Semi-supervised Learning with an Imperfect Supervisor

Massih R. Amini and Patrick Gallinari

Department of Computer Science, University Pierre and Marie Curie,
8 rue du capitaine Scott 75015 Paris, France
{last_name}@poleia.lip6.fr

Abstract. Real-life applications may involve huge datasets with misclassified or partially classified training data. Semi-supervised learning and learning in the presence of label noise have recently emerged as new paradigms in the machine learning community to cope with this kind of problems. This paper describes a new discriminant algorithm for semi-supervised learning. This algorithm optimizes the classification maximum likelihood of a set of labeled-unlabeled data, using a discriminant extension of the Classification Expectation Maximization algorithm. We further propose to extend this algorithm by modeling imperfections in the estimated class labels for unlabeled data. The parameters of this label-error model are learned together with the semi-supervised classifier parameters. We demonstrate the effectiveness of the approach using extensive experiments on different datasets.

1 Introduction

Most statistical classifiers rely on a supervised learning paradigm where a decision function is to be build from a labeled data set $D_l = \{(x_i, y_i) \mid i = 1, ..., n\}$, in which each example is described by a pattern x_i and by the response of a supervisor y_i. Under this paradigm, data are supposed to be drawn independently from a joint distribution $p(x, y)$ and the learned decision rule is supposed to capture the relation between these two variables.

In practice, labeling large amounts of data may require extensive human resources and is often unrealistic. For example, for many *Information Retrieval* problems, labeling data is a time consuming and a difficult task. For other problems such as *medical imaging*, labeling data may require very expensive tests so that only a small set of labeled data may be available. Sometimes noise is inherent to the labeling process which further complicates the learning problem. In such cases, modeling the label noise can lead to performance increase.

The statistics and pattern recognition communities have considered these problems and proposed different methods for learning with partially labeled and misclassified labeled data. More recently, both problems have attracted the interest of the machine learning community.

Our motivation here is to develop efficient and robust semi-supervised algorithms. Most studies on the semi-supervised paradigm rely on a generative approach. Using a mixture density model where mixture components are identified to classes, they attempt to maximize the joint likelihood of labeled and unlabeled data using the `EM` algorithm (Dempster et al, 1977). Recent papers have proposed new semi-supervised models based on the discriminative approach. This is a natural way for classification since discriminant models usually perform better than generative ones.

We describe here new semi-supervised algorithms for classification. Our main contributions are algorithmic and experimental. We first propose a new discriminant semi-supervised algorithm (section 5). We then introduce a label error model which aims at improving the labelings of unlabeled data computed by semi-supervised algorithms. This error model will be learned together with the classifier parameters and may be used in both generative and

discriminative semi-supervised settings (section 6). We also provide an extensive set of experiments for evaluating empirically the contribution of these different ideas and comparing the proposed algorithms with baseline classifiers and state of the art semi-supervised methods.

Throughout the paper, all algorithms will be described using the general setting of Classification Maximum likelihood (CML) and Classification EM (CEM) approaches (Celeux et al, 1982). The main benefit gained by introducing this formalism is that CML, CEM and their extensions described here do provide a natural framework for expressing both generative and discriminative approaches for the semi-supervised learning problem. This is not the case for the usual maximum likelihood approach or for any other method.

CML and CEM have been proposed as a general framework for clustering algorithms using generative models. CML has been extended to semi-supervised learning for generative models by McLachlan (McLachlan, 1992). We introduce here a *discriminative* form of the CML criterion and a CEM algorithm for optimizing this discriminative CML over labeled and unlabeled data. This CEM algorithm will serve as a basis for the proposed discriminant semi-supervised algorithm. Considering only unlabeled data, this would amount to perform clustering with discriminant classifiers by estimating cluster posteriors instead of conditional densities as it is usually done in most clustering approaches (including plain EM). When considering only labeled data, CML reduces to cross-entropy between targets and estimated class posteriors. Maximizing the discriminative CML criterion in the presence of partially labeled data will lead to optimize simultaneously the cross entropy for labeled data and the CML for unlabeled data.

Whereas the reason for using discriminant methods for semi-supervised learning is evident, the motivation for using a label error model requires an explanation. Both the generative and discriminative versions of semi-supervised CEM are iterative algorithms which compute at each iteration tentative labels for unlabeled data. Together with labels of labeled data, these tentative labels are considered as target labels when updating the system parameters. Let us now consider an *ideal* learning problem, where the true labels of unlabeled data would be known. With respect to this *ideal* classification problem, the semi-supervised learning algorithm will compute, at each iteration, erroneous class labels for part of the unlabeled data set. Such labeling errors are inherent to any semi-supervised algorithm. We will make the hypothesis that these label errors do correspond to a stochastic process and propose to learn an error model on labels predicted by the algorithm, simultaneously with the system parameters. As will be seen in the experimental section, the combination of discriminant learning and label error modeling will prove very efficient on all the datasets which have been used here.

The paper is organized as follows. After a review of related existing work on semi-supervised learning and learning in the presence of label noise in section 2, we present the baseline generative CEM algorithm for semi-supervised learning (section 4). In section 5, we introduce our discriminant semi-supervised algorithm using the CEM framework. In section 6, we describe the label-error model for semi-supervised learning and its use for generative and discriminative models. We finally show that discriminant semi-supervised learning performs well on different real size data sets and that modeling the label-error process is a valuable addition to the semi-supervised models for both generative and discriminative cases (section 7. In the appendix (section 9), we detail instances of this model first for the generative semi-supervised CEM by considering discrete and real valued data and then for the discriminant semi-supervised CEM algorithm.

2 Related Work

In the following, we will review the main approaches for respectively semi-supervised learning and learning in the presence of label noise.

2.1 Learning with partially classified training data

A discussion of the respective merits of discriminative and generative approaches to semi-supervised learning and an excellent review of work prior to 2000 in the machine learning community is given by (Seeger, 2000). We give below a synthetic presentation of the work performed by the statistics and the machine learning communities on this subject by distinguishing between generative and discriminative approaches.

Generative approaches Most generative approaches to semi-supervised learning rely on mixture models. In this case, unlabeled data are supposed to be generated from a mixture density while the mixture components of labeled data are known. Training criterion is usually the data log-likelihood of all (labeled + unlabeled) observations. Once the model parameters are learned, unknown data are classified using the mixture components associated to each class.

Algorithmic studies A review of work prior to 1992 in the statistics community may be found in (McLachlan, 1992). Most of these approaches are based on an iterative `EM`-like algorithm working under the assumption of multivariate normal components with a common covariance matrix (McLachlan et al, 1982). Some authors have suggested updating procedures for no-normal group conditional densities using for example kernel estimators for the mixture components (Murray et al, 1978). Most papers in machine learning developed the same type of ideas. (Miller et al, 1996) considered a mixture model where each class is described by several component densities. Nigam et al proposed a semi-supervised `EM` algorithm for text classification making use of a naive Bayes estimator for modeling the different densities (Nigam et al, 2000). Basu et al presented a semi-supervised algorithm for clustering, using available classified data to first generate seed clusters and then to guide the clustering process (Basu et al, 2002). Chapelle and Zien proposed a semi-supervised method which use distributional hypotheses on data in order to find decision boundaries laying down in low density regions (Chapelle et al, 2005).

Theoretical issues There are still very few theoretical contributions for understanding semi-supervised learning and most crucial questions are still open. Some authors attempted to characterize the role and the importance of unlabeled samples for learning. For a mixture of Gaussians in a 2-class classification task, O'Neill et al. proved that if the Mahalanobis distance between the class centroids of two populations is over a fixed threshold, unlabeled data may be helpful for learning (ONeill, 1978). Using the same distributional assumptions, Castelli and Cover proved that with known observation densities and unknown mixing parameters, the labeled and unlabeled data play a very similar role in reducing the probability of classification error (Castelli et al, 1996). They further showed that for unknown observation densities and known mixing parameters the probability of error converges exponentially to zero in the number of labeled samples. Another analysis of the value of unlabeled samples by learning from a mixture of densities was carried out by (Ratsaby et al, 1995) within the PAC framework. They reached the same conclusion as (Castelli et al, 1996), showing that the probability of error decreases exponentially fast in the number of labeled data while it

decreases only as an inverse polynomial in the number of unlabeled observations. Using also the PAC model, (Cozman et al, 2003) were more controversial in their conclusions about the role of unlabeled data for semi-supervised learning. They showed via an example that for *incorrect* models with parameters Θ, for which the data distribution $p(x, y)$ does not belong to the family $p(x, y \mid \Theta)$, performance may degrade with unlabeled training data.

Discriminative approaches More recently, some authors have proposed discriminant algorithms for semi-supervised learning. In the following, we distinguish between co-training like methods which consider multi-modal data representations and others which use classical vector representations.

Multi-modal view The co-training paradigm has been proposed by (Blum et al, 1998) for training classifiers when examples may be described by two modalities assumed conditionally independent given the class variable. Two classifiers are used, one on each modality, operating alternatively as teacher and learner, i.e. tentative labels estimated by the output of a classifier for unlabeled data are used to train the other classifier. Collins and Singer presented an interesting extension of the boosting algorithm which incorporates co-training to perform named entity classification (Collins et al, 1999). The work of (DeSa, 1993) also bears similarities with this technique. Muslea et al. proposed to combine active and semi-supervised learning using the multi-modal view framework (Muslea et al, 2002).

Uni-modal view Up to our knowledge, the earliest work for discriminant semi-supervised learning is the one proposed by Anderson who used together labeled and unlabeled observations for training a logistic regression classifier (Anderson, 1979). As learning criterion, he proposed to maximize the joint likelihood of labeled and unlabeled data. Recently, Joachims proposed a transductive version of SVMs (Joachims, 1999), where each new unlabeled example is used to modify the parameters of an existing classifier. Bennett and Demiriz found small improvements on UCI datasets with this type of transduction (Bennet et al, 1998). Roth and Steinhage proposed a framework for semi-supervised learning which extends classical linear discriminant analysis (LDA) to kernel discriminant analysis (KDA) (Roth et al, 1999). Szummer and Jaakkola presented a kernel expansion algorithm which augments the representation of examples using a Fisher score vector estimated over both labeled and unlabeled data (Szummer et al, 2001). Using these new feature vectors, they derived Bayes optimal decision boundary from the maximum entropy and maximum likelihood frameworks. The runtime of their algorithm is proportional to the product of the number of labeled observations and the total number of examples. Jaakkola et al. proposed a general framework for classification based on maximum entropy discrimination which extends the semi-supervised paradigm (Jaakkola et al, 2000). Recently different authors have studid the geometric structure of the data sets for partially labeled classification. For example, Zhu et al. have presented a gaussian random field model for semi-supervised learning (Zhu et al, 2003) and Belkin and Niyogi have proposed approaches based on the geometry of manifolds (Belkin et al, 2004).

2.2 Learning with misclassified training data

There are different cases where the actual labels of training observations may be subject to error. Practical applications, like remote-sensing, have motivated in the early 70's an intensive research in the pattern recognition community on the problem of learning in the presence of label noise. These studies distinguished between *random* and *no-random* imperfect supervisions, for the latter the probability of misclassification of an observation does depend on its feature vector while it does not for the former.

Random imperfect supervision Random imperfect supervision may arise in the case where the labeling of the training data is made automatically on the basis of a machine output, for example in blood test results (Aitchison et al, 1976). In such cases, the error on the class label does not depend on the input x. McLachlan studied conditional error rates using their asymptotic expansions for the case where one group does not get mislabeled sample (McLachlan, 1972). Chittineni obtained error bounds on the performance of Bayes and nearest neighbour classifiers trained with imperfect labeled observations (Chittineni, 1980). Chhikara and McKoen proved that training classifiers by ignoring mislabeling in the training set can degrade classification performance (Chhikara et al, 1984). Using the maximum likelihood principle, (Krishnan et al, 1987) derived the likelihood estimation of parameters for two group multivariate normal mixtures with a common covariance matrix in a 2-classes classification problem. Under this framework, Krishnan studied the *efficiency* of an imperfect supervision scheme compared to a perfect supervision case (Krishnan, 1988). This efficiency, called the *Asymptotic Relative Efficiency*, measures the relative sample sizes required in both the perfect and imperfect supervision cases in order to achieve the same classification performance. More recently, Lawrence and Scholkopf proposed an algorithm for constructing a kernel Fisher discriminant from training examples in the presence of label noise (Lawrence et al, 2001).

No-random imperfect supervision In the same context of medical diagnosis, no-random imperfect supervision would correspond to the case where the classification of patient diseases is carried out by human experts using disease symptoms. Lachenbrunch studied conditional error rates of no-random misclassification models using Monte Carlo methods (Lachenbrunch, 1974). He expressed the probability of mislabeling as a function of the distance between each sample and its group mean. In (Titterington, 1989), Titterington worked out an `EM` algorithm for estimating the parameters of a logistic-normal distribution (Aitchison, 1986). More recently, Ambroise and Govaert proposed an `EM` algorithm (Ambroise et al, 2000) to estimate the posterior distribution of the true label class with respect to the incomplete data.

As opposed to most studies on imperfect supervised learning, we do not assume here that label errors do come from the data acquisition or from a manual labeling process. The noise over labels does come here from the classification algorithm itself and the label error model will tend to correct these mislabelings.

Like many other approaches, the semi-supervised algorithms proposed here work by iteratively predicting labels for unlabeled data. At each iteration these predicted labels are considered as targets and classifier parameters are learned to predict them. Thus, at each labeling step, the semi-supervised learning system acts as an imperfect *supervisor* for unlabeled data. As the misclassification of an example does not depend on its feature vector, we propose to model this label-error process using the *Random imperfect supervision* framework.

3 Notations

We suppose that each example belongs to one and only one class and that there are available a set of n labeled examples $D_l = \{(x_i, y_i) \mid i = 1, ..., n\}$ and a set of m unlabeled examples $D_u = \{x_i \mid i = n+1, ..., n+m\}$. A classifier is to be trained on the basis of these $n+m$, d-dimensional feature vectors $x \in \mathbb{R}^d$. For each labeled example x_i in D_l, let y_i and $t_i = \{t_{ki}\}_k$ be respectively the class label and the indicator vector class associated to x_i.

$$\forall i \in D_l, \forall k, y_i = k \Leftrightarrow t_{ki} = 1 \text{ and } \forall h \neq k, t_{hi} = 0 \tag{1}$$

During training, unlabeled samples will be given tentative labels. Let $\widetilde{y}$ and $\widetilde{t}$ denote respectively the class label and the class indicator vector of an unlabeled observation x estimated

with a learning system.
The probability density function (pdf) of an observation x in class k is denoted by $f_k(x)$ for $k = 1, ..., c$. $f_k(x, \theta_k)$ will denote a parametric model of this pdf. Θ denotes the vector of all unknown parameters for a generative model, θ_k being the parameters of the k^{th} class.

Under the generative approach, it is assumed that each observation x has been drawn from a mixture of c groups in proportions $\pi_1, ..., \pi_c$, respectively, where

$$\sum_{k=1}^{c} \pi_k = 1 \text{ and } \forall k, \pi_k \geq 0 \tag{2}$$

For discriminant approaches, we will directly estimate the posteriors. $G_k(x_i, \beta_k) = p(y = k \mid x, \beta_k)$ will denote class k posterior estimator function, β_k the specific classifier parameters corresponding to class k and B the set of all classifier parameters.

All methods described here work by iteratively partitioning the unlabeled dataset D_u into c classes. C will denote such a partition. $C^{(j)}$ is then a partition of D_u found at iteration j and $C_k^{(j)}$ the k^{th} class of $C^{(j)}$.

4 Background: `CEM` and generative semi-supervised learning

`CEM` is a general clustering algorithm which relies on a mixture density model of the data. Both `CEM` and its variants have always been used in this generative setting. We first describe below the baseline `CEM` algorithm and then its extension for semi-supervised learning.

4.1 Classification Maximum Likelihood estimation and Classification Expectation-Maximization algorithm

(Symons, 1981) distinguishes two main approaches to clustering: maximum likelihood (ML) and classification maximum likelihood (`CML`). The former optimizes the data likelihood by modeling the component densities, clustering is then performed using the estimated densities and Bayes rule. The latter directly optimizes the classification of data into different clusters. For both approaches, samples are supposed to be generated via a mixture density:

$$p(x, \Theta) = \sum_{k=1}^{c} \pi_k f_k(x, \theta_k) \tag{3}$$

For `CML`, each example belongs to exactly one mixture component and the `CML` criterion is the *complete* data log-likelihood:

$$L_{CML}(C, \Theta) = \sum_{x_i \in D_u} \sum_{k=1}^{c} \widetilde{t}_{ki} \ \log p(x_i, \widetilde{y} = k, \Theta) \tag{4}$$

Here, the class indicator vectors $\tilde{t}$ of unlabeled samples are model parameters and have to be estimated together with the parameters Θ. `CEM` is a general framework which encompasses most `CML` clustering algorithms. It is a general algorithm which aims to find the mixing weights of classes π_k, the parameters θ_k of density functions modeling the data and the clusters C_k under the `CML` approach (Celeux et al, 1982). This algorithm can be seen as a classification version of the `EM` algorithm: it contains an additional `C-step` (Algorithm 1), where each unlabeled example x_i is assigned to one and only one component of the mixture (between the *Expectation* and the *Maximization* steps of the `EM` algorithm).

4.2 Generative Semi-supervised CEM algorithm

McLachlan has extended CML and CEM for generative algorithms to the case where both labeled and unlabeled data are used for learning (McLachlan, 1992, p. 39). In this context, the indicator vector class for labeled data are known whereas they are estimated for unlabeled data. The complete-data log likelihood criterion (4) becomes:

$$L_c(C,\Theta) = \sum_{i=1}^{n}\sum_{k=1}^{c} t_{ki}\ \log p(x_i, y=k, \Theta) + \sum_{i=n+1}^{n+m}\sum_{k=1}^{c} \widetilde{t}_{ki}\ \log p(x_i, \widetilde{y}=k, \Theta) \tag{5}$$

In this expression, the first summation is over the labeled examples, and the second one over the unlabeled samples.

CEM can then be easily adapted to the case of semi-supervised learning by maximizing (5) instead of (4). Algorithm 1 describes both the unsupervised and semi-supervised versions of the CEM algorithm. For the former the initial partition $C^{(0)}$ is chosen randomly and the

Algorithm 1 Generative unsupervised and semi-supervised CEM

Initialization: for unsupervised learning, an initial partition $C^{(0)}$ is chosen at random and the $f_k(.,\theta_k^{(0)})$ are estimated on the corresponding classes. For semi-supervised learning, $f_k(.,\theta_k^{(0)})$ are respectively estimated on the k classes from the labeled data D_l, and $C^{(0)}$ is defined accordingly.
j^{th} iteration, $j \geq 0$:

- **E-step**: Estimate the posterior class probability that each unlabeled example x_i belongs to $C_k^{(j)}$:

$$\forall x_i \in D_u, \forall k \in \{1,...,c\}, E[\tilde{t}_{ki}^{(j)} \mid x_i; C^{(j)}, \Theta^{(j)}] = \frac{\pi_k^{(j)} f_k(x_i, \theta_k^{(j)})}{p(x, \Theta^{(j)})}$$

- **C-step**: Assign each $x_i \in D_u$ to the cluster $C_k^{(j+1)}$ with maximal posterior probability according to $E[\tilde{t} \mid x]$. Let $C^{(j+1)}$ be the new partition.
- **M-step**: Estimate the new parameters $\Theta^{(j+1)}$ which maximize
  $L_{CML}(C^{(j+1)}, \Theta^{(j)})$ for unsupervised learning
  $L_c(C^{(j+1)}, \Theta^{(j)})$ for semi-supervised learning

$f_k(.,\theta_k^{(0)})$ are estimated on this partition. The three steps are then iterated until convergence. For the latter, initial density components $f_k(.,\theta_k^{(0)})$ are estimated using labeled data from class k. The $\widetilde{t}_{ki}$ for unlabeled data are then estimated as in the classical CEM (**C-step**) while they are kept fixed, in all iterations, to their known value for labeled data. For unlabeled data, the class conditional probabilities are estimated (**E-step**) and a classification decision is then made according to Bayes rule (**C-step**). In both cases, the algorithm will converge to a local maximum of $L_{CML}(C^{(j+1)}, \Theta^{(j)})$ for unsupervised learning and $L_c(C^{(j+1)}, \Theta^{(j)})$ for semi-supervised learning.

For comparison, the semi-supervised ML criterion writes (McLachlan, 1992), (Nigam et al, 2000) :

$$L_{ML}(\Theta) = \sum_{i=1}^{n}\sum_{k=1}^{c} t_{ki} \log p(x_i, y=k, \Theta) + \sum_{i=n+1}^{n+m} \log(\sum_{k=1}^{c} \widetilde{t}_{ki} p(x_i, \widetilde{y}=k, \Theta)) \tag{6}$$

The first term corresponds to labeled data (the same as for CML), the second one is the likelihood of unlabeled data instead of the classification likelihood for CML.

We show below how CML can be adapted to a discriminative approach instead of the usual generative one. This formulation will allow to handle the semi-supervised learning problem with a whole set of discriminant techniques (Vittaut et al, 2000), and will lead to a new family of discriminant semi-supervised algorithms.

CML and CEM will thus provide a unified framework for a wide range of semi-supervised techniques, both generative and discriminative. This is the reason why we have introduced this framework here. Usually, generative methods are developed under the ML framework and discriminative methods using a risk minimization setting.

5 Discriminant semi-supervised CEM algorithm

The generative approach to semi-supervised learning indirectly computes posteriors $p(\widetilde{y} = k \mid x, \Theta)$ via conditional density estimation. This is known to lead to poor estimates for high dimensions or when only few data are labeled which is exactly the interesting case for semi-supervised learning.

A more natural approach would be to use a discriminant model in order to directly estimate posterior probabilities. The discriminant semi-supervised algorithm described below makes use of a discriminant classifier. It explicitly makes the hypothesis that the outputs of this classifier estimate the class posteriors.

Let us rewrite the complete data log-likelihood so as to put in evidence the role of posterior probabilities:

$$L_c(C, B) = \sum_{i=1}^{n} \sum_{k=1}^{c} t_{ki} \ \log p(y = k \mid x_i, B) + \sum_{i=n+1}^{n+m} \sum_{k=1}^{c} \widetilde{t}_{ki} \ \log p(\widetilde{y} = k \mid x_i, B) + \sum_{i=1}^{n+m} p(x_i) \quad (7)$$

As no assumption is made on the distributional nature of data, maximizing L_c is equivalent to the maximization of a the following criterion L'_c (McLachlan, 1992, p. 261):

$$L'_c(C, B) = \sum_{i=1}^{n} \sum_{k=1}^{c} t_{ki} \ \log p(y = k \mid x_i, B) + \sum_{i=n+1}^{n+m} \sum_{k=1}^{c} \widetilde{t}_{ki} \ \log p(\widetilde{y} = k \mid x_i, B) \quad (8)$$

(Algorithm 2) describes the semi-supervised discriminant CEM algorithm for optimizing (8). A discriminant classifier is first trained on the labeled data set D_l. The outputs of the classifier $G_k(., \beta_k)$ are then used to estimate the posteriors for unlabeled data. Each unlabeled example x_i is assigned to the class with maximum posterior. Indicator variables $\widetilde{t}_{ki}$ are defined accordingly (C step). Using this set of labels the classifier is trained to optimize L'_c (M step). These new estimators are then used in the next iteration so as to provide new posterior estimates and therefore new labels for the unlabeled data. Note that in this algorithm, labels for labeled data are always kept fixed to their *true* value since they are known. The E-step is trivial here since the posterior estimates are given by the classifier outputs, it does not explicitly appear in algorithm 2. This algorithm iterates the two steps C and M until it converges to a local maximum of $L'_c(C, B)$ (8).

Note that the first term in equation (8) corresponds to the cross-entropy between the true class and the class posterior estimates which is a classical training criterion for supervised learning. The second term in (8) is the CML criterion discussed previously. For both labeled and unlabeled data, algorithm 2 maximizes simultaneously the cross-entropy for labeled data and the CML for unlabeled data. When there is no unlabeled data, this reduces to a discriminant algorithm trained on cross-entropy (initialization step only). When there is only unlabeled

Algorithm 2 semi-supervised discriminant `CEM`

Initialization: Train a discriminant model $LR_{B^{(0)}}$ over D_l. Let $G_k(.,\beta_k^{(0)})$ be the corresponding initial discriminant function estimates.
j^{th} iteration, $j \geq 0$:

- `C-step`: Assign each $x_i \in D_u$ to the cluster $C_k^{(j+1)}$ with maximal posterior probability according to the output G of the classifier estimating $p(\widetilde{y}^{(j)} = k \mid x_i, B)$:

$$\forall x_i \in D_u, \forall k \in \{1,c\}, \widetilde{t}_{ki}^{(j+1)} = \begin{cases} 1 \text{ if } p(\widetilde{y}^{(j)} = k \mid x_i, B) = max_h\, p(\widetilde{y}^{(j)} = h \mid x_i, B) \\ 0 \text{ otherwise} \end{cases}$$

  Let $C^{(j+1)}$ be the new partition obtained from this classifier for the unlabeled data.
- `M-step`: Find new parameters $B^{(j+1)}$ which maximize

$$L'_c(C^{(j+1)}, B^{(j)})$$

data, this is a clustering algorithm where clusters are directly attributed via discriminant functions instead of density estimation as this is usually the case.

This algorithm is guaranteed to converge to a local maximum of the L'_c function (see section 6.3). It can be used with any discriminant classifier provided its outputs can be interpreted as posterior class probabilities and it can be trained to optimize criterion (8). Many different classifier families fulfill these two requirements.

We show below how both algorithms 1 and 2 can be improved by adding a label-error model. The latter will help improve the predictive labeling of unlabeled samples during the iterations of the semi-supervised `CEM` algorithms.

6 Learning a label-error model on tentative labels for semi-supervised classification

For both generative and discriminant semi-supervised `CEM`, tentative labels for unlabeled data are computed at each step. The classifiers are initialized on labeled data and iteratively improve tentative labels for unlabeled data by optimizing the classification likelihood. With respect to a classical supervised classification problem where all the labels were known, the algorithm solves at each iteration a classification problem for which some of the labels - among those computed for unlabeled data - are wrong. Such errors on tentative labels are inherent to semi-supervised learning algorithms. We will make here the hypothesis that the errors of the classifier on these tentative labels come from a stochastic process. If we knew this process, we could try to reduce the classifier errors at each iteration, which should improve the classification performance.

In this section, we propose to learn a probabilistic label-error model on tentative labels during `CEM` iterations simultaneously with the classifier parameters (Amini et al, 2003). This error model will apply only to tentative labels computed for unlabeled data. In our setting, labels of labeled data are known and considered as correct, so that they will not be changed throughout the algorithm iterations. After introducing some notations we show in sections 6.1 and 6.2 how this model could be estimated for respectively generative and discriminant semi-supervised learning. This will lead to two new semi-supervised algorithms which are respectively enhanced versions of algorithm 1 (generative semi-supervised learning) and algorithm 2 (discriminant semi-supervised learning). From now on, $\hat{y}$ and $\hat{t}$ will denote the computed class label and the indicator class vector of an unlabeled observation estimated

with the label-error model, while $\widetilde{y}$ and $\widetilde{t}$ denote as previously, the estimations made by the classifier before applying the label-error model.

Let us consider the *Mislabeling* probabilities:

$$\forall (k,h) \in \{1,...,c\}^2, \alpha_{kh} = p(\hat{y} = k \mid \widetilde{y} = h) \tag{9}$$

Which are subject to the constraint:

$$\forall h, \sum_{k} \alpha_{kh} = 1 \tag{10}$$

Consider now the joint probability of an example and its corrected label:

$$p(x_i, \hat{y} = k) = \sum_{h=1}^{c} p(x_i \mid \widetilde{y} = h, \hat{y} = k) \times p(\hat{y} = k, \widetilde{y} = h) \tag{11}$$

Assume further that:

$$p(x_i \mid \widetilde{y} = h, \hat{y} = k) = p(x_i \mid \widetilde{y} = h) \tag{12}$$

Using (12) and (9), (11) can be rewritten:

$$p(x_i, \hat{y} = k) = \sum_{h=1}^{c} \alpha_{kh} \times p(\widetilde{y} = h) \times p(x_i \mid \widetilde{y} = h) = \sum_{h=1}^{c} \alpha_{kh} \pi_h f_h(x_i, \theta_h) \tag{13}$$

6.1 Updating the parameters of a generative model using a label-error model for unlabeled data

The label-error model will attempt to correct imperfect labels $\widetilde{t}$ estimated with for unlabeled data. The complete-data log-likelihood is then computed with respect to the set of labeled data D_l and to the set of unlabeled data D_u with their corrected $\hat{t}_i,\ i \in \{n+1,...,n+m\}$. For this specification, the complete-data log likelihood writes:

$$L_c(C, \Theta, \Lambda) = \sum_{i=1}^{n} \sum_{k=1}^{c} t_{ki} \log p(x_i, y = k) + \sum_{i=n+1}^{n+m} \sum_{k=1}^{c} \hat{t}_{ki} \log p(x_i, \hat{y} = k) \tag{14}$$

When introducing the probability density functions f_k and the label-error model (9) into the training `CML` criterion (14), from (13) the latter writes:

$$L_c(C, \Theta, \Lambda) = \sum_{i=1}^{n} \sum_{k=1}^{c} t_{ki} \log(\pi_k f_k(x_i, \theta_k)) + \sum_{i=n+1}^{n+m} \sum_{k=1}^{c} \left[\hat{t}_{ki} \log \left(\sum_{h=1}^{c} \alpha_{kh} \pi_h f_h(x_i, \theta_h) \right) \right] \tag{15}$$

Let, $C^{(j)}$ be the current partition of data and $\Lambda^{(j)}$, $\Theta^{(j)}$ respectively the parameters for the misclassification model and for the generative model estimated at iteration j of the algorithm. The learning criterion (15) is a function of C, Θ and Λ. As in section 4.2, we adopt an iterative approach for its maximization. Parameters Θ are first initialized on the labeled data set D_l. The three steps (E, C, M), in Algorithm 3, are then iterated until the convergence to a local maximum of L_c. At each iteration in the `E` and the `C` steps, the error model modifies the assignment of unlabeled examples. All model parameters, including those of the label-error model, are modified in the `M-step`. The new value of the α will depend on the old values and on the current estimated conditional densities. We will provide a proof of convergence of the algorithm in section 6.3.

In the appendix, we give reestimations formulas for the α as well as the details of this algorithm for two particular instances (a discrete naive Bayes classifier and a gaussian model) which have been used in our experiments.

Algorithm 3 Generative semi-supervised `CEM` with label-error modeling

Initialization: $f_k(.,\theta_k^{(0)})$ are respectively estimated on the k classes corresponding to labeled data D_l, $\alpha_{kh}^{(0)}$ are initialized at random between 0 and 1
j^{th} iteration, $j \geq 0$:

- `E-step`: Estimate the joint class probability of each $x_i \in D_u$ and its corrected label:

$$\forall k, \forall i \in \{n+1, ..., n+m\}, p(x_i, \hat{y}^{(j)} = k) = \sum_h \alpha_{kh}^{(j)} \pi_h^{(j)} f_h(x_i, \theta_h^{(j)}) \tag{16}$$

- `C-step`: Assign each $x_i \in D_u$ to the cluster $C_k^{(j+1)}$ with maximal joint probability:

$$\forall i \in \{n+1, ..., n+m\}, \quad \hat{y}_i^{(j+1)} = \underset{k}{argmax}\, p(x_i, \hat{y}^{(j)} = k) \tag{17}$$

  Let $C^{(j+1)}$ be the new partition.
- `M-step`: Estimate the new parameters $(\Theta^{(j+1)}, \Lambda^{(j+1)})$ which maximize L_c
  - $\Theta^{(j+1)} = \underset{\Theta^{(j)}}{argmax}\, L_c(C^{(j+1)}, \Theta^{(j)}, \Lambda^{(j)})$
  - $\Lambda^{(j+1)} = \underset{\Lambda^{(j)}}{argmax}\, L_c(C^{(j+1)}, \Theta^{(j+1)}, \Lambda^{(j)})$

6.2 Updating the parameters of a discriminant model using a label-error model for unlabeled data

We show now how the error model is incorporated in the discriminant `CEM` algorithm (Algorithm 2). Following section 5, the modified complete-data log likelihood in the discriminative case using the corrected labels $\hat{t}$ for the unlabeled data writes:

$$L'_c(C, B, \Lambda) = \sum_{i=1}^{n} \sum_{k=1}^{c} t_{ki} \log p(y = k \mid x_i, B) + \sum_{i=n+1}^{n+m} \sum_{k=1}^{c} \hat{t}_{ki} \log p(\hat{y} = k \mid x_i, B) \tag{18}$$

Using Bayes rule and (13), we get:

$$p(x_i, \hat{y} = k) = p(x_i) \times \sum_{h=1}^{c} \{\alpha_{kh}\ p(\widetilde{y} = h \mid x_i)\} \tag{19}$$

From (19), (18) becomes:

$$L'_c(C, B, \Lambda) = \sum_{i=1}^{n} \sum_{k=1}^{c} t_{ki} \log p(y = k \mid x_i, B) + \sum_{i=n+1}^{n+m} \sum_{k=1}^{c} \left[\hat{t}_{ki} \log \left(\sum_{h=1}^{c} \alpha_{kh}\ p(\widetilde{y} = h \mid x_i, B) \right) \right] \tag{20}$$

The learning criterion (20) is a function of Λ, C and B. In this case, parameters B are first initialized by training the classifier on the labeled dataset D_l. Two steps are then iterated until the convergence of L'_c (Algorithm 4). In the first step, the classifier is considered as an imperfect supervisor for unlabeled data. To make a decision over the class of an unlabeled observation x, the outputs of the classifier, $G_k(x, \beta_k)$, are weighted using the mislabeling probabilities α_{kh}. In the `M-step`, the parameters of the label-error model and of the classifier are updated using the imperfect labels obtained in the previous step as well as the labeled data. At this step, a local maximum of $L'_c(C^{(j+1)}, B^{(j)}, \Lambda^{(j)})$ wrt the classifier parameters B

Algorithm 4 Discriminant semi-supervised learning with label-error modeling

Initialization: Train a discriminant model $LR_{B^{(0)}}$ over D_l, initialize the $\alpha_{kh}^{(0)}$ at random between 0 and 1. Let $C^{(0)}$ be the initial partition obtained from this model on D_u
j^{th} iteration, $j \geq 0$:

- `C-step`: Apply $LR_{B^{(j)}}$ on D_u, estimate the imperfect class posterior probabilities using the output $G_k^{(j)}(x, \beta_k)$ of the classifier. Apply the label-error model to the output of the model to obtain corrected labels for each $x_i \in D_u$:

$$\forall x_i \in D_u, \hat{y}_i^{(j+1)} = \underset{k}{argmax} \sum_{h=1}^{c} \alpha_{kh}^{(j)} \; p(\widetilde{y}^{(j)} = h \mid x_i)$$

  Let $C^{(j+1)}$ be the new partition obtained from this classifier for the unlabeled data.
- `M-step`:
  a) Maximize $L'_c(C^{(j+1)}, B^{(j)}, \Lambda^{(j)})$ with respect to $B^{(j)}$.
  b) Maximize $L'_c(C^{(j+1)}, B^{(j+1)}, \Lambda^{(j)})$ with respect to $\Lambda^{(j)}$ subject to constraints $\forall k, \forall h, \alpha_{kh}^{(j+1)} \in [0, 1]$ and $\forall h, \sum_k \alpha_{kh}^{(j+1)} = 1$.

and the error model Λ, will be reached. As for Algorithm 2, the `E-step` directly follows from the estimates $G_k(., \beta_k)$ and does not appear explicitly in Algorithm 4.

The iterative algorithm will converge to a local maximum of (20) as shown bellow. Reestimation formulas for the special case of a logistic classifier, used for the experiments, are provided in the appendix.

6.3 Convergence

Semi-supervised algorithms 1 to 4, converge to a local optimum of their objective function. We show below a proof for algorithm 4. The same proof applies for the other algorithms.

Lemma *The modified `CML` criterion, L'_c, increases for every sequence $\left(C^{(j)}, B^{(j)}, \Lambda^{(j)}\right)$ of the Algorithm 4 and the sequence $L'_c\left(C^{(j)}, B^{(j)}, \Lambda^{(j)}\right)$ converges to a stationary point.*

Proof. We first show that L'_c is increasing.

- Since $\forall x_i \in D_u$, $\hat{y}_i^{(j+1)} = k \Leftrightarrow p(\hat{y}^{(j+1)} = k \mid x_i) \geq p(\hat{y}^{(j+1)} = k' \mid x_i)$ for $k \neq k'$ (`C-step`), we have
$$L'_c(C^{(j+1)}, B^{(j)}, \Lambda^{(j)}) \geq L'_c(C^{(j)}, B^{(j)}, \Lambda^{(j)}) \tag{21}$$
- and, since $\left(C^{(j+1)}, B^{(j+1)}, \Lambda^{(j)}\right)$ and $\left(C^{(j+1)}, B^{(j+1)}, \Lambda^{(j+1)}\right)$ maximize iteratively L'_c (`M-step`), we have
$$L'_c(C^{(j+1)}, B^{(j+1)}, \Lambda^{(j+1)}) \geq L'_c(C^{(j+1)}, B^{(j)}, \Lambda^{(j)}) \tag{22}$$

Finally as there is a finite number of partitions of the example into c-classes, the increasing sequence $L'_c\left(C^{(j)}, B^{(j)}, \Lambda^{(j)}\right)$ takes a finite number of values and thus, converges to a stationary value. This is a local optimum of the objective function L'_c.

7 Experimental Results

We will now describe and analyze results obtained with the semi-supervised algorithms on datasets with different characteristics. We first describe these datasets and the evaluation measures we have been using. After that, we present and discuss a series of experiments.

7.1 Data sets

In our experiments we used the `Email spam` and `Mushroom` collections from the UCI repository[1] (Blake et al, 1998), the `7sectors` data set from the CMU Web-kB project[2] as well as the `Computation and Language (Cmp_lg)` collection of TIPSTER SUMMAC[3] for text summarization. All collections but `7sectors` correspond to 2-classes classification problems. Table 1 summarizes the characteristics of these datasets.

Table 1. Characteristics of the datasets and percentage of different classes.

Data sets		size	attributes	proportion for the classes (%)
`Email spam`		4601	57	39.4 - 60.6
`Mushroom`		8124 (−2480)	22	38.2 - 61.8
`Cmp_lg`		28985	5	10 - 90
	basic	949		21.2
	energy	355		7.9
	financial	964		21.5
`7sectors`	health	400	3000 for each	8.9
	transportation	511		11.4
	technology	998		22.3
	utilities	300		6.8

The `7sectors` data set consists of 4477 *html* articles partitioned in an hierarchical order. We labeled each document in this collection with its initial parent class label, namely `basic, energy, financial, health, transportation, technology` and `utilities`. In order to test the algorithms on many different classification problems, we considered $n(n-1)/2$ binary classification problems where each class is classified against any of the other classes. Documents are tokenized by removing *html* tags as well as words on a stop list. Low document frequency words (occurring in less than three documents) are also removed. Stemming is then performed using the Porter algorithm. For each training set, log-odds ratio feature selection (Mladenic et al, 1998) is used to prune the vocabulary to 3000 words. Documents are represented in the vector space by their term frequency.

The `Cmp_lg` collection is composed of 183 scientific articles. This collection is used for text summarization in the Summac competition (Summac, 1998). The aim is to propose a summary for each article in the collection. For this, we adopt the text-span extraction paradigm which amounts at selecting a subset of representative document sentences. This is a classification problem where sentences are to be classified as relevant or non-relevant for the extracted summary. There are 28985 sentences in the collection. We represent each sentence using a continuous version of the features proposed by (Kupiec et al, 1995). This characterization has given good results in previous work (Amini et al, 2002). Each sentence i, with length $l(i)$ is characterized by $\boldsymbol{x}_i = (\phi_1^i, \phi_2^i, \phi_3^i, \phi_4^i, \phi_5^i)$, where $\phi_1^i = \frac{l(i)}{\sum_j l(j)}$ is the normalized sentence length, $\phi_2^i = \frac{\text{frequency of cue words in } i}{l(i)}$ is the normalized number of cue words (such as "in conclusion", "this article", etc.) in sentence i, $\phi_3^i = \frac{\text{frequency of acronyms in } i}{l(i)}$ is the normalized number of acronyms (such as "USA", "NASA", "IBM", etc.) in i, ϕ_4^i is the position indicator feature of i in the document it belongs to (beginning, middle or end of the

[1] ftp://ftp.ics.uci.edu/pub/machine-learning-databases/
[2] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/bootstrappingIE/7sectors.tar.gz
[3] http://www.itl.nist.gov/iaui/894.02/related_projects/tipster_summac

document) and ϕ_5^i is the normalized number of common terms between a generic query q and the sentence i. In our experiments, query q is generated using the title and the most frequent words of each document. This is the usual setting for generic summarization.

`Email spam` and `Mushroom` are classical classification benchmarks from the UCI collection. The `Email spam` database consists in 4601 e-mails gathered manually from personal e-mails. The class label is spam or no-spam and there are 57 quantitative predictors.

The `Mushroom` collection is composed of 8124 observations corresponding to 23 species of mushrooms. Each observation is identified as edible or poisonous and characterized with 22 qualitative attributes. We removed 2480 examples with missing attributes from this database.

7.2 Evaluation measures

For the two UCI datasets there is approximately the same proportion of examples for each class (Table 1). We used as performance criterion the percentage of good classification (PGC) defined as:

$$\text{PGC} = \frac{\text{\# of examples in the test set correctly classified by the system}}{\text{\# of examples in the test set}} \tag{23}$$

For text summarization, we followed the SUMMAC evaluation by using a 10% compression ratio with respect to the original document. Hence, for each document in the test set, we have formed its summary by selecting the top 10% sentences having the higher score with respect to the output of the classifier. For evaluation, we compared these sentences with the desired summary of each document. The desired summaries were generated from the abstract of each article using the text-span alignment method described in (Banko et al, 1999). They too consist of 10% sentences from the original document so that desired and computed summaries have the same number of sentences. Since the collection is not well balanced between positive and negative examples, PGC is meaningless. For the evaluation, we used the average precision (AP) measure defined below.

$$\text{AP} = \frac{\text{\# of sentences extracted by the system which are in the target summaries}}{\text{total \# of sentences extracted by the system}} \tag{24}$$

Note that in this particular case, precision and recall are equal.

For the text classification task on the `7sectors` data set, we used the break even point measure.[4]

7.3 Experiments

For the experiments, we have proceeded as follows. For the generative algorithms we used a naive Bayes and a gaussian mixture classifier for respectively discrete and continuous data. For the discriminative approach, we used a logistic classifier. We also performed tests with more complex non-linear discriminant classifiers. Since the observed results and behavior were similar to the simple logistic system, the latter was adopted for the experiments here. The different algorithms (baselines, generative and discriminant with and without error model) are compared on different datasets. We also used the three classifiers (naive Bayes, Gaussian, logistic) as baseline fully supervised models by training them on the same proportion of labeled data as for the semi-supervised methods. We also provide a comparison between `CEM` and a classical generative `EM` semi-supervised algorithm (Nigam et al, 2000), and between our discriminant algorithms and the transductive SVM algorithm used for semi-supervised learning in (Joachims, 1999).

[4] the point where precision and recall (defined as the PGC of the relevant class) are equal.

7.4 Results

Generative classifiers Let us start this comparison with generative classifiers. We will first use naive Bayes as baseline and three semi-supervised methods: the classical generative `EM` semi-supervised (Nigam et al, 2000), the generative semi-supervised `CEM` (Algorithm 1) and the generative `semi-supervised CEM` with label-error modeling (Algorithm 3). We compare the performance of these three algorithms on 5 cross validation runs on the `Mushroom` and `7sectors` data sets, using a fixed proportion of labeled-unlabeled data on the training set for learning. For each dataset and each cross validation run, 25% of the examples are held aside as a test set. Results for the `Mushroom` and `7sectors` datasets appear respectively in Table 2 and Table 3.

For both datasets, we used a fixed proportion of labeled-unlabeled data in the training set. In the results described here, this proportion is $5\% - 95\%$ for `Mushroom` and $1\% - 99\%$ for `7sectors`, but similar conclusions also hold for any other proportion. In both cases, the same behavior is observed. Using unlabeled data considerably increases the performance (+36.4 for Mushrooms and +8.9 for 7sectors) and the error-label model allows for an additional increase compared to generative `semi-supervised CEM` (+4.6%) for `Mushroom` and (+3.6%) in mean for `7sectors`. Note that for `7sectors` we used only a very small number of labeled examples. The performance increase tends to be lower for smaller datasets such as `energy` or `utilities`. For the `Mushroom` database, the performance increase with the error model is more than 40% compared to the baseline naive Bayes classifier. Let us now examine the algorithms

Table 2. Percentage of good classification for the supervised NB classifier and three generative semi-supervised algorithms: generative `semi-supervised EM`, generative `semi-supervised CEM` and generative semi-supervised `CEM` with label error modeling on the UCI `Mushroom` data set using $5\% - 95\%$ as the proportion of labeled-unlabeled data on the training set.

Learning Algorithms	NB	Semi-sup. EM	Semi-sup. CEM	Gen. Semi-Sup. CEM lab-err.
PGC(%)	48.5 ± 5	85.8 ± 2	86.1 ± 3	90.7 ± 2

performance evolution when the proportion of labeled-unlabeled data in the training set is varied. Figures 1 and 2 respectively show performance on the test sets for `Email spam` and `Cmp_lg` collections as a function of the proportion of labeled data in the training set. On the x-axis, 5% means that 5% of labeled data in the training set were used for training, the 95% remaining being used as unlabeled training data. Each experiment was carried out on twenty paired trials of randomly selected training-test splits. On the y-axis, each point thus represents the mean performance for the 20 runs and the error bars correspond to the standard deviation for the estimated performance (Tibshirani, 1996). For both datasets, attribute values are on a continuous scale. For the generative methods, data were then assumed to be drawn from two normal populations. Performance curves (Figure 1 - top for `Email spam` and Figure 2 for `Cmp_lg`) of the generative semi-supervised learning algorithms (generative `semi-supervised CEM` and generative semi-supervised `CEM` with label error modeling) confirm the conclusions obtained on the previous datasets for all labeled-unlabeled data proportions. One also may observe that `semi-supervised CEM` and `semi-supervised EM` behave similarly and that a significant performance increase is reached for all labeled-unlabeled proportions when using the error model.

Table 3. Break even point on the `7sectors` data set for Naive Bayes and for the three generative semi-supervised algorithms (generative `semi-supervised EM`, `semi-supervised CEM` and generative semi-supervised with label error modeling) using $1\% - 99\%$ as the proportion of labeled-unlabeled data on the training set.

	Algorithms	energy	financ.	health	techn.	transp.	utilities
basic	NB sup.	57.1 ± 3.1	78.4 ± 4.3	83.8 ± 2.5	77 ± 2.8	79.2 ± 3.4	56.1 ± 2.1
	Semi-sup. EM	64.9 ± 1.8	87.1 ± 1.9	85.7 ± 1.5	84.5 ± 1.4	81.6 ± 1.8	60.8 ± 1.8
	Semi-sup. CEM	65.2 ± 2.3	87.3 ± 2.8	85.9 ± 1.9	84.7 ± 1.8	82.9 ± 1.9	59.9 ± 1.7
	Semi-sup. CEM imper.	**68.8±1.2**	**93.4±.7**	**87.6±.9**	**87.5±1.2**	**86.3±1.4**	**62.6±.8**
energy	NB sup.	-	73.2 ± 2.7	65.5 ± 2.4	53.3 ± 2.5	66.2 ± 2.1	73 ± 2.4
	Semi-sup. EM	-	80.1 ± 1.5	80.7 ± 1.3	60.5 ± 1.4	70.6 ± 1.1	81.9 ± 1.5
	Semi-sup. CEM	-	82.1 ± 1.4	80.8 ± 1.2	60.9 ± 1.6	74.1 ± 1.3	82.8 ± 1.2
	Semi-sup. CEM imper.	-	**85.9±1.3**	**82.1±.9**	**66.7±1.2**	**79.7±1.5**	**86.2±.8**
financ	NB sup.	-	-	67±2.1	88.7±1.8	78.3±2.3	53.6±3.2
	Semi-sup. EM	-	-	73.7 ± 1.1	91.1 ± 1.1	87.6 ± 1.4	56.4 ± 1.3
	Semi-sup. CEM	-	-	74.9 ± 1.5	$90.9 \pm .7$	$87.8 \pm .9$	57.9 ± 1.6
	Semi-sup. CEM imper.	-	-	**82±.8**	**92.8±.6**	**91.9±.5**	**62.9±1.3**
health	NB sup.	-	-	-	85 ± 1.3	79 ± 1.5	70.9 ± 1.8
	Semi-sup. EM	-	-	-	86.5 ± 1.8	85.6 ± 1.8	79.4 ± 1.2
	Semi-sup. CEM	-	-	-	87.9 ± 1.1	85.1 ± 1.4	78.9 ± 1.5
	Semi-sup. CEM imper.	-	-	-	**88.6±.9**	**89.8±1.2**	**81.6±1.3**
techn	NB sup.	-	-	-	-	78.9 ± 2.1	80.4 ± 1.5
	Semi-sup. EM	-	-	-	-	86.6 ± 1.5	85.4 ± 1.3
	Semi-sup. CEM	-	-	-	-	87.1 ± 1.4	84.5 ± 1.6
	Semi-sup. CEM imper.	-	-	-	-	**90.5±.9**	**87.3±1.1**
transp.	NB sup.	-	-	-	-	-	51.2 ± 3.2
	Semi-sup. EM	-	-	-	-	-	69.4 ± 1.8
	Semi-sup. CEM	-	-	-	-	-	69.9 ± 1.7
	Semi-sup. CEM imper.	-	-	-	-	-	**72.3±1.3**

Discriminant classifiers The same tests have been performed with three discriminant algorithms: a baseline supervised logistic classifier and two semi-supervised algorithm (`CEM`-discriminant (Algorithm 2) and discriminant semi-supervised `CEM` with label-error modeling (Algorithm 4)). Results are shown respectively on Figure 1 - bottom and Figure 2 for `Email spam` and `Cmp_lg` .

Comparing these three algorithms, it can be seen that as for the generative case, semi-supervised training allows for a considerable performance increase for all labeled-unlabeled training data proportions. The label-error model also offers for both datasets and all proportions a significant increase. For example, if we consider the text summarization task (Figure 2), using only 5% of the labeled sentences, the discriminant semi-supervised algorithm `CEM` with error model allows to increase performance by 12% compared to a fully supervised logistic classifier trained on 5% labeled data. 40% labeled data are needed to reach the same performance level with the baseline logistic method. At 5% labeled data the error model increases the performance of the discriminant semi-supervised algorithm by about 10%. This increase is lower when the proportion of labeled data is increased but remains consequent (about 5% on the average precision for 10% labeled data on Figure 2). The discriminant semi-supervised algorithm with label-error modeling thus provides an important performance increase compared to both the discriminant baseline and `CEM`-discriminant especially when there are only few labeled data available for training which is the most interesting situation in semi-supervised learning.

Discriminant vs generative A second observation is that discriminant training clearly outperforms generative training on both datasets for all proportions. On Figure 1-bottom and Figure 2, the curve for the best generative model (generative semi-supervised `CEM` with label error modeling) is below that of the semi-supervised discriminant classifiers with and without label-error model. It can be seen that maximal performance, obtained with 100% labeled training data for supervised learning, can be reached much sooner when using the discriminant semi-supervised `CEM` with label-error modeling (about 20% labeled data for `Email spam` and 50% for `Cmp_lg`).

EM vs CEM `EM` and `CEM` do have a similar behavior in the context of generative semi-supervised learning. Experiments performed with a generative semi-supervised `EM` algorithm (Nigam et al, 2000) and the generative semi-supervised `CEM` led to similar results. Table 2 gives the performance of `EM` and `CEM` for the mushroom dataset and they are indeed similar. The major interest and advantage of `CEM` here is that, besides the usual generative approach it also provides a natural framework for introducing discriminant semi-supervised algorithms which is not the case for `EM`. `CML` and `CEM` thus allow to express a whole family of generative and discriminant semi-supervised methods. As seen in section 5, semi-supervised discriminant `CEM` (Algorithm 2) optimizes simultaneously cross-entropy for labeled data and classification likelihood for unlabeled data, thus providing a link between classification and clustering.

Why a simple logistic classifier for semi-supervised learning ? All the algorithms introduced here can be instantiated with different density estimator functions or discriminant classifiers. We also performed tests with different discriminant classifiers instead of the logistic classifier used here. Linear regression gave slightly lower results. More complex non linear methods did not improved performance wrt logistic regression. It seems natural for semi-supervised learning to choose low variance classifiers like those used in the experiments reported here. There are many arguments for that. When using small amounts of labeled

data, it is likely that complex methods will not be able to capture the intrinsic non linearity of data by lack of information. Complex classifiers could then produce high variance error terms. Also for most real problems, the decision frontier for small amounts of labeled data is likely to be nearly linear and using unlabeled data will not bring any evidence for learning non linear frontiers in this case. However the complexity issue for the classifiers used in semi-supervised classification is still to be investigated. We have also compared the `CEM` algorithms introduced here with with the transductive SVM method (Joachims, 1999). Average performance for this transductive SVM is similar to the semi-supervised discriminant logistic with no error-model introduced in this paper (Figure 1 - bottom and Figure 2). However, in our experiments, the variance for the different runs was much higher for this transductive SVM than for the semi-supervised logistic classifier especially when only few labeled data are used which is the interesting case. There is no fundamental reason for that, but this has been observed in all our experiments and for different parameter settings of the SVM algorithm. In all the experiments, transductive SVM is below the discriminant and label-error model algorithm.

To summarize the conclusions, for all datasets and for both generative and discriminant methods, semi-supervised training allowed for an important performance increase compared to supervised training on the same amount of labeled data. Discriminant training with the logistic classifier clearly outperforms generative training (either naive Bayes or gaussian mixture). Training with a label-error model provides a significant additional performance increase in all cases for the above experiments. The combination of discriminant `CEM` and label-error training appears as a powerful semi-supervised method.

These conclusions are supported by strong experimental evidence on a variety of datasets and experimental conditions. Our results are algorithmic and experimental. There remains to develop a theoretical framework for semi-supervised learning which could help explaining the observed behavior. Up to now, the few theoretical results we know of have been obtained under very strong hypothesis and cannot be used as a basis for explaining the observed phenomena or analyzing the value of unlabeled data on practical data sets.

8 Conclusion

We have proposed a new family of discriminant algorithms for semi-supervised learning. These methods have been introduced using the `CML-CEM` formalism. This extension to the classical generative setting of `CML-CEM` shows that `CML-CEM` can be used as a general framework for describing both generative and discriminant semi-supervised methods. We have also proposed to learn a label-error model for the tentative labels iteratively computed for unlabeled data.

The models have been tested on a variety of classification problems using different datasets. The combination of discriminant training and error modeling proved particularly efficient. In all cases, empirical evidence clearly supports the validity of the proposed ideas. It is remarkable that the same algorithm behavior has been observed for all datasets and experimental conditions. In particular, the respective ranking of the different methods is similar in all cases. This provides additional support for the conclusions which can be drawn from this set of experiments. Additionally, the proposed models are simple and easy to implement. Of course, there remain many open problems for assessing when and how semi-supervised learning should be used and for predicting or measuring the importance of unlabeled samples for learning.

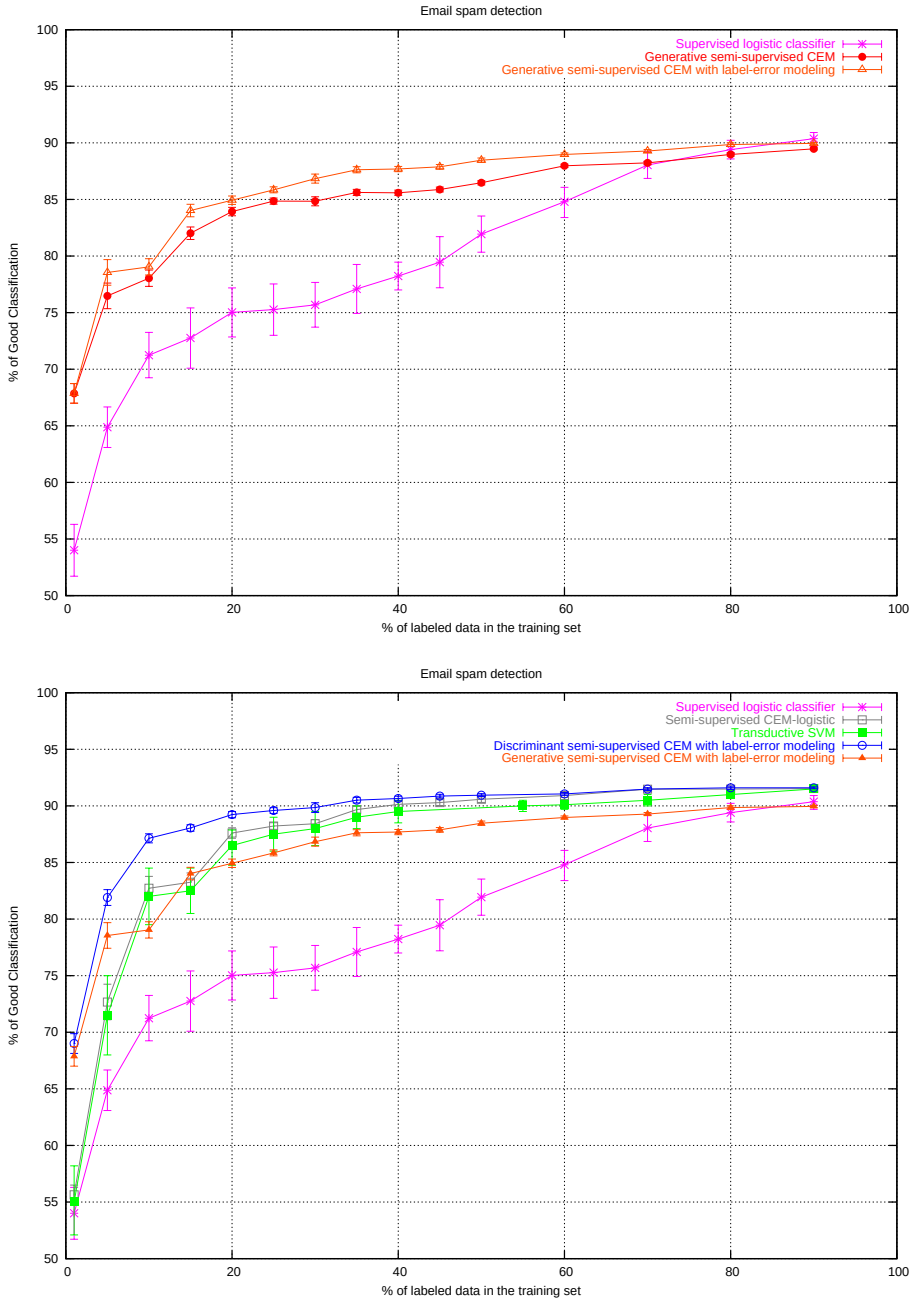

Fig. 1. Performance curves for the `Email spam` dataset comparing the baseline generative `semi-supervised CEM` (circle) and the generative semi-supervised CEM with label-error modeling (triangle) with the fully supervised logistic classifier (star) - top . The baseline semi-supervised Logistic-CEM algorithm (empty square), the transductive SVM (full square), the discriminant semi-supervised `CEM` algorithm with label-error modeling (empty circle) and the generative semi-supervised CEM with label-error modeling (full triangle) - down. Each point represents the mean performance for 20 arbitrary runs. The error bars show standard deviations for the estimated performance.

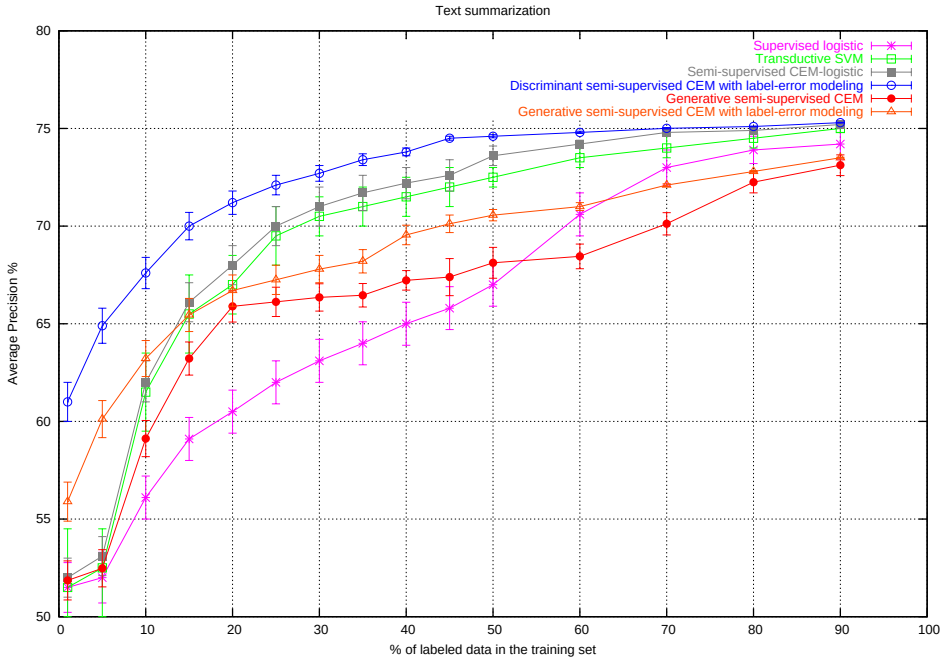

Fig. 2. Performance curves for the `Cmp_lg` dataset, for a logistic classifier trained in a fully supervised scheme (star), the baseline semi-supervised Logistic-CEM algorithm (full square), the transductive SVM (empty square), the discriminant semi-supervised `CEM` algorithm with label-error modeling (empty circle), the baseline generative `semi-supervised CEM` (full circle) and the generative semi-supervised `CEM` with label-error modeling (empty triangle). Each point represents the mean performance for 20 arbitrary runs. The error bars show standard deviations for the estimated performance.

References

Aitchison, J., Begg, C.B. (1976) Statistical diagnosis when basic cases are not classified with certainty. Biometrika 63(1): 1–12

Aitchison, J. ((1986) The statistical analysis of compositional data. Chapman and Hall, London.

Ambroise, C., Govaert, G. (2000) EM for partially known labels. Proceedings of the 7^{th} International Federation of Classification Societies Namur, Belgium, pp 161–166

Amini, M.R., Gallinari, P. (2003) Semi-supervised Learning with Explicit Misclassification Modeling. Proceedings of the 18^{th} International Joint Conference on Artificial Intelligence Acapulco, Mexico, pp 555–560

Amini, M.R., Gallinari, P. (2002) The use of unlabeled data to improve supervised learning for text summarization. Proceedings of SIGIR-02, 25^{th} ACM International Conference on Research and Development in Information Retrieval Tampere, Finland, pp 105–112

Anderson, J.A. (1979) Multivariate logistic compounds. Biometrika 66(1): 17–26

Anderson, J.A. (1982) Logistic discrimination. Handbook of statistics, P.R. Krishnaiah and L. Kanal (eds.) 2: 169–191

Banko M., Mittal V., Kantrowitz M., Goldstein J. (1999) Generating extraction-based summaries from hand-written done by text alignment. Proceedings of the Pacific Association for Computational Linguistics

Basu S., Banerjee A., Mooney R. (2002) Semi-supervised Clustering by seeding. Proceedings of the 19^{th} International Conference on Machine Learning Sydney, Australia, pp 19–26.

Belkin M., Niyogi P. (2004) Semi-supervised Learning on Riemannian Manifolds. Machine Learning 56: 209–23

Bennett K.P., Demiriz A. (1999) Semi-supervised support vector machines. Advances in Neural Information Processing Systems 11 Denver, USA, pp 368–374.

Blake C.L., Merz C.J.: UCI Repository of machine learning databases, url = "http://www.ics.uci.edu/∼mlearn/MLRepository.html. University of California, Irvine, Dept. of Information and Computer Sciences (1998).

Blum A., Mitchell T. (1998) Combining labeled and unlabeled data with Co-training. Proceedings of the Workshop on Computational Learning Theory, Madison, Wisconsin, USA, pp 92–100.

Castelli V., Cover T.M. (1996) The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. IEEE Transactions on Information Theory, 42(6): 2102–2117.

Celeux G., Govaert G. (1992) A Classification EM algorithm for clustering and two stochastic versions. Computational Statistics and Data Analysis, 14(3): 315–332

Chapelle O., Zien A.: Semi-Supervised Classification by Low Density Separation. AI & Statistics, (2005).

Chhikara R.S., McKeon J. (1992) Linear discriminant analysis with misallocation in training samples. Journal of the American Statistical Assocation, 79: 899–906.

Chittineni C.B. (1982) Learning with imperfectly labeled patterns. Pattern Recognition, 12(5): 169–191.

Collins M., Singer Y. (1999) Unsupervised models for named entity classification. Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, Maryland, USA, pp 189–196.

Cozman F.G., Cohen I., Cirelo M.C. (2003) Semi-supervised Learning of Mixture Models. Proceedings of the 20^{th} International Conference on Machine Learning, Washington DC, USA, pp 99–106.

DeSa V.R. (1994) Learning classification with unlabeled data. Advances in Neural Information Processing Systems 6, Denver, USA, pp 112–119.

Dempster A., Laird N., Rubin D. (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of Royal Statistical Society, 39(1): 1–38.

Jaakkola T., Meila M., Jebara T. (2000) Maximum entropy discrimination. Advances in Neural Information Processing Systems 12, Denver, USA, pp 470–477.

Joachims T. (1999) Transductive inference for text classification using support vector machines. Proceedings of the 16^{th} International Conference on Machine Learning, Bled, Slovenia, pp 200–209.

Krishnan T., Nandy S.C. (1987) Discriminant analysis with a stochastic suprevisor. Pattern Recognition, 20(4): 379–384.

Krishnan T. (1988) Efficiency of learning with imperfect supervision. Pattern Recognition, 21: 183–188.

Kupiec J., Pederson J., Chen F.A. (1995) Trainable Document Summarizer. Proceedings of SIGIR-95, 18^{th} ACM International Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, pp 68–73.

Lawrence N.D., Schölkopf B. (2001) Estimating a kernel Fisher discriminant in the presence of label noise. Proceedings of the 18^{th} International Conference on Machine Learning, Massachusetts, USA, pp 306–313.

Lachenbrunch P.A. (1974) Discriminant functions when the initial samples are misclassified. II. Nonrandom misclassification models. Technometrics, 16: 419–424.

McLachlan G.J. (1972) Asymptotic results for discriminant analysis when the initial samples are misclassified. Technometrics, 14: 415–422.

McLachlan G.J., Ganesalingam S. (1982) Updating a discriminant function in basis of unclassified data. Communication on Statistics-Simulation and Computation, 11(6): 753–767.

McLachlan G.J. (1992) Discriminant Analysis and Statistical Pattern Recognition. John Wiley & Sons, New York

Miller D., Uyar H. (1997) A mixture of experts classifier with learning based on both labeled and unlabeled data. Advances in Neural Information Processing Systems 9, Denver, USA, pp 571–577.

Mladenic D., Grobelnik M. (1998) Feature selection for classification based on text hierarchy. Working Notes of Learning from Text and the Web, Conference Automated Learning and Discovery, Carnegie Mellon University, Pittsburgh.

Murray G.D., Titterington D.M. (1978) Estimation problems with data from a mixture. Applied Statistics, 27(3): 325–334.

Muslea I., Minton S., Knoblock C. (2002) Active + semi-supervised learning = robust multi-view learning. Proceedings of the 19^{th} International Conference on Machine Learning, Syndey, Australia, pp 435–442.

Nigam K., McCallum A.K., Thrun S., Mitchell T. (2000) Text classification from labeled and unlabeled documents using EM. Machine Learning, 39(2/3): 127–163.

O'Neill, T.J. (1978) Normal discrimination with unclassified observations. Journal of the American Statistical Association, 73(364): 821–826.

Ratsaby J., Venkatesh S.S. (1995) Learning from a mixture of labeled and unlabeled examples with parametric side informaion. Proceedings of the 8^{th} Annual Conference on Computational Learning Theory, Santa Cruz, California, USA, pp 412–417.

Roth V., Steinhage V. (1999) Nolinear discriminant analysis using kernel functions. Advances in Neural Information Processing Systems 12, Denver, USA, pp 568–574.

Seeger M.: Learning with labeled and unlabeled data. University of Edinburgh, (2000) Technical report. www.dai.ed.ac.uk/homes/seeger/papers/review.pdf.

SUMMAC: TIPSTER Text Summarization Evaluation Conference. http://ww-nlpir.nist.gov/related_projects/tipster_summac/, (1998).

Szummer M., Jaakkola T. (2001) Kernel expansions with unlabeled examples. Advances in Neural Information Processing Systems 13, Vancouver, British Columbia, Canada, pp 626–632.

Symons M.J. (1981) Clustering criteria and multivariate normal mixture. Biometrics, 37(1): 35–43.

Tibshirani R. (1996) A comparison of some error estimates for neural networks models. Neural Computation, 8: 182–163.

Titterington D.M. (1989) An alternative stochastic supervisor in discriminant analysis. Pattern Recogntion, 22(1): 91–95.

Vittaut J.N., Amini M.R., Gallinari P. (2002) Learning classification with both labeled and unlabeled data. Proceedings of the 13^{th} European Conference on Machine Learning, Helsinki, Finland, pp 468–479.

Zhu X., Ghahramani Z., Lafferty J. (2003) Semi-supervised learning using Gaussian fields and harmonic functions. Proceedings of the 20^{th} International Conference on Machine Learning, Washington DC, USA, pp 912–919.

9 Appendix

We provide in this appendix the reestimation formulas for the label-error model, the generative and discriminant CEM algorithms 2, 3 and 4. For simplifying the notations, we consider here the two classes classification case, extension to more than two classes is straightforward.

9.1 Reestimation of the label-error parameters

According to the constraint (10), the parameters of the label-error model, in the case of a 2-class classification problem, depend on two factors, say α_{12} and α_{21}. Let us introduce the following notation:

$$\begin{aligned} T_{1i} &= \hat{t}_{1i}\ (1-\alpha_{21})\frac{p(x_i,\widetilde{y}=1)}{p(x_i,\hat{y}=1)} + \hat{t}_{2i}\ \alpha_{21}\frac{p(x_i,\widetilde{y}=1)}{p(x_i,\hat{y}=2)} \\ T_{2i} &= \hat{t}_{1i}\ \alpha_{12}\frac{p(x_i,\widetilde{y}=2)}{p(x_i,\hat{y}=1)} + \hat{t}_{2i}\ (1-\alpha_{12})\frac{p(x_i,\widetilde{y}=2)}{p(x_i,\hat{y}=2)} \end{aligned} \tag{25}$$

From (13), we have:

$$\forall i, T_{1i} + T_{2i} = 1 \tag{26}$$

Differentiating (15) with respect to these parameters, equating to zero and using (26) leads to the following results:

$$\begin{aligned} \Gamma_1 &= \sum_{i=n+1}^{n+m} T_{1i} = \sum_{i=n+1}^{n+m} \hat{t}_{2i}\ \frac{p(x_i,\widetilde{y}=1)}{p(x_i,\hat{y}=2)} \\ \Gamma_2 &= \sum_{i=n+1}^{n+m} T_{2i} = \sum_{i=n+1}^{n+m} \hat{t}_{1i}\ \frac{p(x_i,\widetilde{y}=2)}{p(x_i,\hat{y}=1)} \end{aligned} \tag{27}$$

From (27), the maximum likelihood estimates of the label error model parameters, α_{12} and α_{21}, are given by

$$\begin{aligned} \alpha_{21} &= \frac{1}{\Gamma_1}\sum_{i=n+1}^{n+m} \hat{t}_{2i} \times \frac{1}{1+\frac{(1-\alpha_{12})}{\alpha_{21}}\frac{\pi_2 f_2(x_i,\theta_2)}{\pi_1 f_1(x_i,\theta_1)}} \\ \alpha_{12} &= \frac{1}{\Gamma_2}\sum_{i=n+1}^{n+m} \hat{t}_{1i} \times \frac{1}{1+\frac{(1-\alpha_{21})}{\alpha_{12}}\frac{\pi_1 f_1(x_i,\theta_1)}{\pi_2 f_2(x_i,\theta_2)}} \end{aligned} \tag{28}$$

9.2 Reestimation for the generative CEM model with a label error model (Algorithm 3)

We used two density models for our experiments: a naive Bayes model for discrete data and a gaussian model for real valued data.

Discrete-valued data: Naive-bayes model We examine here parameter estimation for a Naive-Bayes classifier. Each example is described by a d-dimensional discrete vector $x_i = \langle n(j,x_i)\rangle_{j\in\{1,\ldots,d\}}$. We suppose that observations are generated independently from a mixture density (4) with parameters Θ; the set of class priors π_k and binomials p_{jk} is:

$$\Theta = \{\pi_k; p_{jk}; k \in \{1,2\}, j \in \{1,...,d\}\} \tag{29}$$

By making the feature independence assumptions of the naive Bayes model, we further assume that the length of each example is identically distributed. In this case, the pdf for each class writes $f_k(x) \equiv \prod_{j=1}^{d} p_{jk}^{n(j,x)}$. Differentiating (15) in turn with respect to π_k and p_{jk}, using (25) and Lagrange multipliers to enforce the constraints $\sum_k \pi_k = 1$ and $\forall k, \sum_j p_{jk} = 1$, we get the maximum likelihood estimates of π_k and p_{jk}:

$$\pi_k = \frac{\sum_{i=1}^{n} t_{ki} + \sum_{i=n+1}^{n+m} T_{ki}}{n+m} \tag{30}$$

$$p_{jk} = \frac{\sum_{i=1}^{n} t_{ki}\ n(j,x_i) + \sum_{i=n+1}^{n+m} T_{ki}\ n(j,x_i)}{\sum_{i=1}^{n} t_{ki}\ |x_i| + \sum_{i=n+1}^{n+m} T_{ki}\ |x_i|} \tag{31}$$

Where, $|x_i| = \sum_{j=1}^{d} n(j,x_i), \forall i$. Note that the discriminant function in this case is linear in the observations.

Real-valued data: Normal case We now consider the case of two d-dimensional normal populations with a common covariance matrix Σ, $\mathcal{N}_d(\mu_1, \Sigma)$ and $\mathcal{N}_d(\mu_2, \Sigma)$, occurring respectively in proportions π_1 and π_2. As previously, the maximum likelihood estimates of the mixing parameters π_k are given by equation (30). Now, let introduce μ_1, μ_2 and Σ in equation (15):

$$L_c(C,\Theta,\Lambda) = const. + \sum_{k=1}^{2}\sum_{i=1}^{n} t_{ki} \log \pi_k - \frac{n+m}{2} \log|\Sigma| + \tag{32}$$
$$\sum_{i=1}^{n}\sum_{k=1}^{2} t_{ki} \left[-\frac{1}{2}(x_i-\mu_k)^t \Sigma^{-1}(x_i-\mu_k)\right] + \sum_{i=n+1}^{n+m}\sum_{k=1}^{2}\left[\hat{t}_{ki} \log\left(\sum_{k=1}^{2} \alpha_{kh}\pi_h\zeta_{hi}\right)\right]$$

where, $\forall x_i \in D_u, h \in \{1,2\}, \zeta_{hi} = exp\left[-\frac{1}{2}(x_i-\mu_h)^t\Sigma^{-1}(x_i-\mu_h)\right]$. Using notations from (25), the maximum likelihood estimates of $\mu_k, k \in \{1,2\}$ and Σ are:

$$\mu_k = \frac{\sum_{i=1}^{n} t_{ki}\ x_i + \sum_{i=n+1}^{n+m} T_{ki}\ x_i}{\sum_{i=1}^{n} t_{ki} + \sum_{i=n+1}^{n+m} T_{ki}},\ \forall k \in \{1,2\}$$

$$\Sigma = \frac{1}{n+m}\sum_{i=1}^{n}\sum_{k=1}^{2} t_{ki}(x_i-\mu_k)(x_i-\mu_k)^t +$$
$$\frac{1}{n+m}\sum_{i=n+1}^{n+m}\sum_{k=1}^{2} T_{ki}(x_i-\mu_k)(x_i-\mu_k)^t \tag{33}$$

9.3 Logistic reestimation formulas (Algorithms 2 and 4)

As for the discriminant classifier we used a logistic classifier (Anderson, 1982) in the experiments. For the two classes case, the logistic regression model LR_B, reduces to a simple

logistic unit G whose parameters are $B = (\beta_0, \beta)$. The output of LR_B for an input x is $G(x) = \frac{1}{1+e^{-(\beta_0+\beta^t.x)}}$. In order to maximize L'_c, in the `M-step` of algorithms 2 and 4, we used a quasi-Newton gradient procedure. This requires to compute only the first derivatives of L'_c with regard to the classifier parameters.
For algorithm 2, estimates for the parameters B of the logistic classifier are:

$$\frac{\partial L'_c}{\partial \beta_j} = \sum_{i=1}^{n} \left[t_{1i} - G(x_i)\right] x_{ji} + \sum_{i=n+1}^{n+m} \left[\widetilde{t}_{1i} - G(x_i)\right] x_{ji},\ j \in \{0, ..., d\}$$

Where x_{ji} denotes the j^{th} feature of x_i. We suppose further that $\forall i \in \{1, ..., n+m\}, x_{0i} = 1$.

For Algorithm 4, both the classifier and the label-error model parameters are updated at each iteration in order to maximize (20). Let

$$Z_{ki} = p(\hat{y} = k \mid x_i) = \sum_{h=1}^{2} \{\alpha_{kh} p(\widetilde{y} = h \mid x_i)\}, \quad \forall i \in \{n+1, ..., n+m\}, k \in \{1, 2\} \tag{34}$$

The first partial derivatives of L'_c with respect to the parameters B of the logistic classifier and Λ of the label-error, are:

$$\frac{\partial L'_c}{\partial \beta_j} = \sum_{i=1}^{n} \left(t_{1i} - G(x_i)\right) x_{ji} +$$
$$\sum_{i=n+1}^{n+m} \left[\left(\frac{(1 - \alpha_{12} - \alpha_{21}) \times G(x_i) \times (1 - G(x_i))}{Z_{1i} \times Z_{2i}}\right) (\hat{t}_{1i} - Z_{1i})\ x_{ji}\right], \quad j \in \{0, ..., d\} \tag{35}$$

And,

$$\frac{\partial L'_c}{\partial \alpha_{kh}} = \sum_{i=n+1}^{n+m} \frac{p(\widetilde{y} = h \mid x_i, B)}{Z_{ki} \times Z_{hi}} \times (\widetilde{t}_{ki} - Z_{ki}), \quad (k, h) \in \{1, 2\}^2, k \neq h \tag{36}$$

Starting from an initial labeling, these approaches proceed by computing at each `E-step`, posterior probabilities of class membership for unlabeled observations given the current parameters of the model and update at the `M-step`, these parameters by maximizing the data log-likelihood, using the estimated posteriors from the previous step. Usually, for continuous variables, mixture components are assumed to be normal densities and for discrete variables, non-parametric techniques, such as histograms, are often used in practice.