Joint 48th IEEE Conference on Decision and Control and
28th Chinese Control Conference
Shanghai, P.R. China, December 16-18, 2009

Context-Dependent Multi-class Classification with Unknown Observation and Class Distributions with Applications to Bioinformatics

Alexander S. Baras and John S. Baras

Abstract— **We consider the multi-class classification problem, based on vector observation sequences, where the conditional (given class observations) probability distributions for each class as well as the unconditional probability distribution of the observations are unknown. We develop a novel formulation that combines training with the quality of classification that can be obtained using the 'learned' (via training) models. The parametric models we use are finite mixture models, where the same component densities are used in the model for each class, albeit with different mixture weights. Thus we use a model known as All-Class-One-Network (ACON) model in the neural network literature. We argue why this is a more appropriate model for context-dependent classification, as is common in bioinformatics. We derive rigorously the solution to this joint optimization problem. A key step in our approach is to consider a tight (provably) bound between the average Bayes error (the true minimal classification error) and the average model-based classification error. We rigorously show that the parameter estimates maximize the likelihood of the model-based class posterior probability distributions. We illustrate by application examples in the bioinformatics of cancer.**

I. Introduction

The identification of principal structures and features within large sets of high-dimensional data and the generation of simplified models for such data distributions are important tasks, which arise in many and diverse technical fields including pattern recognition [22], [15] or the study of complex physical systems. Various approaches have been proposed [22], [11], [19], [20]. In this paper we consider the multi-class classification problem, based on vector observation sequences, where the conditional (given class observations) probability distributions for each class as well as the unconditional probability distribution of the observations are *unknown*. We formulate the problem of training in classifying observations, using parametric models for the conditional pdf for each class, as well as the unconditional pdf of the observations. The parametric models we use for these pdfs are finite mixtures of normal densities [37], [2], [32], [21]. We develop a novel formulation that *combines training with the quality of classification* that can be obtained using the 'learned' models. Our approach differs from existing approaches in the literature in two fundamental ways.

First, the common approach in the literature [22] is to use labelled observations from each class to obtain a maximum likelihood estimate of each class conditional pdf, which is typically obtained via Expectation Maximization (EM) iterations [22], [11]. In the neural network literature this is known as *One-Class-One-Network (OCON)* model [22]. Then these estimated class conditional pdfs are used for classifying observations (data) using a variety of criteria (and methods) for obtaining optimal decisions (classifications). In our approach we formulate the problem of *optimal decision making* (i.e. learning the optimal decision rule) and *model parameter estimation* as a *joint optimization problem*. Second, the parametric models we use are finite mixture models, where the *same component densities* are used in the model for each class, albeit with different mixture weights. Thus we use a model known as *All-Class-One-Network (ACON)* model in the neural network literature [22].

We argue that the ACON model is a more appropriate model for context-dependent classification, as is common in bioinformatics [13], [1], [28], or in network security [38]. The advantages and disadvantages are discussed in [22]. In biology as well as in security (intrusion detection and classification) the patterns and classes are very dependent on the context of the data collection (observations). On the other hand in problems such as face recognition, or speech recognition, the signal patterns are much less sensitive to context, and in these cases the OCON model offers several advantages [22].

Discriminative criteria for classification and regression have attracted recent attention by the machine learning community. Examples include support vector machines [22], decision directed probabilistic neural networks [23], [25], [20], discriminatively trained HMMs [19], [26], [4], [29], predictively trained neural networks [23], [22]. These newer approaches optimize models and model parameters driven by the goal of better performing classifiers; the main task at hand. This is to be contrasted with more traditional methods where models and model parameters are optimally estimated using maximum likelihood (ML) or maximum aposteriori probability (MAP) so that each density is trained separately to escribe observations [22] rather than trained to aid in better classification. Naturally this affects performance adversely and at the same time increases the complexity of the classification algorithm, which in turn increases the possibility of overfitting. Some recent papers [22] have indeed substantiated these adverse effects on performance.

The problem considered here is the classification of C classes, using N samples of vector valued data, while using K component densities. These component densities are also known as 'features' in pattern recognition [22]. The number of components used in these approximations is a design

Alexander S. Baras is with the School of Medicine, Department of Pathology, University of Virginia Health System, Charlottesville, VA 22901.

John S. Baras is with the Institute for Systems Research, Department of Electrical and Computer Engineering, Fischell Department of Bioengineering, and the Applied Mathematics, Statistics and Scientific Computation Program, University of Maryland, College Park, MD 20742.

978-1-4244-3872-3/09/$25.00 ©2009 IEEE

variable to be selected as well; we describe our results on this aspect of the problem elsewhere [5]. In our approach, described in this paper, we derive rigorously the solution to this joint optimization problem. A key step in our approach is to consider a tight (provably) bound between the average Bayes error (the true minimal classification error) and the average model-based classification error. The bound is given in terms of the average (with respect to observations) of the Kullback-Leibler divergence (or relative entropy) of the true (but unknown) class a-posteriori distribution given the measurements and the approximate (model-based) class a-posteriori distribution given the measurements.

We minimize this bound to determine the optimal parametric model for each class that obtains the best approximation to the average Bayes error. We rigorously show that the parameter estimates maximize the likelihood of the *model-based class posterior probability distributions*. This is different from the common approach where first the model parameters are estimated by maximizing the likelihood of the class conditional probability distribution. Our approach is closer to the more recently developed methods for *discriminative training* [20], [26], [2], [14], *conditional maximum likelihood* [27], [14], [33], [18], or *maximum mutual information* [27], [12], [7], [4], albeit in a multi-class, multi-component setting.

As discriminative model estimation and conditional likelihood maximization have attracted more attention recently, the need for developing similar lower bounding and maximization, two step algorithms (i.e. EM-like) has increased. Some developments towards this end have recently appeared in [18], [9], [33]. Using properly constructed upper and lower bounds we have derived [6] an 'EM-like' algorithm for computing these 'most discriminating' model parameter estimates.

II. Preliminaries, Notation, Models

The formulation is as follows. We have N data samples, $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_t, ..., \boldsymbol{x}_N$ all vectors in $\mathcal{R}^D$. These data points can come from C distinct classes. These data points together with the class index Y_t, $t = 1, 2, ..., N$, for each, constitute the *training data* in our framework. That is Y_t is a categorical (discrete) variable with values in $\{1, 2, ..., c, ..., C\}$. Thus we let the training set be denoted as

$$\mathcal{S} = \{(\boldsymbol{x}_t, y_t);\ t = 1, 2, \ldots, N\}. \qquad (1)$$

When data come from class c , we will denote by $p^c_{\boldsymbol{X}}(.)$ the class-conditional probability distribution of the random variable $\boldsymbol{X}$. We will also use the notation $p_{\boldsymbol{X}|c}(.)$ for the same pdf. We will give to almost all variables a probabilistic interpretation, and then we will use the more generic notations $p_{\boldsymbol{X}_t|Y_t}(.|.)$ and $p_{\boldsymbol{X}_t|Y_t}(\boldsymbol{x}_t|y_t)$. The *true probability distribution functions* $p_{\boldsymbol{X}}(.)$ and $p_{\boldsymbol{X}|c}(.|.)$ are *unknown*. We will use approximate models for both from the class of *finite normal mixture models*. Thus we will use the model

$$p_{\boldsymbol{X}|Y}(\boldsymbol{x}|y;\boldsymbol{\theta}) = p_{\boldsymbol{X}|Y}(\boldsymbol{x}|c;\boldsymbol{\theta}) = \sum_{k=1}^{K} \alpha_{c,k} G(\boldsymbol{x};\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where $G(\boldsymbol{x};\boldsymbol{\mu}_k, \boldsymbol{Sigma}_k)$ is a multivariable (D - dimensional) normal (Gaussian) probability distribution with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. These Gaussians constitute the *component distributions* (or simply the *components*) of the finite mixture model. In our formalism above there are K components, *the same for all classes*, consistent with our use of the *ACON* model. The weights $\alpha_{c,k}$ represent the '*mixture coefficients*' in the model, and satisfy the constraints

$$0 \le \alpha_{c,k} \le 1\,, \text{and } \sum_{k=1}^{K} \alpha_{c,k} = 1, \text{ for all } c.$$

We will also use the notation whereby the coefficients $\alpha_{c,k}$ for the same class c are collected into a vector $\boldsymbol{\alpha}_c$. We collect these coefficients into a $C \times K$ matrix $\boldsymbol{\mathcal{A}}$; a *row-stochastic* matrix.

In this finite mixture model $\boldsymbol{\theta}$ represents the vector of parameters of this parametric class of models

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\mu}_k; & k = 1, 2, \ldots, K \\ \boldsymbol{\Sigma}_k; & k = 1, 2, \ldots, K \\ \alpha_{c,k}; & k = 1, 2, \ldots, K; c = 1, 2, \ldots, C \end{bmatrix}. \qquad (2)$$

We let $\pi_c, c = 1, 2, ..., C$, be the prior probabilities that data belong to class c. Then the model we use to approximate the unknown probability distribution of the data $\boldsymbol{X}$ is

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \sum_{c=1}^{C} p_{\boldsymbol{X}|c}(\boldsymbol{x};\boldsymbol{\theta})\pi_c = \sum_{c=1}^{C}\sum_{k=1}^{K} \pi_c \alpha_{c,k} G(\boldsymbol{x};\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

The data vectors $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_t, ..., \boldsymbol{x}_N\}$ are considered as independent and identically distributed samples from the unknown probability distribution $p_{\boldsymbol{X}}(.)$. In the classification literature these data are referred to as the *incomplete data* [22]. On the other hand the set of data vectors together with their classes $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_t, y_t), ..., (\boldsymbol{x}_N, y_N)\}$ are referred to as the *complete data* [22].

III. ML and EM for Models with Hierarchical Structure

In this section we formulate and solve the ML estimation problem of the parameter vector $\boldsymbol{\theta}$ when we have a hierarchical structure, i.e. we have both classes and components. We let $\boldsymbol{x}_t$ denote the generic D- vector of measurements, y_t denote the generic categorical value indicating the class of the experimental sample, z_t denote the generic categorical value indicating the component of the experimental sample. Figure 1 illustrates the hierarchical mixture model with two classes and three components.

The training data $\mathcal{S}$ are used to "learn" a good *classification rule*, which will be applied to experimental data from similar situations as the training data, but which have not been used in the training [22], [15]. One way to learn a good classification rule is to learn a good set of the model parameters, for the models we use to represent the classes, and then use the learned models in constructing a good classification rule. In this section we solve the problem following this approach. The novelty of the results rests

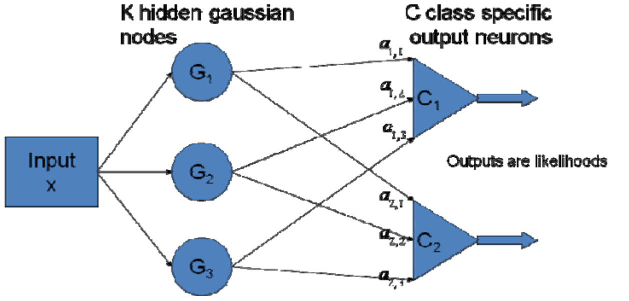

Fig. 1. Illustrating a hierarchical mixture model with $K = 3$ and $C = 2$.

with the derivation of the algorithms using only convexity arguments and bounds, and with the treatment of the ACON model.

Omitting some details, the likelihood function for the entire set of training data, is

$$\mathcal{L}(\boldsymbol{\theta};\boldsymbol{x}_1^N,y_1^N)) = \sum_{t=1}^{N}\log p_{\boldsymbol{X}_t,Y_t}(\boldsymbol{x}_t,y_t;\boldsymbol{\theta})$$
$$= \sum_{t=1}^{N}\{\log\sum_{k=1}^{K}\{\alpha_{y_t,k}G(\boldsymbol{x}_t;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\}+\log\pi_{y_t}\} \quad (3)$$

Thus the ML parameter estimation problem becomes:

$$\hat{\boldsymbol{\theta}}(\boldsymbol{x}_1^N,y_1^N) = \arg\max_{\boldsymbol{\theta}}\sum_{t=1}^{N}\log\{\sum_{k=1}^{K}\{\alpha_{y_t,k}G(\boldsymbol{x}_t;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\}$$

Our solution of this optimization problem uses convex analysis [8] and does not utilize differentials. The method has important implications for deriving an efficient EM-like algorithm (see [6] for details).

The EM algorithm is an iterative optimization technique specifically designed for probabilistic models. It uses a different strategy than gradient decent or Newton's method and sometimes provides faster convergence. A very insightful explanation of why EM has some key properties, including the property that it always computes a local maximum, can be given via lower-bound maximization. EM constructs a local approximation that is a lower bound to the objective function. The lower bound can have any functional form, in principle. Choosing the new estimate of the maximum as the maximizer of the lower bound will always be an improvement over the previous estimate, unless the gradient was zero there. So EM alternates between computing a lower bound (the 'E-step') and maximizing this lower bound (the 'M-step'), until a point where the gradient zero is reached. This two step approach for optimization is captured in its more general form by the following lemma.

Lemma 3.1: Let $f(\boldsymbol{\theta})$ be a scalar function of the vector valued variable $\boldsymbol{\theta}$, which is to be maximized over $\boldsymbol{\theta}$. Let $\boldsymbol{\theta}^{(n)}$ be a sequence of points, and $b_n(\boldsymbol{\theta})$ a sequence of scalar functions, such that: (i) b_n is a local lower bound for f around $\boldsymbol{\theta}^{(n)}$, that is $b_n(\boldsymbol{\theta}) \leq f(\boldsymbol{\theta})$ everywhere in a neighborhood of $\boldsymbol{\theta}^{(n)}$; (ii) $\boldsymbol{\theta}^{(n+1)}$ is the maximum of b_n in a neighborhood of $\boldsymbol{\theta}^{(n)}$ and (iii) the functions b_n and f touch at $\boldsymbol{\theta}^{(n)}$, that is $b_n(\boldsymbol{\theta}^{(n)}) \leq f(\boldsymbol{\theta}^{(n)})$. Then $f(\boldsymbol{\theta}^{(n+1)}) \geq f(\boldsymbol{\theta}^{(n)})$.

Proof: Follows immediately from the inequalities $f(\boldsymbol{\theta}^{(n+1)}) \geq b_n(\boldsymbol{\theta}^{(n+1)}) \geq b_n(\boldsymbol{\theta}^{(n)}) = f(\boldsymbol{\theta}^{(n)})$. ■

Thus progress is guaranteed through these iterations. It is not absolutely necessary to maximize the lower bound over $\boldsymbol{\theta}$. The so-called 'generalized EM' demonstrates that any improvement of the lower bound is sufficient.

The first step, in the development of the explicit iterations for the E-M algorithm, which will lead us to the E-step computation is to compute a lower bound to $\mathcal{L}(\theta\,;\,\mathbf{x}_1^N,\,y_1^N)$. We introduce an arbitrary distribution $q_{Z_1^N}$ over the component variables, over the set $\{1\,,\,2\,,...,\,K\}$, and using Jensen's inequality [10], [6] we obtain the lower bound

$$\mathcal{L}(\boldsymbol{\theta};\boldsymbol{x}_1^N,y_1^N) \geq B(\boldsymbol{\theta};q_{Z_1^N},\,\boldsymbol{x}_1^N,y_1^N)$$
$$= -\sum_{t=1}^{N}D\left\{q_{Z_t}(.)||p_{Z_t|\boldsymbol{X}_t,Y_t}(.|\boldsymbol{x}_t,y_t;\boldsymbol{\theta})\right\}$$
$$+\sum_{t=1}^{N}\log p_{\boldsymbol{X}_t|Y_t}(\boldsymbol{x}_t|y_t;\boldsymbol{\theta})+\sum_{t=1}^{N}\log p_{Y_t}(y_t) \quad (4)$$

The $q_{Z_1^N}$ maximizing this lower bound is clearly

$$\tilde{q}_{Z_1^N}(z_1^N) = \prod_{t=1}^{N}p_{Z_t|\boldsymbol{X}_t,Y_t}(z_t|\boldsymbol{x}_t,y_t;\boldsymbol{\theta})\,. \quad (5)$$

For the finite normal mixture of interest here

$$p_{Z_t|\boldsymbol{X}_t,Y_t}(z_t|\boldsymbol{x}_t,y_t;\boldsymbol{\theta}) = \frac{\alpha_{y_t,\,k}G\left(\boldsymbol{x}_t\,;\,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)}{\sum\limits_{k=1}^{K}\alpha_{y_t,\,k}G\left(\boldsymbol{x}_t\,;\,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)} \quad (6)$$

The corresponding maximum value of $B(\boldsymbol{\theta};q_{Z_1^N},\,\boldsymbol{x}_1^N,y_1^N)$, which is the 'best' lower bound to $\mathcal{L}(\boldsymbol{\theta};\boldsymbol{x}_1^N,y_1^N)$ at $\boldsymbol{\theta}$, is

$$\sum_{t=1}^{N}\log p_{\boldsymbol{X}_t|Y_t}(\boldsymbol{x}_t|y_t;\boldsymbol{\theta})+\sum_{t=1}^{N}\log p_{Y_t}(y_t) = \mathcal{L}(\boldsymbol{\theta};\boldsymbol{x}_1^N,y_1^N).$$

Thus conditions (ii) and (iii) of Lemma (3.1) are satisfied.

We next proceed to derive the E- and M-steps of the EM algorithm for hierarchical mixture models, following the convex optimization approach of [6], where we refer for details. We simplify notation and rewrite the bound (4) as

$$B(\boldsymbol{\theta};\tilde{q}_{Z_1^N},\,\boldsymbol{x}_1^N,y_1^N) = \hat{B}^{(n)}(\boldsymbol{\theta};\hat{\boldsymbol{\theta}}^{(n)},\,\boldsymbol{x}_1^N,y_1^N)$$
$$= \tilde{B}^{(n)}(\boldsymbol{\theta};p_{Z_1^N|\boldsymbol{X}_1^N,Y_1^N}(.|\boldsymbol{x}_1^N;\hat{\boldsymbol{\theta}}^{(n)}(\boldsymbol{x}_1^N,y_1^N)),\boldsymbol{x}_1^N,y_1^N)$$
$$= -\sum_{t=1}^{N}D\left\{p_{Z_t|\boldsymbol{X}_t,Y_t}(.|\boldsymbol{x}_t,y_t;\hat{\boldsymbol{\theta}}^{(n)})||p_{Z_t|\boldsymbol{X}_t,Y_t}(.|\boldsymbol{x}_t,y_t;\boldsymbol{\theta})\right\}$$
$$+\sum_{t=1}^{N}\log p_{\boldsymbol{X}_t|Y_t}(\boldsymbol{x}_t|y_t;\boldsymbol{\theta})+\sum_{t=1}^{N}\log p_{Y_t}(y_t) \quad (7)$$

The next step is to maximize $\hat{B}^{(n)}(\boldsymbol{\theta};\hat{\boldsymbol{\theta}}^{(n)},\boldsymbol{x}_1^N,y_1^N)$ (locally) w.r.t. $\boldsymbol{\theta}$ to obtain the next estimate (M-step) as

$$\hat{\boldsymbol{\theta}}^{(n+1)}(\boldsymbol{x}_1^N,y_1^N) = \arg\max_{\boldsymbol{\theta}}\hat{B}^{(n)}(\boldsymbol{\theta};\hat{\boldsymbol{\theta}}^{(n)},\boldsymbol{x}_1^N,y_1^N). \quad (8)$$

Applying Lemma 3.1 we obtain the desired increase in the log-likelihood. We decompose the best lower bound as follows

$$\hat{B}^{(n)}(\boldsymbol{\theta};\, \hat{\boldsymbol{\theta}}^{(n)}, \boldsymbol{x}_1^N, y_1^N) = Q^{(n)}(\boldsymbol{\theta}) + H^{(n)} + \log p_{Y_1^N}(y_1^N). \tag{9}$$

where $Q^{(n)}(\boldsymbol{\theta})$is the expected complete log-likelihood:

$$Q^{(n)}(\boldsymbol{\theta}) =$$
$$\sum_{t=1}^{N}\sum_{z_t=1}^{K} p_{Z_t|\boldsymbol{X}_t,Y_t}(z_t|\boldsymbol{x}_t, y_t; \hat{\boldsymbol{\theta}}^{(n)})\log p_{\boldsymbol{X}_t,\, Z_t|Y_t}(\boldsymbol{x}_t, z_t|y_t; \boldsymbol{\theta}),$$

and $H^{(n)}$ is the entropy of the distribution $p_{Z_1^N|\boldsymbol{X}_1^N,Y_1^N}(.\,|\boldsymbol{x}_1^N,\, y_1^N; \hat{\boldsymbol{\theta}}^{(n)}(\boldsymbol{x}_1^N, y_1^N))$.

The resulting EM algorithm for the ML estimate of the model parameter vector $\boldsymbol{\theta}$ given $(\boldsymbol{x}_1^N, y_1^N)$ can be summarized in the following two steps:

1) E – step: compute $p_{Z_t|\boldsymbol{X}_t,Y_t}(.|\boldsymbol{x}_t, y_t; \hat{\boldsymbol{\theta}}^{(n)}(\boldsymbol{x}_1^N, y_1^N))$, for $t = 1, \ldots, N$ and use it to compute the conditional expectation (with respect to the pdf $p_{Z_1^N|\boldsymbol{X}_1^N,\, Y_1^N}(.|\boldsymbol{x}_1^N, y_1^N; \hat{\boldsymbol{\theta}}^{(n)}(\boldsymbol{x}_1^N, y_1^N))$
$$Q^{(n)}(\boldsymbol{\theta}) = \mathcal{E}\{\log p_{\boldsymbol{X}_1^N,\, Z_1^N|Y_1^N}(\boldsymbol{x}_1^N, .|y_1^N; \boldsymbol{\theta})\}$$
2) M – step: compute
$$\hat{\boldsymbol{\theta}}^{(n+1)}(\boldsymbol{x}_1^N) = \arg\max_{\boldsymbol{\theta}}\{Q^{(n)}(\boldsymbol{\theta})\},$$

.

The E-step, consists of first computing the current conditional distribution of the hidden component-indicator categorical variable $p_{Z_t|\boldsymbol{X}_t,Y_t}(.\,|\boldsymbol{x}_t, y_t; \hat{\boldsymbol{\theta}}^{(n)}(\boldsymbol{x}_1^N, y_1^N))$, for $t = 1, \ldots, N$, given the current estimate of the parameter vector $\hat{\boldsymbol{\theta}}^{(n)}$. Subsequently the E-step uses this conditional probability distribution to compute the conditional expectation $Q^{(n)}(\theta)$. From eq. (6) we already have the answer to the first step

$$\begin{aligned} h_k^{(n)}(\boldsymbol{x}_t, y_t) &= p_{Z_t|\boldsymbol{X}_t,Y_t}(k|\boldsymbol{x}_t, y_t; \hat{\boldsymbol{\theta}}^{(n)}) \\ &= p_{Z_t|\boldsymbol{X}_t,Y_t}(\boldsymbol{z}_{t,\,k} = 1|\boldsymbol{x}_t, y_t; \hat{\boldsymbol{\theta}}^{(n)}) = \mathcal{E}\left\{\boldsymbol{z}_{t,\,k}|\boldsymbol{x}_t, y_t, \hat{\boldsymbol{\theta}}^{(n)}\right\} \\ &= \frac{\hat{\alpha}_{y_t,k}^{(n)}\, G\left(\boldsymbol{x}_t\,;\hat{\mu}_k^{(n)},\, \hat{\boldsymbol{\Sigma}}_k^{(n)}\right)}{\sum_{k=1}^{K}\hat{\alpha}_{y_t,k}^{(n)}\, G\left(\boldsymbol{x}_t\,;\, \hat{\boldsymbol{\mu}}_k^{(n)},\, \hat{\boldsymbol{\Sigma}}_k^{(n)}\right)}, \quad k = 1, 2, ..., K. \end{aligned} \tag{10}$$

The second computation of the E-step results in

$$\begin{aligned} &Q^{(n)}(\boldsymbol{\theta}) \\ &= -\frac{1}{2}ND\log(2\pi) + \frac{1}{2}\sum_{t=1}^{N}\sum_{k=1}^{K}\left\{h_k^{(n)}(\boldsymbol{x}_t, y_t)\log|\boldsymbol{\Sigma}_k^{-1}|\right\} \\ &\quad -\frac{1}{2}\sum_{t=1}^{N}\sum_{k=1}^{K}\left\{h_k^{(n)}(\boldsymbol{x}_t, y_t)(\boldsymbol{x}_t - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_t - \boldsymbol{\mu}_k)\right\} \\ &\quad +\sum_{t=1}^{N}\sum_{k=1}^{K}\left\{h_k^{(n)}(\boldsymbol{x}_t, y_t)\log\alpha_{y_t,\,k}\right\} \end{aligned} \tag{11}$$

The M – step consists of maximizing $Q^{(n)}(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$, i.e. w.r.t. $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, $k = 1, \ldots, K$ and $\alpha_{c,\,k}$, $c = 1, 2, \ldots, C$, $k = 1, 2, \ldots, K$. We rewrite the part of $Q^{(n)}(\boldsymbol{\theta})$ that involves components of the model parameters, namely $\boldsymbol{\mu}_k$ and $\boldsymbol{\Lambda}_k$ (where $\boldsymbol{\Lambda}_k = \boldsymbol{\Sigma}_k^{-1} \geq 0$ and symmetric) as (we left out the 1/2 factor)

$$\begin{aligned} \mathcal{F}(\boldsymbol{\mu}, \boldsymbol{\Lambda}) = \sum_{k=1}^{K} Tr\Big[&\rho_k\log\boldsymbol{\Lambda}_k - \boldsymbol{\Gamma}_k\boldsymbol{\Lambda}_k + \frac{1}{\rho_k}\boldsymbol{\xi}_k\boldsymbol{\xi}_k^T\boldsymbol{\Lambda}_k \\ &- \rho_k\left(\boldsymbol{\mu}_k - \frac{1}{\rho_k}\boldsymbol{\xi}_k\right)\left(\boldsymbol{\mu}_k - \frac{1}{\rho_k}\boldsymbol{\xi}_k\right)^T\boldsymbol{\Lambda}_k\Big]. \end{aligned} \tag{12}$$

In eq. (12) we have introduced scalars (ρ_k), D-vectors $\boldsymbol{\xi}_k$ and matrices ($D \times D$ nonnegative and symmetric) $\boldsymbol{\Gamma}_k$):

$$\begin{aligned} \rho_k &= \sum_{t=1}^{N} h_k^{(n)}(\boldsymbol{x}_t, y_t),\ k = 1, 2, \ldots, K, \quad \sum_{k=1}^{K}\rho_k = N, \\ \boldsymbol{\Gamma}_k &= \sum_{t=1}^{N} h_k^{(n)}(\boldsymbol{x}_t, y_t)\boldsymbol{x}_t\boldsymbol{x}_t^T, \quad k = 1, 2, \ldots, K, \\ \boldsymbol{\xi}_k &= \sum_{t=1}^{N} h_k^{(n)}(\boldsymbol{x}_t, y_t)\boldsymbol{x}_t, \quad k = 1, 2, \ldots, K. \end{aligned}$$

It is well known [8], [10], that $\log Det[.]$ is concave over the set of symmetric positive definite matrices. It then follows, since $Tr[\mathbf{B}\boldsymbol{\Lambda}]$ is linear in $\boldsymbol{\Lambda}$ for any matrix $\mathbf{B}$ and the negative of a positive semidefinite quadratic form is concave, that $\mathcal{F}(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ is concave jointly in all variables. Since it is also continuous in all variables it has a unique maximum, achieved at the point $(\boldsymbol{\mu}^*, \boldsymbol{\Lambda}^*)$.

Let

$$\begin{aligned} \mathcal{G}(\boldsymbol{\Lambda}) &= \textstyle\sum_{k=1}^{K} Tr\left[\rho_k\log\boldsymbol{\Lambda}_k - \boldsymbol{\Gamma}_k\boldsymbol{\Lambda}_k + \frac{1}{\rho_k}\xi_k\xi_k^T\boldsymbol{\Lambda}_k\right] \\ &= \textstyle\sum_{k=1}^{K}\rho_k\log Det\,\boldsymbol{\Lambda}_k - Tr\left[\left(\boldsymbol{\Gamma}_k - \frac{1}{\rho_k}\xi_k\xi_k^T\right)\boldsymbol{\Lambda}_k\right]. \end{aligned}$$

Since $\mathcal{G}(\boldsymbol{\Lambda})$is also concave and continuous jointly in all variables, it also has a unique maximum. Note that the matrix inside the trace, pre-multiplying $\boldsymbol{\Lambda}_k$ is symmetric and positive semidefinite. Now considering the function $\mathcal{H}(\boldsymbol{\Lambda}) = \alpha\log Det\,\boldsymbol{\Lambda} - Tr[\boldsymbol{B}\boldsymbol{\Lambda}]$, where α is a positive scalar and $\boldsymbol{B}$ is a positive definite symmetric matrix, it can be shown [6] that the unique maximum of $\mathcal{H}(\boldsymbol{\Lambda})$ is at $\boldsymbol{\Lambda}^* = \alpha\boldsymbol{B}^{-1}$. Then it can be shown [6] that

$$\begin{aligned} \mathcal{F}(\boldsymbol{\mu}, \boldsymbol{\Lambda}) &\leq \mathcal{F}(\hat{\boldsymbol{\mu}}^{(n+1)}, \boldsymbol{\Lambda}) = \mathcal{G}(\boldsymbol{\Lambda}) \leq \mathcal{G}(\boldsymbol{\Lambda}^*) \\ &= \mathcal{F}(\hat{\boldsymbol{\mu}}^{(n+1)}, (\hat{\boldsymbol{\Sigma}}^{(n+1)})^{-1})\ \text{, for all } \boldsymbol{\mu}, \boldsymbol{\Sigma}. \end{aligned} \tag{13}$$

Therefore the maximization of $Q^{(n)}(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ is achieved at the $\hat{\boldsymbol{\mu}}_k^{(n+1)}$ and $\hat{\boldsymbol{\Sigma}}_k^{(n+1)}$ provided by the iterations of eq. (17) below.

Third, with respect to the mixture coefficients $\alpha_{c,\,k}$, $c = 1, 2, \ldots, C$, $k = 1, 2, \ldots, K$, or equivalently the class-component mixture coefficient $C \times K$ matrix $\boldsymbol{\mathcal{A}}$, the part of $Q^{(n)}(\boldsymbol{\theta})$ that depends on these variables is

$$\begin{aligned}\mathcal{I}(\boldsymbol{\mathcal{A}}) &= \sum_{t=1}^{N}\sum_{k=1}^{K}\left\{h_k^{(n)}(\boldsymbol{x}_t, y_t)\log\alpha_{y_t,k}\right\}\\ &= \sum_{t=1}^{N}\sum_{c=1}^{C}\sum_{k=1}^{K} h_k^{(n)}(\boldsymbol{x}_t, y_t)\delta(c, y_t)\log\alpha_{c,\,k}\ \ , \qquad (14)\end{aligned}$$

where $\delta(c, y_t)$ is the Kronecker delta.

For each $c = 1, \dots, C$, let $\mathcal{N}_c = \{t \in \{1, 2, \dots, N\} |\ y_t = c\}$, and $N_c = |\mathcal{N}_c|$ be its cardinality. The function $\mathcal{I}(\boldsymbol{\mathcal{A}})$ is a convex combination of concave functions of each of its coordinates, and thus has clearly a unique maximum over the multi-simplex S^C in $\mathcal{R}^{KC}$, where S is the set of all vectors $\boldsymbol{\gamma}$ in $\mathcal{R}^K$ with $0 \leq \gamma_k \leq 1,\ k = 1, \dots, K$ and $\sum\limits_{k=1}^{K}\gamma_k = 1$. By re-writing eq. (14) as

$$\mathcal{I}(A) = \sum_{c=1}^{C}\sum_{k=1}^{K}\beta_{c,\,k}\log\alpha_{c,\,k}, \qquad (15)$$

where

$$\beta_{c,\,k} = \sum_{t=1}^{N} h_k^{(n)}(\boldsymbol{x}_t, y_t)\delta(c, y_t) = \sum_{t\in\mathcal{N}_c} h_k^{(n)}(\boldsymbol{x}_t, c).$$

and using the log-sum inequality [10] we can show that for each c the maximum is achieved for (see [6] for details)

$$\begin{aligned}&\frac{\beta_{c,\,k}}{\hat{\alpha}_{c,\,k}^{(n+1)}} = \gamma_c = const., \quad k = 1, 2, \dots, K\\ &\sum_{k=1}^{K}\sum_{t\in\mathcal{N}_c} h_k^{(n)}(\boldsymbol{x}_t, c) = \gamma_c\sum_{k=1}^{K}\hat{\alpha}_k^{(n+1)}, \quad \text{or} \quad N_c = \gamma_c,\end{aligned} \qquad (16)$$

from which the last iteration, eq. (17), below results.

We have therefore established the following

Theorem 3.1: The EM algorithm for the hierarchical model starts from an initial value of the estimate $\hat{\boldsymbol{\theta}}^{(0)}$ and computes iterative estimates $\hat{\boldsymbol{\theta}}^{(n)}$ as follows

$$\begin{aligned}h_k^{(n)}(\boldsymbol{x}_t, y_t) &= \frac{\hat{\alpha}_{y_t,k}^{(n)} G(\boldsymbol{x}_t; \hat{\boldsymbol{\mu}}_k^{(n)}, \hat{\boldsymbol{\Sigma}}_k^{(n)})}{\sum_{k=1}^{K}\hat{\alpha}_{y_t,k}^{(n)} G(\boldsymbol{x}_t; \hat{\boldsymbol{\mu}}_k^{(n)}, \hat{\boldsymbol{\Sigma}}_k^{(n)})},\\ k &= 1, 2, \dots, K\\ \hat{\boldsymbol{\mu}}_k^{(n+1)} &= \frac{\sum_{t=1}^{N} h_k^{(n)}(\boldsymbol{x}_t, y_t)\boldsymbol{x}_t}{\sum_{t=1}^{N} h_k^{(n)}(\boldsymbol{x}_t, y_t)},\\ k &= 1, 2, \dots, K\\ \hat{\boldsymbol{\Sigma}}_k^{(n+1)} &= \frac{\sum_{t=1}^{N} h_k^{(n)}(\boldsymbol{x}_t, y_t)[\boldsymbol{x}_t - \hat{\boldsymbol{\mu}}_k^{(n+1)}][\boldsymbol{x}_t - \hat{\boldsymbol{\mu}}_k^{(n+1)}]^T}{\sum_{t=1}^{N} h_k^{(n)}(\boldsymbol{x}_t, y_t)}\\ k &= 1, 2, \dots, K\\ \hat{\alpha}_{c,k}^{(n+1)} &= \frac{\sum_{t\in\mathcal{N}_c} h_k^{(n)}(\boldsymbol{x}_t, y_t)}{N_c}, \qquad (17)\\ c &= 1, \dots, C,\ k = 1, \dots, K\end{aligned}$$

When this algorithm converges, we have, through the parameters, estimates of the class probability distributions, for each class c, which was learned from the training data. These class models can then be used to compute likelihood ratios for new data and assign the data to the class with the highest value of the likelihood. First, compute for the new sample $\boldsymbol{x}_s$ the posterior probability using the "learned" model

$$\Pr\{Y = c|\boldsymbol{X} = \boldsymbol{x}_s; \hat{\boldsymbol{\theta}}\} = \frac{\pi_c\sum_{k=1}^{K}\hat{\alpha}_{c,k} G(\boldsymbol{x}_s; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)}{\sum_{c=1}^{C}\sum_{k=1}^{K}\pi_c\hat{\alpha}_{c,k} G(\boldsymbol{x}_s; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)}$$

for $c = 1, \dots, C$. Then compute the classification decision

$$\hat{d}(\boldsymbol{x}_s) = \arg\max_c \pi_c\sum_{k=1}^{K}\hat{\alpha}_{c,k} G(\boldsymbol{x}_s; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k). \qquad (18)$$

IV. Solution of the Joint Optimization Problem

We want to find a decision rule for assigning a class to each data vector (i.e. a classification strategy) that is optimal in some sense. That is we want to design a function, usually called the *classifier*,

$$d : \mathcal{R}^D \rightarrow \{1, 2, \dots, C\},$$

where $d(\boldsymbol{x})$ is the class estimate for the data vector $\boldsymbol{x}$.

A. Costs and Approximate Costs

We follow a Bayesian framework [22]. Let $B_{cc'}$ denote the penalty (cost) for deciding that the class of a data vector $\boldsymbol{x}$ is c' ($d(\boldsymbol{x}) = c'$) when the true (but unknown) class is c . We consider the simpler cost [22]

$$B_{cc'} = \begin{cases} 0, & \text{if } c = c' = d(\boldsymbol{x})\\ 1, & \text{if } c \neq c' = d(\boldsymbol{x})\end{cases} = 1 - \delta(c, c') = 1 - \mathcal{I}_{d(\boldsymbol{x})=c}$$

where $\delta(c, c')$ denotes the Kronecker delta, as it captures all the essential ingredients of the problem and of our approach. The expression $1 - \mathcal{I}_{\{d(\boldsymbol{x})=c\}}$, with c being the true class of the data $\boldsymbol{x}$, and $\mathcal{I}_H$ the characteristic function of the set H, is commonly known as the *classification error* associated with the classifier $d(.)$. The *probability of correct classification* for the classifier $d(.)$ is $Pr\{d(\boldsymbol{x}) = c\}$.

Following the Bayesian framework we want to find a classifier $d^*(.)$ to minimize the expected cost for a classifier $d(.)$ over the joint distribution of data vectors and their classes, which in this case becomes the *expected classification error* for classifier $d(.)$. As is well known the optimal classifier for this formulation is the *Bayes decision rule*

$$d^*(\boldsymbol{x}) = \arg\max_{c'}\{p_{Y|\boldsymbol{X}}(c'|\boldsymbol{x})\}, \qquad (19)$$

that is assign to the data vector $\boldsymbol{x}$ the class $\hat{c}$ for which $p_{Y|\boldsymbol{X}}(\hat{c}|\boldsymbol{x}) \geq p_{Y|\boldsymbol{X}}(c'|\boldsymbol{x})$, for all $c' \neq \hat{c}$; i.e. *the class with the maximum posterior probability given* $\boldsymbol{x}$.

’The corresponding minimum classification error for a data vector $\boldsymbol{x}$ when the Bayes rule d^* is used, the *Bayes error*, is

$$\begin{aligned}J(d^*; \boldsymbol{x}) &= \mathcal{E}_{Y|\boldsymbol{X}}\{1 - \delta(y, d^*(\boldsymbol{x}))\} = 1 - p_{Y|\boldsymbol{X}}(d^*(\boldsymbol{x})|\boldsymbol{x})\\ &= \sum_{c=1}^{C} p_{Y|\boldsymbol{X}}(c|\boldsymbol{x})[1 - \delta(c, d^*(\boldsymbol{x}))] \qquad (20)\end{aligned}$$

while the *average Bayes error*, is

$$J(d^*) = \mathcal{E}_{\boldsymbol{X}}\{J(d^*;\boldsymbol{x})\} = \min_{d} J(d) \tag{21}$$

Proceeding in exactly the same steps we obtain that for a parametric model of all relevant distributions, like the finite mixture models introduced earlier, i.e. $p_{\boldsymbol{X},Y}(\boldsymbol{x},y;\boldsymbol{\phi})$ and all marginals and conditional distributions obtained from this model, the optimal classifier is the *parametric Bayes rule*

$$\tilde{d}(\boldsymbol{x};\boldsymbol{\phi}) = \arg\max_{c'}\{p_{Y|\boldsymbol{X}}(c'|\boldsymbol{x};\boldsymbol{\phi}\} \tag{22}$$

The corresponding *minimal parametric (or model-based) classification error* for a data vector $\boldsymbol{x}$ is then

$$J(\tilde{d};\boldsymbol{x},\boldsymbol{\theta}) = \mathcal{E}_{Y|\boldsymbol{X}}\{1-\delta(y,\tilde{d}(\boldsymbol{x};\boldsymbol{\theta}))\} = 1 - p_{Y|\boldsymbol{X}}(\tilde{d}(\boldsymbol{x};\boldsymbol{\theta}|\boldsymbol{x})$$
$$= \sum_{c=1}^{C} p_{Y|\boldsymbol{X}}(c|\boldsymbol{x})[1-\delta(c,\tilde{d}(\boldsymbol{x};\boldsymbol{\theta}))] \tag{23}$$

where we have used the true posterior distribution of the classes given the data vectors $p_{Y|\boldsymbol{X}}$. Proceeding, we compute the optimal expected parametric (model-based) classification error over the true distribution of the data vectors $p_{\boldsymbol{X}}$, for this optimal parametric Bayes rule $\tilde{d}(.;\boldsymbol{\theta})$

$$J(\tilde{d};\boldsymbol{\theta}) = \mathcal{E}_{\boldsymbol{X}}\{J(\tilde{d};\boldsymbol{x},\boldsymbol{\theta})\} = \min_{d} J(d;\boldsymbol{\theta}). \tag{24}$$

This quantity is also known as the *optimal model-based expected classification error* and as the *model-based (or parametric) Bayes error* [22]. We are interested to select the parametric models so as to minimize the expected parametric classification error. This is equivalent to minimizing $J(\tilde{d};\boldsymbol{\theta})$ with respect to the vector of the model parameters $\boldsymbol{\theta}$. This minimization needs to be done based on what we have available - the training data set $\mathcal{S} = \{(\boldsymbol{x}_t, y_t);\ t = 1,2,...,N\}$. This direct approach is infeasible because of the nonlinearities involved in the functional $J(\tilde{d};\boldsymbol{\theta})$ and most importantly because the true distribution $p_{\boldsymbol{X}}$ is *unknown*.

Instead, in the next subsection, we establish an *upper bound* for the optimal model-based (parametric) expected classification error (24), with the idea being to obtain the 'best' model parameters by minimizing this upper bound.

B. A Tight Upper Bound for the Model-Based Bayes Error

We derive bounds between the (true) Bayes error and the minimum model-based classification error for a parametrized class of models. From (19) and (22) we have the inequality

$$\begin{aligned} J(\tilde{d};\boldsymbol{x},\boldsymbol{\theta}) - J(d^*;\boldsymbol{x}) &= \\ &= \Big(1 - p_{Y|\boldsymbol{X}}(\tilde{d}(\boldsymbol{x};\boldsymbol{\theta})|\boldsymbol{x})\Big) - \big(1 - p_{Y|\boldsymbol{X}}(d^*(\boldsymbol{x})|\boldsymbol{x})\big) \\ &\le |p_{Y|\boldsymbol{X}}(d^*(\boldsymbol{x})|\boldsymbol{x}) - p_{Y|\boldsymbol{X}}(d^*(\boldsymbol{x})|\boldsymbol{x};\boldsymbol{\theta})| \\ &\quad + |p_{Y|\boldsymbol{X}}(\tilde{d}(\boldsymbol{x};\boldsymbol{\theta})|\boldsymbol{x}) - p_{Y|\boldsymbol{X}}(\tilde{d}(\boldsymbol{x};\boldsymbol{\theta})|\boldsymbol{x};\boldsymbol{\theta})|. \end{aligned} \tag{25}$$

We can now proceed to obtain, as a direct consequence of the inequality (25), the following useful bound

$$J((\tilde{d};\boldsymbol{x},\boldsymbol{\theta}) - J(d^*;\boldsymbol{x}) \le \sum_{c=1}^{C} |p_{Y|\boldsymbol{X}}(c|\boldsymbol{x}) - p_{Y|\boldsymbol{X}}(c|\boldsymbol{x};\boldsymbol{\theta})|. \tag{26}$$

The usefulness of this bound comes from the fact that as the approximate model class posterior probability distribution $p_{Y|\boldsymbol{X}}(.|\boldsymbol{x};\boldsymbol{\theta})$ becomes a better approximation to the true class posterior probability distribution $p_{Y|\boldsymbol{X}}(.|\boldsymbol{x})$, the optimal model-based classification error $J(\hat{d};\boldsymbol{x},\boldsymbol{\theta})$ comes closer to the Bayes error $J(d^*;\boldsymbol{x})$, for any observed data vector $\boldsymbol{x} \in \mathcal{R}^D$. Thus the bound can be considered *tight* from this perspective. We next obtain a bound between the expected Bayes error $J(d^*)$ and the optimal model-based expected classification error $J(\tilde{d};\boldsymbol{\theta})$, by taking the expectation, with respect to the true (but unknown) probability distribution of the data $p_{\boldsymbol{X}}(.)$, of both sides of (26).

For two probability distributions $p_1(.)$, $p_2(.)$ on the finite set $\mathcal{C} = \{1,2,...,C\}$, a useful and well known measure of *dissimilarity* of the two distributions is the *Kullback Leibler 'distance'* , *relative entropy*, or *information divergence* between p_1 and p_2 [10]: $D(p_1||p_2) = \sum_{c=1}^{C} p_1(c)\log\frac{p_1(c)}{p_2(c)} = \mathcal{E}_{p_1}\left\{\log\frac{p_1(Y)}{p_2(Y)}\right\}$; where $\log$ denotes natural logarithms. Another well known distance between two probability distributions is the l_1 *distance* or *variational distance* [10]: $V(p_1,p_2) = ||p_1 - p_2||_1 = \sum_{c=1}^{C}|p_1(c) - p_2(c)|$. The Pinsker inequality [10] establishes a useful relationship between these two important measures

$$D(p_1 \parallel p_2) \ge \frac{1}{2}V(p_1,p_2)^2.$$

Using the Pinsker inequality we obtain the following bound

Theorem 4.1:

$$\begin{aligned} &\Big(J(\tilde{d};\boldsymbol{\theta}) - J(d^*)\Big)^2 \\ &= \left(\int V(p_{Y|\boldsymbol{X}}(.|\boldsymbol{x}), p_{Y|\boldsymbol{X}}(.|\boldsymbol{x};\boldsymbol{\theta}))p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x}\right)^2 \\ &\le 2\int D(p_{Y|\boldsymbol{X}}(.|\boldsymbol{x}) \parallel p_{Y|\boldsymbol{X}}(.|\boldsymbol{x};\boldsymbol{\theta}))p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x} \\ &= 2\int\left(\sum_{c=1}^{C} p_{Y|\boldsymbol{X}}(c|\boldsymbol{x})\log\frac{p_{Y|\boldsymbol{X}}(c|\boldsymbol{x})}{p_{Y|\boldsymbol{X}}(c|\boldsymbol{x};\boldsymbol{\theta})}\right)p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x} \end{aligned} \tag{27}$$

This bound between the average Bayes error (the true minimal classification error) and the average model-based classification error, is given in terms of the average (with respect to observations, or data) of the Kullback-Leibler divergence (or relative entropy) of the true (but unknown) class a posteriori distribution given the measurements and the approximate (model-based) class a posteriori distribution given the measurements.

C. Optimal Discriminating Model Parameter Estimate

In this section we use the bound (27) to obtain rigorously the "optimal discriminating" model parameter estimate. Thus the first use of the bound is to determine the optimal parametric model; i.e. the parametric model for a given class that obtains the best approximation to the average Bayes error. For the finite normal mixtures used in our framework, this amounts to finding the estimate $\check{\boldsymbol{\theta}}$ that minimizes the bound (27) [6]. However, the true probability distribution of the data $p_{\boldsymbol{X}}(.)$ is unknown, and what we have to work with is the set of training data $\mathcal{S}$. Using these labelled training data, we approximate, as usual, the unknown probability distribution $p_{\boldsymbol{X},Y}(\boldsymbol{x},y)$, by the empirical distribution

$$\hat{p}_{\boldsymbol{X},Y}(\boldsymbol{x},y)=\frac{1}{N}\sum_{t=1}^{N}\delta(\boldsymbol{x}-\boldsymbol{x}_t)\delta(y,y_t). \tag{28}$$

Thus letting

$$\boldsymbol{\mathcal{B}}(\boldsymbol{\theta})=\int\left(\sum_{c=1}^{C}p_{Y|\boldsymbol{X}}(c|\boldsymbol{x})\log\frac{p_{Y|\boldsymbol{X}}(c|\boldsymbol{x})}{p_{Y|\boldsymbol{X}}(c|\boldsymbol{x};\boldsymbol{\theta})}\right)p_{\boldsymbol{X}}d\boldsymbol{x},$$

we need to compute

$$\check{\boldsymbol{\theta}}=\arg\min_{\boldsymbol{\theta}}\left\{\boldsymbol{\mathcal{B}}(\boldsymbol{\theta})\right\}. \tag{29}$$

First, we expand $\boldsymbol{\mathcal{B}}(\boldsymbol{\theta})$ as follows

$$\begin{aligned}\boldsymbol{\mathcal{B}}(\boldsymbol{\theta})=&\int\sum_{c=1}^{C}\log p_{Y|\boldsymbol{X}}(c|\boldsymbol{x})p_{\boldsymbol{X},Y}(\boldsymbol{x},c)d\boldsymbol{x}\\&-\int\sum_{c=1}^{C}\log p_{Y|\boldsymbol{X}}(c|\boldsymbol{x};\boldsymbol{\theta})p_{\boldsymbol{X},Y}(\boldsymbol{x},c)d\boldsymbol{x}\end{aligned} \tag{30}$$

Let

$$\boldsymbol{\mathcal{M}}(\boldsymbol{\theta})=\int\sum_{c=1}^{C}\log p_{Y|\boldsymbol{X}}(c|\boldsymbol{x};\boldsymbol{\theta})p_{\boldsymbol{X},Y}(\boldsymbol{x},c)d\boldsymbol{x}. \tag{31}$$

Since the first component in the expansion (30) of $\boldsymbol{\mathcal{B}}(\boldsymbol{\theta})$ does not depend on $\boldsymbol{\theta}$, we have

$$\check{\boldsymbol{\theta}}=\arg\min_{\boldsymbol{\theta}}\left\{\boldsymbol{\mathcal{B}}(\boldsymbol{\theta})\right\}=\arg\max_{\boldsymbol{\theta}}\left\{\boldsymbol{\mathcal{M}}(\boldsymbol{\theta})\right\}.$$

Using the empirical distribution approximation (28) we obtain an empirical approximation to $\boldsymbol{\mathcal{M}}(\boldsymbol{\theta})$

$$\hat{\boldsymbol{\mathcal{M}}}(\boldsymbol{\theta})=\frac{1}{N}\sum_{t=1}^{N}\log p_{Y|\boldsymbol{X}}(y_t|\boldsymbol{x}_t;\boldsymbol{\theta}). \tag{32}$$

Thus we have established the following

Theorem 4.2: To get the best approximation to the Bayes error, by learning from the labelled training data we compute the estimate of the model parameters $\boldsymbol{\theta}$ as follows

$$\check{\boldsymbol{\theta}}=\arg\max_{\boldsymbol{\theta}}\left\{\sum_{t=1}^{N}\log p_{Y|\boldsymbol{X}}(y_t|\boldsymbol{x}_t;\boldsymbol{\theta})\right\}. \tag{33}$$

This model parameter estimate *maximizes the likelihood of the model-based class* ***posterior*** *probability distribution* $p_{Y|\boldsymbol{X}}(y|\boldsymbol{x};\boldsymbol{\theta})$. This is different from the common approach where first the model parameters are estimated by maximizing the likelihood of the class conditional probability distribution $p_{\boldsymbol{X}|Y}(\boldsymbol{x}|y;\boldsymbol{\theta})$. The criterion and maximization we developed here are similar to *conditional maximum likelihood* [33], or *maximum mutual information* [4], [7]. Several recent papers on classification comparisons have shown that these methods provide better classification performance.

The second important use of the bound (27) that we have developed (see [6]) is to derive an elegant and efficient recursive (in terms of observed data sequences of increasing length) algorithm, which is inspired by the EM algorithm for maximum likelihood estimation of model parameters (see section III), for the computation of the optimal discriminating model parameter estimate $\check{\boldsymbol{\theta}}$.

V. Context Dependence in Bioinformatics and ACON

Let us first clarify, that when discussing bioinformatics we are primarily referring to technologies that assess the level of expression for many genes (~20,000) in cells from a given sample. Context dependent learning in classification problems implies that without knowledge of all classes possible, one cannot construct accurate decision boundaries. In other words, we need the full context of the classification question in order to properly characterize it.

One possible reason for this is the notion of heterogeneous classes. This concept has received much attention recently in cancer biology. The advent of new high-throughput technologies has allowed researchers and clinicians to dramatically increase the number of features they can describe tumors samples by. Rather straightforward unsupervised clustering of tumors previously considered rather homogenous, e.g. breast cancer specimens, has shown that a significant level of unappreciated heterogeneity exists [17], [30], [34], [35]. The relevance of this heterogeneity to clinical parameters is currently an active area of research. Particularly with regards to examining if this information can help to better predict if and with what type of agent a therapeutic response could be achieved [3], [16], [24], [31], [36].

We argue that an ACON learning approach should fair better than an OCON in this setting. This intuition basically stems from the high possibility of overlapping groups given the scenario described above. To illustrate this notion we present a hypothetical example. Imagine that our task is to classify instruments of an orchestra composed only of flutes, saxophones, clarinets, and violins into two possibilities: 1) made of wood 2) made of metal. This classification may seem odd given our complete understanding of how these instruments generate their sounds. However, the analogous knowledge in characterizing disease states in biology is not fully characterized. Rather we initially only had a crude ability to observe the “instruments” and have noted that some are made of wood and some are made of metal. Additionally, we do not have the luxury of being able to isolate each specific instrument. We can only isolate the groups of instruments, wood versus metal. A new recording technology now available, allows us to record many more features from a given sample of instruments. From this data we now observe heterogeneity is the wood instruments and metal instrument with some overlap though.

Conceptually, we can consider the cell to be a black box system with inputs and outputs. This is not far from reality; we currently only have a marginal understanding of all the circuits within the cell. The inputs to this system could represent various biological states or phenomena and the outputs represent the expression levels observed for the ~20,000 genes queried. In the most straightforward scenario, a classification task would involve learning the biological phenomenon of the cell from the expression data. Context dependence in these types of scenarios arises primarily because of the large degree of indirectness between the phenomena examined and the data used to characterize them. A large amount of processing happens in the cellular black box that then may eventually result in changes of expression patterns of the genes queried. The various different tissue types (i.e. contexts with the body) may initiate different expression programs in response to the same stimuli or under the same biological state; however, some overlap may be expected. A straightforward example of this concept comes from cancer biology. It is now well established that given samples of a specific tumor type, lung cancer, and its benign counterpart, normal lung tissue, one will find many gene expression alterations that can differentiate the two.

In our current research we are applying the methods and algorithms described here in various lymphomas and lung cancers. After training our ACON Gaussian Mixture Model (GMM) in this feature space, independent test samples are well characterized. In practice, the feature space, where our ACON GMM will form decision boundaries, are usually of 5-10 dimensions [6].

REFERENCES

[1] A. Alizadeh and et al, "Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling," *Nature 2000*, vol. 403, pp. 503–511, 2000.

[2] S. Axelrod, V. Goel, R. Gopinath, P. Olsen, and K. Visweswariah, "Discriminative estimation of subspace constrained gaussian mixture models for speech recognition," *IEEE Trans. On Audio, Speech and language Processing*, vol. 15, no. 1, pp. 172–189, Jan. 2007.

[3] M. Ayers *et al.*, "Gene expression profiles predict complete pathologic response to neoadjuvant paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide chemotherapy in breast cancer," *J Clin Oncol*, vol. 22, no. 12, pp. 2284–93, 2004.

[4] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden markov model parameters for speech recognition," in *Proc. ICASSP*, 1986, p. 4952.

[5] A. Baras and J. Baras, "Joint parameter estimation and feature selection for bioinformatics classification," *submitted for publication*, 2009.

[6] ——, "Joint parameter estimation and misclassification error minimization and applications," *submitted for publication*, 2009.

[7] D. Barber and F. Agakov, "The IM algorithm: A variational approach to information maximization," *Advances in Neural Information Processing Systems*, vol. 16, pp. 201–208, 2004.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[9] W. Buntine, "Variational extensions to EM and multinomial PCA," in *Proceedings of the 13th European Conference on Machine Learning*, 2002, pp. 23–34.

[10] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley, 1991.

[11] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. of Royal Statistical Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[12] F. Fleuret, "Fast binary feature selection with conditional mutual information," *J. of Machine Learning Research*, vol. 5, pp. 1531–1555, 2004.

[13] T. Golub and et al, "Molecular classification of cancer class discovery and class prediction by gene expression monitoring," *Science 1999*, vol. 286, pp. 531–537, 1999.

[14] A. Gunawardana and W. Byrne, "Discriminative speaker adaptation with conditional maximum likelihood linear regression," in *Eur. Conf. Speech Commun. Technol.*, 2001, pp. 1203–1206.

[15] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Verlag, 2001.

[16] K. R. Hess *et al.*, "Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer," *J Clin Oncol*, vol. 24, no. 26, pp. 4236–44, 2006.

[17] Z. Hu *et al.*, "The molecular portraits of breast tumors are conserved across microarray platforms," *BMC Genomics*, no. 7, p. 96, 2006.

[18] T. Jebara and A. Pentland, "Maximum conditional likelihood via bound maximization and the CEM algorithm," *Advances in Neural Information Processing Systems*, vol. 11, pp. 494–500, 1999.

[19] B. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. On Signal Processing*, vol. 40, no. 12, pp. 3043–3054, 1992.

[20] S. Katagiri, C. Lee, and B. Juang, "Discriminative multilayer feedforward networks," in *Proc. IEEE Workshop Neural Networks for Signal Processing*, Princeton, NJ, 1991, pp. 11–20.

[21] A. Klautau, N. Jevtic, and A. Orlitsky, "Discriminative gaussian mixture models: A comparison with kernel classifiers," in *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, 2003, pp. 353–360.

[22] S. Kung, M. Mak, and S. Lin, *Biometric Authentication, A Machine Learning Approach*. Prentice Hall, 2005.

[23] S. Kung and J. Taur, "Decision-based neural networks with signal/image classification applications," *IEEE Trans. On Neural Networks*, vol. 6, no. 1, pp. 170–181, 1995.

[24] J. Lee *et al.*, "A strategy for predicting the chemosensitivity of human cancers and its application to drug discovery," *Proc Natl Acad Sci U S A*, vol. 104, no. 32, pp. 13 086–91, 2007.

[25] S. Lin and S. Kung, "Probabilistic DBNN via expectation-maximization with multi-sensor classification applications," in *Proc. 1995 IEEE International Conference on Image Processing*, vol. III, Washington, DC, Oct. 1995, pp. 236–239.

[26] C. Liu, C. Lee, B. Juang, and A. Rosenberg, "Speaker recognition based on minimum error discriminative training," in *Proc. ICASSP94*, vol. 1, 1994, pp. 325–328.

[27] C. Liu, H. Wang, and C. Lee, "Speaker verification using normalized log-likelihood score," *IEEE Trans. On Speech and Audio Processing*, vol. 49, no. 1, pp. 56–60, 1996.

[28] H. C. M. Soukup and J. Lee, "Robust classification modeling on microarray data using misclassification penalized posterior," *Bioinformatics 2005*, vol. 21, pp. 423–430, 2005.

[29] Y. Normandin and S. D. Morgera, "An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition," in *Proceedings 1991 International Conference on Acoustics, Speech and Signal Processing (ICASSP 91)*, 1991, pp. 537–540.

[30] C. Perou *et al.*, "Molecular portraits of human breast tumours," *Nature*, vol. 406, no. 6797, pp. 747–52, 2000.

[31] A. Potti *et al.*, "Genomic signatures to guide the use of chemotherapeutics," *Nat Med*, vol. 12, no. 11, pp. 1294–300, 2006.

[32] R. Redner and H. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, 1984.

[33] J. Salojärvi and K. Puolamäki, "Expectation maximization algorithms for conditional likelihoods," in *Proceedings of the 22nd International Conference on Machine Learning (ICML-2005)*. ACM Press, 2005, pp. 753–760.

[34] T. Sorlie *et al.*, "Repeated observation of breast tumor subtypes in independent gene expression data sets," *Proc Natl Acad Sci U S A*, vol. 100, no. 14, pp. 8418–23, 2003.

[35] C. Sotiriou and L. Pusztai, "Gene-expression signatures in breast cancer," *N Engl J Med*, vol. 360, no. 8, pp. 790–800, 2009.

[36] J. Staunton *et al.*, "Chemosensitivity prediction by transcriptional profiling," *Proc Natl Acad Sci U S A*, vol. 98, no. 19, pp. 10 787–92, 2001.

[37] D. Titterington, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Distributions*. New York, Wiley, 1985.

[38] Y. Wang, *Statistical Techniques for Network Security: Modern Statistically based Intrusion Detection and Protection*. IGI Global, 2008.