A MIXTURE MODEL AND EM ALGORITHM FOR ROBUST CLASSIFICATION, OUTLIER REJECTION, AND CLASS DISCOVERY

David J. Miller and John Browning

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

ABSTRACT

Several authors have addressed learning a classifier given a mixed labeled/unlabeled training set. These works assume each unlabeled sample originates from one of the (known) classes. Here, we consider the scenario in which unlabeled points may belong either to known/predefined *or* to heretofore *undiscovered* classes. There are several practical situations where such data may arise. We propose a novel statistical mixture model which views as observed data not only the feature vector and the class label, but also the *fact* of label presence/absence for each point. Two types of mixture components are posited to explain label presence/absence. “Predefined” components generate both labeled and unlabeled points and assume labels are *missing at random*. “Non-predefined” components only generate unlabeled points – thus, in localized regions, they capture data subsets that are *exclusively unlabeled*. Such subsets may represent an outlier distribution, or new classes. The components’ predefined/non-predefined natures are *data-driven*, learned along with the other parameters via an algorithm based on expectation-maximization (EM). There are three natural applications: 1) robust classifier design, given a mixed training set with outliers; 2) classification with rejections; 3) identification of the unlabeled points (and their representative components) that originate from unknown classes, i.e. *new class discovery*. We evaluate our method and alternative approaches on both synthetic and real-world data sets.

1. INTRODUCTION

Several authors have proposed use of unlabeled data, along with labeled data, when learning a statistical classifier e.g. [1], [2], [3]. For classifiers based on an underlying statistical model and maximum likelihood (ML) for model learning, it has been found that augmenting a small pool of labeled data with a larger pool of unlabeled data can improve accuracy in the estimation of the class-conditional densities, and hence in classification[1]. These works assume that no unlabeled samples are outliers. Moreover, they assume all the data originate from one of the *known* classes, defined for the given domain. However, the presence of unlabeled samples may suggest that one or both of these assumptions is faulty – in either case, the labeler may have had difficulty in labeling some of the data. It is well-known that even a few outliers can have a dramatic, deleterious effect on estimation.

When all data from a particular ground truth class are unlabeled, we will say that *the data set contains an unknown class*. This definition covers two different scenarios:
i) where the problem involves some classes that are defined for the domain, but for which no labeled data has been made available, e.g. character recognition where all training instances of ‘2’ and ‘z’ are left unlabeled due to uncertainty on the part of the data labeler. Likewise, the labeler may have randomly selected a subset of points for labeling that happened not to include any instances of ‘z’; ii) where the data includes examples from classes that have not even been *defined* for the given domain, e.g. a galaxy data set which includes unlabeled examples from a galaxy type that has not yet been discovered by astronomers. In this case, the new galaxy type is unknown, both with respect to the training set *and* more generally.

There are several reasons why, in practice, some (often many) samples may be missing labels. Providing labels is a time-consuming activity that in some domains also requires expertise and hence expense. Also, if outlier samples are present, these may be difficult to label. There are also several scenarios where unknown classes may arise. If a huge database (e.g. text, multispectral, scientific) starts out purely unlabeled, then an expert may first need to *choose* the set of classes and *then* to label a subset of the data. Since the size of the data set precludes exhaustive human exploration, it is quite possible the expert might overlook some meaningful groups when defining the classes. Unknown classes may also occur if there is uncertainty as to the origin of the data set or to the environment from which it was obtained, e.g. land use classes associated with multispectral data may depend on whether the land is agrarian, urban, industrial, or military. Unknown classes may also arise when there is disagreement or subjectivity even in the definition of a *core* set of classes, e.g. the choice of function classes for genes in molecular biology [4], or text categorization. Finally, unknown classes may occur in scientific domains when their existence is inconsistent with current theory.

Data sets of this type – with mixed labeled/unlabeled samples and with some unknown classes – are relevant to a number of tasks, including robust classifier learning, outlier detection, sam-

THIS WORK WAS SUPPORTED BY NATIONAL SCIENCE FOUNDATION GRANT NSF IIS-0082214.

[1] Unlabeled data is not always helpful and in some cases can lead to performance degradation. However, empirical studies do suggest that augmenting a small labeled corpus with a large unlabeled pool can lead to significantly enhanced classification accuracy in certain domains [1],[2],[3].

ple rejection, and the emergent problem of new class discovery. In this work, we propose a novel statistical mixture model, along with ML learning and model-based inference, tailored for these data sets and directly applicable to all of the aforementioned tasks. In section 2, we develop our mixture model and ML learning. In section 3, we identify several different inference rules based on our model, useful in various tasks. In section 4 our experimental results are reported. The paper concludes with discussion of some related and future work.

2. MIXTURE MODEL AND EM ALGORITHM

Consider a data set $\mathcal{X}_m = \{\mathcal{X}_l, \mathcal{X}_u\}$, where $\mathcal{X}_l = \{(\underline{x_1}, c_1), (\underline{x_2}, c_2), \ldots, (\underline{x_{N_l}, c_{N_l}})\}$ is the *labeled* subset, with $\underline{x_i} \in \mathcal{R}^n$ and $c_i \in \mathcal{P}_c$, and where $\mathcal{X}_u = \{\underline{x_{N_l+1}}, \ldots, \underline{x_N}\}$ is the *unlabeled* subset[2]. Here, $\mathcal{P}_c$ is the set of known classes. This mixed labeled/unlabeled data scenario was considered in [1],[2], and [3]. Unlike these works, a key element in our approach is that we treat the fact of 'missingness' for unlabeled samples *as observed data*. Accordingly, we redefine $\mathcal{X}_m = \{\mathcal{X}_l, \mathcal{X}_u\}$, where now $\mathcal{X}_l = \{(\underline{x_1}, \text{“l”}, c_1), (\underline{x_2}, \text{“l”}, c_2), \ldots, (\underline{x_{N_l}}, \text{“l”}, c_{N_l})\}$ and $\mathcal{X}_u = \{(\underline{x_{N_l+1}}, \text{“m”}), \ldots, (\underline{x_N}, \text{“m”})\}$. Here we have introduced the *new* random observation $\mathcal{L} \in \{\text{“l”}, \text{“m”}\}$, taking on values indicating a sample is either *labeled* or *missing* the label. We propose a mixture model that explains *all* the observed data, including the fact of label presence/absence. Two types of mixture components are posited, differing in the mechanism they use for generating label presence or absence. "Predefined" components generate both labeled and unlabeled points and assume labels are *missing at random*. "Non-predefined" components *only* generate unlabeled points – thus, in localized regions, they capture data subsets that are *purely* unlabeled. Such subsets may represent an outlier distribution or new classes.

Let $\mathcal{M}_k$, $k = 1, \ldots, M$ denote the kth mixture component, with M the number of components. Let $\mathcal{C}_{\text{pre}}$ denote the subset of 'predefined' components, with the remaining subset denoted $\bar{\mathcal{C}}_{\text{pre}}$. Let $C \in \mathcal{P}_c$ be a random variable defined over the predefined classes, with $c(\underline{x}) \in \mathcal{P}_c$ the class label paired with $\underline{x}$. Let α_k denote the prior probability for component k with $\sum_{k=1}^{M} \alpha_k = 1$, θ_k the parameter set specifying component k's (component-conditional) feature density, and let $f(\underline{x}|\theta_k)$ denote this density. We also define the probability that a class label is produced, given that the sample-generating component is 'predefined', i.e. $P[\mathcal{L} = \text{“l”}|\mathcal{M}_p]$, where $\mathcal{M}_p$ generically denotes a predefined generating component. Finally, the probability that $C = c$ given that the sample is generated by predefined component k and given that a label is produced, i.e. $P[C = c|\mathcal{M}_k, \mathcal{L} = \text{“}l\text{”}]$ [3]. In summary, our model is based on the parameter set $\Lambda = \{\{\alpha_k\}, \{\theta_k\}, \mathcal{C}_{\text{pre}}, \{P[\mathcal{L}|\mathcal{M}_p]\}, \{P[C|\mathcal{M}_k, \mathcal{L} = \text{“}l\text{”}]\}\}$.

Hypothesis for Random Generation of the Data

Our model hypothesizes that each sample from $\mathcal{X}_m$ is generated independently, based on Λ, according to the following stochastic generation process:

i) Select component $\mathcal{M}_j$ according to $\{\alpha_k\}$ and then $\underline{x}$ using $f(\underline{x}|\theta_j)$.
ii) If $\mathcal{M}_j \in \mathcal{C}_{\text{pre}}$, select $v \in \{\text{“l”}, \text{“m”}\}$ based on $P[\mathcal{L}|\mathcal{M}_p]$.
iii) If $\mathcal{M}_j \in \mathcal{C}_{\text{pre}}$ and $v = \text{“l”}$, select a label c based on $P[C|\mathcal{M}_j, \text{“l”}]$. Form the datum $(\underline{x}, c, \text{“l}'')$.
iv) If $\mathcal{M}_j \in \bar{\mathcal{C}}_{\text{pre}}$, form the datum $(\underline{x}, \text{“m”})$.

Note that with this model, we effectively have $P[\mathcal{L} = \text{“m”}|\mathcal{M}_{\text{np}}] = 1$, i.e. non-predefined components ($\mathcal{M}_{\text{np}}$) *deterministically* explain missing labels, whereas predefined ones hypothesize labels missing at random.

Joint Data Likelihood

Let $v_k = 1$ if $\mathcal{M}_k \in \mathcal{C}_{\text{pre}}$, else $v_k = 0$. Then, the joint data likelihood is

$$L_m = \left(\prod_{\underline{x} \in \mathcal{X}_u} \sum_{k=1}^{M} \alpha_k f(\underline{x}|\theta_k)((1 - v_k) + v_k P[\text{“m”}|\mathcal{M}_p]) \right) \times \left(\prod_{\underline{x} \in \mathcal{X}_l} \sum_{k=1}^{M} v_k \alpha_k f(\underline{x}|\theta_k) P[\text{“l”}|\mathcal{M}_p] P[C = c(\underline{x})|\mathcal{M}_k, \text{“l”}] \right) \tag{1}$$

EM Algorithm

Before developing our learning algorithm based on EM [5], it is informative to give a brief aside justifying our model and learning approach. In particular, consider the $\{v_k\}$ variables, which specify $\mathcal{C}_{\text{pre}}$. We treat these 0-1 variables as *parameters*, to be learned. However, there are two alternatives: i) we could allow components to be predefined/nonpredefined *in probability* and, thus, learn the parameters $\{P[\mathcal{M}_k \in \mathcal{C}_{\text{pre}}]\}$; ii) we could treat the $\{v_k\}$ as *missing* data and estimate their expected values within the EM framework. It turns out that neither of these approaches yields practically feasible learning. We illustrate for case i). The data set in this case is generated by *first* randomly generating the predefined/nonpredefined nature for each component – there are 2^M such configurations. Then, for the chosen configuration, the data is stochastically generated as described before. Unfortunately, this model's likelihood is obtained by *averaging* over the 2^M *configuration dependent* likelihoods, with each such likelihood based on (1). The expectation is measured with respect to the joint configuration pmf $\{P[V_1 = v_1, V_2 = v_2, \ldots, V_M = v_M] = \prod_{m=1}^{M} P[\mathcal{M}_k \in \mathcal{C}_{\text{pre}}]^{v_k} (1 - P[\mathcal{M}_k \in \mathcal{C}_{\text{pre}}])^{(1-v_k)}\}$. The concomitant complexity of learning (based e.g. on the EM algorithm) grows exponentially with the number of components. This can be avoided by treating the $\{v_k \in \{0, 1\}\}$ as *parameters to be learned*, as seen next.

2.1. Formulation

We perform an iterative optimization with each iteration consisting of two steps: i) maximize $\log L_m$ over the natures $\{v_k\}$ given the remaining parameters in Λ held fixed; ii) using the EM algorithm, maximize the remaining parameters, given the $\{v_k\}$ held fixed. Each step is nondecreasing in $\log L_m$.

Optimization over Component Natures

There are several approaches to maximizing $\log L_m$ over the $\{v_k \in \{0, 1\}\}$. One choice, if M is not too large, is simply exhaustive search over all 2^M configurations. A second method is to use a

[2] For concreteness, in the subsequent development we will consider feature vectors $\underline{x} \in \mathcal{R}^n$.

[3] For concision, we do not explicitly indicate that $\mathcal{M}_k \in \mathcal{C}_{\text{pre}}$.

global optimization technique. As a computationally simple alternative, when exhaustive search is infeasible, we propose to iteratively select the $\{v_k\}$ one component at a time, keeping the remaining ones held fixed. Each v_k is chosen simply by evaluating $\log L_m$ for the two cases, $v_k = 0, 1$ and selecting the value yielding the greater likelihood. Cycling over the components continues until there are no further changes. This does not guarantee convergence to a global, or even a local optimum. However, it does guarantee ascent in $\log L_m$.

EM Algorithm for the Remaining Parameters
Let $\Gamma = \Lambda - \{v_k\}$. Further, denote the estimates after the t-th EM iteration by $\Gamma^{(t)}$. Our EM method optimizes Γ given fixed $\{v_k\}$.

E-step:
Following the EM framework [5], we define the *missing* data quantities $\{V_{\underline{x}k}\}$, with $V_{\underline{x}k} = 1$ if $\underline{x} \in \mathcal{M}_k$ and 0 otherwise, and with $\sum\limits_k V_{\underline{x}k} = 1$. Here, $\underline{x} \in \mathcal{M}_k$ means that $\underline{x}$ was generated by $\mathcal{M}_k$. In the E-step, we compute the expected complete data log-likelihood given the current parameter estimates, $\Lambda^{(t)} \equiv \{\Gamma^{(t)}, \{v_k\}\}$. This quantity is based on the expectations $E[V_{\underline{x}k}|\underline{x} \in \mathcal{X}_l; \Lambda^{(t)}]$ and $E[V_{\underline{x}k}|\underline{x} \in \mathcal{X}_u; \Lambda^{(t)}]$. Note that, based on the definition of $V_{\underline{x}k}$, these expected quantities are simply probabilistic assignments of points to components, i.e. $E[V_{\underline{x}k}|\underline{x} \in \mathcal{X}_l; \Lambda^{(t)}] \equiv P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_l; \Lambda^{(t)}]$. These probabilities, derived via Bayes rule, are given by

$$P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_l; \Lambda^{(t)}] = \frac{v_k \alpha_k^{(t)} f(\underline{x}|\theta_k^{(t)}) P[c(\underline{x})|\mathcal{M}_k, \text{“l”}]^{(t)}}{\sum\limits_{n \in \mathcal{C}_{\text{pre}}} \alpha_n^{(t)} f(\underline{x}|\theta_n^{(t)}) P[c(\underline{x})|\mathcal{M}_n, \text{“l”}]^{(t)}} \tag{2}$$

$$P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_u; \Lambda^{(t)}] = \frac{v_k \alpha_k^{(t)} f(\underline{x}|\theta_k^{(t)}) P[\text{“m”}|\mathcal{M}_p]^{(t)}}{\sum\limits_{n \in \mathcal{C}_{\text{pre}}} \alpha_n^{(t)} f(\underline{x}|\theta_n^{(t)}) P[\text{“m”}|\mathcal{M}_p]^{(t)} + \sum\limits_{n \in \bar{\mathcal{C}}_{\text{pre}}} \alpha_n^{(t)} f(\underline{x}|\theta_n^{(t)})} + \frac{(1-v_k)\alpha_k^{(t)} f(\underline{x}|\theta_k^{(t)})}{\sum\limits_{n \in \mathcal{C}_{\text{pre}}} \alpha_n^{(t)} f(\underline{x}|\theta_n^{(t)}) P[\text{“m”}|\mathcal{M}_p]^{(t)} + \sum\limits_{n \in \bar{\mathcal{C}}_{\text{pre}}} \alpha_n^{(t)} f(\underline{x}|\theta_n^{(t)})} \tag{3}$$

M-step:
For concreteness, suppose that $f(\cdot)$ is a joint Gaussian density function, with parameter set given by the mean vector and covariance matrix, i.e. $\theta_k = \{\underline{m}_k, \Sigma_k\}$ [4]. In the M-step, the expected complete data log likelihood is maximized over the Γ parameters, yielding $\Lambda^{(t+1)} = \{\Gamma^{(t+1)}, \{v_k\}\}$. For our model, this M-step is given by the (decoupled) parameter estimates[5]:

$$\alpha_k^{(t+1)} = \frac{\sum\limits_{\underline{x} \in \mathcal{X}_l} P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_l; \Lambda^{(t)}] + \sum\limits_{\underline{x} \in \mathcal{X}_u} P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_u; \Lambda^{(t)}]}{N} \quad \forall k \tag{4}$$

$$\underline{m}_k^{(t+1)} = \frac{\sum\limits_{\underline{x} \in \mathcal{X}_l} \underline{x} P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_l; \Lambda^{(t)}] + \sum\limits_{\underline{x} \in \mathcal{X}_u} \underline{x} P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_u; \Lambda^{(t)}]}{\sum\limits_{\underline{x} \in \mathcal{X}_l} P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_l; \Lambda^{(t)}] + \sum\limits_{\underline{x} \in \mathcal{X}_u} P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_u; \Lambda^{(t)}]} \quad \forall k \tag{5}$$

$$P[c|\mathcal{M}_k, \text{“l”}]^{(t+1)} = \frac{\sum\limits_{\underline{x} \in \mathcal{X}_l : c(\underline{x}) = c} P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_l; \Lambda^{(t)}]}{\sum\limits_{\underline{x} \in \mathcal{X}_l} P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_l; \Lambda^{(t)}]} \tag{6}$$

$$\forall c, k : \mathcal{M}_k \in \mathcal{C}_{\text{pre}}$$

$$P[\text{“l”}|\mathcal{M}_p]^{(t+1)} = \frac{\sum\limits_{\underline{x} \in \mathcal{X}_l} \sum\limits_{k \in \mathcal{C}_{\text{pre}}} P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_l; \Lambda^{(t)}]}{\sum\limits_{\underline{x} \in \mathcal{X}_l} \sum\limits_{k \in \mathcal{C}_{\text{pre}}} P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_l; \Lambda^{(t)}] + \sum\limits_{\underline{x} \in \mathcal{X}_u} \sum\limits_{k \in \mathcal{C}_{\text{pre}}} P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_u; \Lambda^{(t)}]}. \tag{7}$$

Comments:
1) Both the $\{v_k\}$ and EM optimizations ascend in $\log L_m$. Thus, our iterative method is a hillclimbing algorithm in L_m.
2) Initially, we choose $v_k = 1 \ \forall k$. This choice is in some sense least biased, since it is difficult to have any *a priori* knowledge of which components are ‘non-predefined’.
3) If a component switches from $v_k = 1$ to $v_k = 0$, its predefined parameters $\{P[C = c|\mathcal{M}_k, \text{“l”}]\}$ are held static and saved for use in the optimization of component natures. If the component later switches to $v_k = 1$, the EM algorithm will update these parameters starting from their current, saved values.

3. STATISTICAL INFERENCES FROM THE MODEL

Our learned model is naturally applied to classification (to the known classes), predefined vs. unknown class discrimination, and sample rejection. For classification, we require evaluation of the *a posteriori* predefined class probabilities. These probabilities are given (via Bayes rule) by:

$$P[C = c|\underline{x}; \Lambda] = \frac{\sum\limits_{k \in \mathcal{C}_{\text{pre}}} \alpha_k f(\underline{x}|\theta_k) P[C = c|\mathcal{M}_k, \text{“}l\text{”}]}{\sum\limits_{k \in \mathcal{C}_{\text{pre}}} \alpha_k f(\underline{x}|\theta_k)}, c \in \mathcal{P}_c. \tag{8}$$

In order to discriminate between the hypotheses that an unlabeled sample originates from a predefined versus an unknown class, we need the *a posteriori* probability that a given feature vector is generated by a non-predefined component. This is simply

$$P[\mathcal{M}_{\text{np}}|\underline{x} \in \mathcal{X}_u] = 1 - \sum_{k \in \mathcal{C}_{\text{pre}}} P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_u; \Lambda], \tag{9}$$

where $P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_u; \Lambda]$ is given in (3). (9) also forms the basis for sample rejections when the goal is classification to one of the predefined classes. Usually in pattern recognition, samples are rejected when it is determined that a reliable decision cannot be reached. Alternatively, our model rejects samples *under the hypothesis that they are either outliers or belong to unknown classes.*

[4] It is straightforward to modify the M-step development here for other continuous feature models, as well as for categorical feature models, e.g. a naive Bayes model. The choice of multivariate Gaussians is merely for illustration.

[5] We omit the update of Σ_k for concision of expression. Its form follows naturally from those given for α_k and $\underline{m}_k$.

4. EXPERIMENTAL RESULTS

Synthetic and real-world data were used for evaluation. For the synthetic data, we generated five 2-D sets of 560 samples, each consisting of seven isotropic, equal-mass Gaussian components, with random centers and variance one in both dimensions for all components. Three of the components represent predefined classes and four represent unknown classes. The second data set is Deterding's 10-D *vowel* set. This has 990 samples consisting of 11 vowels where we used the first 6 as known classes and the last 5 as unknown classes. For all the experiments, 50% of the data from known classes (and all data from unknown classes) was taken to be unlabeled. For both predefined/unknown class discrimination and for classification (to one of the known classes), we compared with a method which we dub 'Supervised Clustering' and with a method based on [2][6]. In Supervised Clustering, we first perform standard Gaussian mixture modelling using only the labeled data. Each component is then hard-assigned to its majority class. For classification, the component centers then simply form a nearest-prototype classifier. For predefined/unknown class discrimination, an unlabeled sample is declared to be from an unknown class if it is deemed to be an outlier with respect to each of these (predefined) components. This is determined by thresholding the component's (Gaussian) density, evaluated at the sample[7]. For the method based on [2], each mixture component is deemed 'predefined' if it "owns" any labeled samples (in the MAP sense); else, the component is 'non-predefined'. Predefined/unknown class discrimination is thus performed similar to our method, based on (9), but with $P[\mathcal{M}_k|\underline{x} \in \mathcal{X}_u]$ now specified in [2]. For our method, iterative cycling was used to choose the $\{v_k\}$. For all three competing methods, the model size was selected to minimize a minimum description length (MDL) cost[8]. Predefined/unknown class discrimination performance, measured over the unlabeled data subset, is shown in Table 1. The synthetic data results are averaged over all five data sets. Classification performance is shown in Table 2, again measured on the unlabeled subset. For all three methods, a two-step classification was invoked, with predefined/unknown class discrimination first applied, and then, for samples deemed predefined, classification to one of the known classes. If either step is incorrect, an error is counted. Note the poor performance of [2], which assumes all the training points come from known classes.

Method	Synthetic data	Vowel data
new EM	.115	.212
Supervised clustering	.245	.344
[2]	.515	.365

Table 1. Fraction of incorrect predefined/unknown class decisions for the synthetic data and the vowel data.

Method	Synthetic data	Vowel data
new EM	.139	.319
Supervised clustering	.249	.447
[2]	.548	.526

Table 2. Fraction of incorrect classifications on the two data sets.

5. RELATED AND FUTURE WORK

One prior work on robust learning for mixed data sets is [6]. This is a mixture modelling approach similar to [1] except that a robust variant of the EM algorithm was employed. The main modelling differences between our approach and [6] are that i) we treat label presence/absence as observed data; ii) we explicitly model the new classes/outliers. Thus, unlike [6] our approach can be directly used to tackle new class discovery. The class discovery problem has been considered before, e.g [4]. There, all the data was assumed labeled, with new classes corresponding essentially to *mislabeled* data, rather than to purely unlabeled components. While we have only considered the mixed labeled/unlabeled scenario here, we believe that an extension of our approach suitable for the correctly labeled/mislabeled scenario can also be developed.

We view class discovery as consisting of two (stratified) goals. The first, addressed here, is to identify the subset of points which do not belong to known classes. A more ambitious objective is to *validate* the components which own these 'unknown class' samples. In our current work, we have only evaluated our MDL-based model selection strategy in terms of the classification and predefined versus unknown class discrimination performance. Model selection for the validated discovery of new classes remains to be investigated. In addition, we plan to investigate applications to knowledge discovery in Internet searches and to scientific domains.

6. REFERENCES

[1] B. Shashahani and D. Landgrebe. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Trans. on Geoscience and Remote Sensing*, vol. 32, pages 1087-1095,1994.

[2] D. Miller and H. S. Uyar. A Mixture of Experts Classifier with Learning based on both Labelled and Unlabelled Data. In *Advances in Neural Information Processing Systems 9*, pages 571–577, 1997.

[3] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, Text classification from labeled and unlabeled documents using EM, *Machine Learning*, pp. 1-34, 2000.

[4] A. Sierra and F. Corbacho. Reclassification as Supervised Clustering. *Neural Computation*,12 pages 2537-2546, 2000.

[5] A. Dempster, N. Laird and D.B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Roy. Stat. Soc., Series B*, vol. 39, pages 1-38 1977.

[6] S. Tadjudin and D. Landgrebe. Robust Parameter Estimation for Mixture Model. *IEEE Trans. on Geoscience and Remote Sensing*, vol. 38, pages 439-445, 2000.

[6] Since our particular mixed data scenario is, we believe, novel, there are very few existing competing methods. Thus, we created two methods for comparison.

[7] The threshold was chosen to make both error types equally likely.

[8] This cost is $\frac{N_p}{2}\log N - \log L$, with N_p the number of model parameters and L the likelihood.