% of the data points were given constraints. As preprocessing, standard principal component analysis was used to reduce the dimension for the data sets *Ecoli*, *Indian diabetes*, *Ionosphere*, and *Breast cancer*, down to the dimension shown in Table 1. For the *Ecoli* data set, the original 8 classes were merged into 3 classes. Specifically, the 6 classes with different membrane locations were merged into one because some of them own too few data points to support the cluster structure. Some comments are in order regarding the choice of D in our method. As noted earlier, parallel updates do not guarantee convergent learning. We have observed non-monotonicity of learning iterations, just as in [10], when D is made too large. Two possible solutions are to use sequential updates or to use parallel updates but adopt a relatively small value of D. For these experiments we chose the latter approach, using parallel updates and $D = 2$. The results are shown in Table 1. The most important statement to make about these results is that our method (MCGMM) correctly estimated the number of classes present, for *each* of these real-world data sets. For example, for *Ecoli*, BIC-based model selection yielded $K = 5$ and, thus, $L_{\mathrm{m}} = 5$. However, in the mixture solution with $K = 5$, as expressed by the $\{\beta_{l|k}\}$, only three classes were used. The purity-accuracy performance of our method was the best on most data sets. One interesting observation is that tied-covariance (TC) versions of both methods often improved the results, indicating that the data sets were too small to support use of full covariances.

4. CONCLUSIONS

In this work, we have extended semisupervised learning with constraints in several respects: 1) to improve the representation of the classes by allowing multiple components per class, with component allocation automatically determined by the learning; 2) unlike previous approaches, our method automatically estimates the *number* of classes in the data.

	SGMM	MCGMM
Ecoli	0.7925	0.8525,K=5
(N=336,d=5,L=3)	0.8615(TC)	0.8742(TC)
Liver disorders	0.6348	0.6946,K=2
(N=345,d=6,L=2)	0.6527(TC)	0.7221(TC)
Indian diabetes	0.6394	0.7103,K=3
(N=768,d=6,L=2)	0.6255(TC)	0.7410(TC)
Breast cancer	0.9306	0.9446,K=5
(N=683,d=4,L=2)	0.9720(TC)	0.9635(TC)
Ionosphere	0.9054	0.8884,K=3
(N=351,d=15,L=2)	0.7963(TC)	0.8041(TC)

Table 1. The results for UC Irvine data sets. "TC" means that only one, tied covariance is used for all components. N is the number of data points, d is the reduced dimension, L is the number of classes, and K is the cluster number determined via the BIC model cost.

5. REFERENCES

[1] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell, "Distance metric learning with application to clustering with side-information," in *NIPS*, 2002.

[2] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall, "Computing gaussian mixture models with EM using equivalence constraints," in *NIPS*. 2003, The MIT press.

[3] M. Seeger, "Learning with labeled and unlabeled data," Tech. Rep., University of California at Berkeley, 2000.

[4] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained k-means clustering with background knowledge," in *ICML*, 2001, pp. 577–584.

[5] S.X. Yu and J. Shi, "Segmentation given partial grouping constraints," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 2, pp. 173–183, Feb 2004.

[6] D. Klein, S.D. Kamvar, and C.D. Manning, "From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering," in *ICML*, 2002.

[7] S. Basu, M. Bilenko, and R. Mooney, "Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering," in *ICML-2003 Wkshp. on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining Systems*, Washington DC, 2003, pp. 42–49.

[8] M.H.C. Law, A. Topchy, and A.K. Jain, "Clustering with soft and group constraints," in *Joint IAPR Intl. Wkshp. Syntactical and Structural Pattern Recognition and Statistical Pattern Recognition*, 2004.

[9] G. Schwarz, "Estimating the dimension of a model," *The Annals of Stats.*, vol. 6, no. 2, pp. 461–464, 1978.

[10] T. Hofmann and J.M. Buhmann, "Pairwise data clustering by deterministic annealing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 1, pp. 1–14, Jan 1997.