A Recursive Method for Discriminative Mixture Learning

Minyoung Kim MIKIM@CS.RUTGERS.EDU
Vladimir Pavlovic VLADIMIR@CS.RUTGERS.EDU
Department of Computer Science, Rutgers University, Piscataway, NJ 08854 USA

Abstract

We consider the problem of learning density mixture models for classification. Traditional learning of mixtures for density estimation focuses on models that correctly represent the density at all points in the sample space. Discriminative learning, on the other hand, aims at representing the density at the decision boundary. We introduce a novel discriminative learning method for mixtures of generative models. Unlike traditional discriminative learning methods that often resort to computationally demanding gradient search optimization, the proposed method is highly efficient as it reduces to generative learning of individual mixture components on weighted data. Hence it is particularly suited to domains with complex component models, such as hidden Markov models or Bayesian networks in general, that are usually too complex for effective gradient search. We demonstrate the benefits of the proposed method in a comprehensive set of evaluations on time-series sequence classification problems.

1. Introduction

Generative probabilistic models such as Bayesian networks (BNs) are an attractive choice in a number of data-driven modeling tasks. Among their advantages are the ability to easily incorporate domain knowledge, factorize complex problems into self-contained models, handle missing data and latent factors, and offer interpretability to results. While such models are implicitly employed for joint density estimation, they have recently been shown to also yield performance comparable to sophisticated discriminative classifiers such as SVMs and C4.5 (Friedman et al., 1997).

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

In the classification setting, maximizing a conditional likelihood (CML) is known to achieve better classification performance than the traditional Maximum Likelihood (ML) fitting in a variety of situations (Greiner & Zhou, 2002; Pernkopf & Bilmes, 2005). Unfortunately, the CML optimization problem is, in general, complex with non-unique solutions. Typical CML learning resorts to gradient-based numerical optimization methods. Despite the improved prediction performance, the gradient search makes standard approaches computationally demanding.

Effective use of generative models for classification often requires estimation of the model's structure as well as parameters. Intractability of the structure search may sometimes be avoided by an ensemble-based approach (Jing et al., 2005). However, the resulting model is *not* a generative model which may limit its domain of applications to classification tasks only.

In this paper, we focus on the class of density *mixture models*. A mixture model has a potential to yield superior classification performance to a single BN model, as well as serve as a rich density estimator. Traditional discriminative mixture learning commonly relies on the same gradient search used for single BN models (e.g., (Beaufays et al., 1999)). This paper formulates an efficient and theoretically sound approach to discriminative mixture learning that avoids the parametric gradient optimization.

The proposed method exploits the properties of mixtures to alleviate the complex learning task. In a greedy fashion, the mixture components are added recursively while maximizing the conditional likelihood. More specifically, at each iteration a new mixture component f is found that, when added to the current mixture F, maximally decreases the conditional loss.

Formulated as a functional gradient boosting, the procedure yields data weights with which the new component f will be learned. Our particular weighting scheme effectively emphasizes the data points on the decision boundary, a desirable property for success-

ful *classification*. At the same time, it focuses on the insufficiently modeled points, a characteristic of traditional density estimators and a property useful in general *data fitting*.

A crucial benefit of this method is efficiency: finding a new f requires ML learning on the weighted data, a tractable task for a large family of distributions. Thus this approach is particularly suited to domains with complex component models (e.g., hidden Markov models (HMMs) in time-series classification) that are usually too complex for effective gradient search. In addition, the recursive approach is amenable to optimal order estimation and lower sensitivity to initial parameter choices.

The paper is organized as follows: The previous approaches to discriminative learning of generative models are discussed in Sec. 2. Our proposed algorithm is described in Sec. 3 with comparison to related work in Sec. 4. In the experimental evaluation in Sec. 5, our algorithm is compared with many standard methods in an extensive set of sequence classification problems.

2. Background

We consider the supervised classification problem to predict a class label $c \in \{1, \ldots, K\}$ for a point with the attribute a which is either vector-valued or structured like a sequence. Let $f(c, a)$[1] denote a BN with a class variable c and attribute variables a. As a generative model, it is natural to model it by the (multinomial) class prior $f(c)$ and the class conditional densities $f(a|c) = f_c(a)$. For example, $f_c(a)$ could be a Gaussian for the continuous vector a, or a product of independent multinomials for a discrete vector a. $f_c(a)$ may also include many latent variables (e.g., in the sequence classification where a is a sequence of measurement, $f_c(a)$ can be modeled as the HMM with state variables hidden).

As a classifier, the class prediction of a new measurement a can be accomplished by the MAP decision rule: $c^* = arg\max_c f(c|a)$. Given training data $D = \{(c^i, a^i)\}_{i=1}^n$, learning a joint density $f(c, a)$ that minimizes the prediction error is the main issue. The traditional ML learning optimizes the data joint likelihood, $\sum_{i=1}^n \log f(c^i, a^i)$. However, ML does not necessarily yield optimal prediction performance unless we are given not only the correct model structure but also a large number of train samples. We briefly review several learning methods that have been (empirically) shown to yield better prediction performance than ML.

[1] We use $f(c, a)$ to represent either a BN or a likelihood at a data point (c, a) interchangeably.

2.1. Conditional Likelihood Maximization

The conditional log-likelihood (CLL) objective function for $f(c, a)$ is:

$$CLL = \sum_{i=1}^{n} \log f(c^i|a^i) = \sum_{i=1}^{n} \Big[\log f(c^i, a^i) - \log f(a^i) \Big].$$

CLL is directly related with the prediction task. However, CLL optimization in general does not admit closed-form solutions for most generative models. One typically maximizes it using a gradient-based search. The gradient w.r.t. the parameters θ of $f(c, a)$ is:

$$\frac{\partial CLL}{\partial \theta} = \sum_{i=1}^{n} \left[\frac{\partial}{\partial \theta} \log f(c^i, a^i) - \frac{\partial}{\partial \theta} \log f(a^i) \right]. \quad (1)$$

The first term, the gradient of the *joint* log-likelihood, is straightforward[2] to evaluate if $f(c, a)$ has no hidden variables. The presence of hidden variables z in $f(c, a)$, on the other hand, trivially results in the expectation of the gradient of the joint (including z) log-likelihood:

$$\begin{aligned} \frac{\partial}{\partial \theta} \log f(c, a) &= \frac{1}{f(c, a)} \frac{\partial}{\partial \theta} \int_z f(c, a, z) \\ &= E_{f(z|c,a)} \left[\frac{\partial}{\partial \theta} \log f(c, a, z) \right]. \quad (2) \end{aligned}$$

The second term of Eq-(1), the derivative of the *measurement* log-likelihood, is the expectation (over c) of the joint log-likelihood (in the same manner as Eq-(2) by treating c as hidden). That is, $\frac{\partial}{\partial \theta} \log f(a) = E_{f(c|a)}[\frac{\partial}{\partial \theta} \log f(c, a)]$. Several previous works demonstrate that CML outperforms ML when the model structure is suboptimal (Greiner & Zhou, 2002; Pernkopf & Bilmes, 2005). However, the computational overhead of the gradient-based numerical search is highly demanding especially for complex models such as HMMs and general BN structures.

2.2. Boosted Bayesian Networks (BBN)

(Jing et al., 2005) proposed a very efficient discriminative learning method for BNs. They treat $f(c, a)$ as a (weak) hypothesis, namely $c = h(a) = arg\max_c f(c|a)$, in a boosting (Freund & Schapire, 1995) framework. For each stage, AdaBoost's weights w on data (c, a) are used to learn the next hypothesis (BN) via *weighted* ML learning: $arg\max_f \sum_{i=1}^n w^i \cdot \log f(c^i, a^i)$. This approach has been shown to inherit certain benefits from AdaBoost such as good generalization by max-margin. However, the resulting ensemble cannot be simply interpreted as a generative model since the learned BNs are just weak classifiers to be combined for the classification task.

[2] We assume that all the conditional densities in the BN belong to the exponential families.

3. Boosted Mixture Learning

Let $F(c,a)$ denote a mixture of BNs, that is, $F(c,a) = \sum_{m=1}^{M} \alpha_m f_m(c,a)$, where $\alpha_m \geq 0$ and $\sum_m \alpha_m = 1$. Note that each component of the mixture is a BN $f_m(c,a)$. The mixture model is learned in a greedy recursive (boosted) manner: at each stage we add a new BN component $f(c,a)$ to the current mixture so that it optimizes a certain objective. Two potential advantages of this approach over the standard EM-based mixture learning are (1) the lack of need for a predetermined mixture order M, and (2) the decreased sensitivity to the initial parameter choice.

Formally, for a given objective functional $J(F)$ for the mixture F, we search for a new component f such that when we replace F with $((1-\epsilon)F+\epsilon f)$ for some small positive ϵ, $J((1-\epsilon)F+\epsilon f)$ is maximally increased. Due to the convex combination constraint of a mixture, f should make the projection of the functional gradient of $J(F)$ onto $(f-F)$ maximized. It results in the optimization problem, $f^* = arg\max_f < f-F, \nabla J(F) >$, which is equivalent to:

$$f^* = arg\max_f \sum_{i=1}^{n} w(c^i,a^i)\cdot f(c^i,a^i), \qquad (3)$$

where $w(c,a) = \nabla_{F(c,a)} J(F) = \partial J(F)/\partial F(c,a)$. Thus $\nabla_{F(c,a)} J(F)$ serves as a weight for the data point (c,a) with which the new f will be learned.

When the objective is the joint log-likelihood (generative learning), $J_{Gen}(F) = \sum_{i=1}^{n} \log F(c^i,a^i)$, the functional gradient is $\partial J_{Gen}(F)/\partial F(c,a) = 1/F(c,a)$ yielding the generative data weight $w_{Gen}(c,a) = 1/F(c,a)$ for (c,a). On the other hand, the conditional log-likelihood objective, $J_{Dis}(F) = \sum_{i=1}^{n} \log F(c^i|a^i)$, gives birth to the discriminative mixture model.

3.1. Discriminative Mixture Learning

The discriminative objective to be maximized is:

$$J_{Dis}(F) = \sum_{i=1}^{n} \log F(c^i|a^i) = \sum_{i=1}^{n} \log \frac{F(c^i,a^i)}{F(a^i)}. \qquad (4)$$

The functional gradient of Eq-(4) for the point (or index of dimension) (c^i,a^i) can be derived as:

$$\frac{\partial J_{Dis}(F)}{\partial F(c^i,a^i)} = \frac{\partial}{\partial F(c^i,a^i)} \log \frac{F(c^i,a^i)}{F(a^i)} = \frac{F(\neg c^i|a^i)}{F(c^i,a^i)}, \qquad (5)$$

where $F(\neg c^i|a^i) = \sum_{c\neq c^i} F(c|a^i) = 1 - F(c^i|a^i)$. The discriminative data weight for (c,a) is $w_{Dis}(c,a) = (1-F(c|a))/F(c,a)$.

The discriminative weight indicates that the new f is learned by weighted data proportional to $(1-F(c^i|a^i))$, at the same time, inversely proportional to $F(c^i,a^i)$. Hence the data points *unexplained by the model*, i.e., $F(c^i,a^i) \to 0$, and *incorrectly classified* by the current mixture, i.e., $(1-F(c^i|a^i)) \to 1$, are focused on in the next stage. This is an intuitively appealing argument. In contrast, the generative mixture model, would only focus on unexplained points with weights $1/F(c^i,a^i)$.

Once the optimal component f^* has been selected, its optimal contribution to the mixture α^* can be obtained as:

$$\alpha^* = arg\max_{\alpha\in[0,1]} \sum_{i=1}^{n} \log\Big(\frac{(1-\alpha)F(c^i,a^i)+\alpha f^*(c^i,a^i)}{(1-\alpha)F(a^i)+\alpha f^*(a^i)}\Big). \qquad (6)$$

The complete discriminative mixture learning algorithm is outlined in Algorithm 1. Selection of the first component can be done using the ML learning. Note that the choice of the initial model is not very critical as we keep enhancing (initially weak) mixture recursively. Optimization in Eq-(3) is, by taking a log(), a *log-of-sum* (instead of *sum-of-log*), which can be done via a lower bound maximization technique, by recursively completing a few iterations of ML learning of f on the weighted data with the weights $q_i = w_{Dis}(c^i,a^i)\cdot f(c^i,a^i)$. Optimal α can be found using any line search method.

The time complexity of the discriminative mixture learning is of the order $O(M\cdot(C_0\cdot N_{ML}+N_{LS}))$ where N_{ML} stands for the complexity of the ML learning, C_0 is the number of iterations of the the recursive ML, and N_{LS} is the complexity of the line search. In practice, ML recursions are dominant resulting in the overall complexity $O(MC_0N_{ML})$ with $C_0 \approx 1$. Hence, the discriminative mixture learning algorithm complexity is a constant factor of the simple generative learning of the base model on weighted data.

To illustrate the behavior of this discriminative algorithm consider a simple example in Figure 1 of two classes, each modeled with a mixture of three Gaussians from which 200 samples were drawn (top). The central lobe of each class models the majority of samples. Of the two side lobes, one is irrelevant for classification while the other carries crucial samples. Suboptimal to the true model, our component BN f is assumed to have a *single* Gaussian for each class.

The ML learned initial mixture component $f_1(c,a)$ models the majority of the samples (shown in the bottom). In the middle, we depict the next stage weight distributions determined by $f_1(c,a)$ for two learning criteria. In the discriminative learning, the points close to the boundary are incorrectly classified by $f_1(c,a)$ and receive high weights $w_{Dis}(c,a)$ while the unex-

Algorithm 1 Discriminative Mixture Learning.

Input: A set of samples $D = \{(c^i, a^i)\}_{i=1}^n$.
Output: Mixture $F(c,a) = \sum_m \alpha_m f_m(c,a)$.
Select initial f.
$F \leftarrow f$.
for $m = 2, 3, \ldots$ **do**
  Select f^* by solving Eq-(3) with $w = w_{Dis}$.
  Select α^* by solving Eq-(6).
  Update $F \leftarrow (1-\alpha^*)F + \alpha^* f^*$.
end for

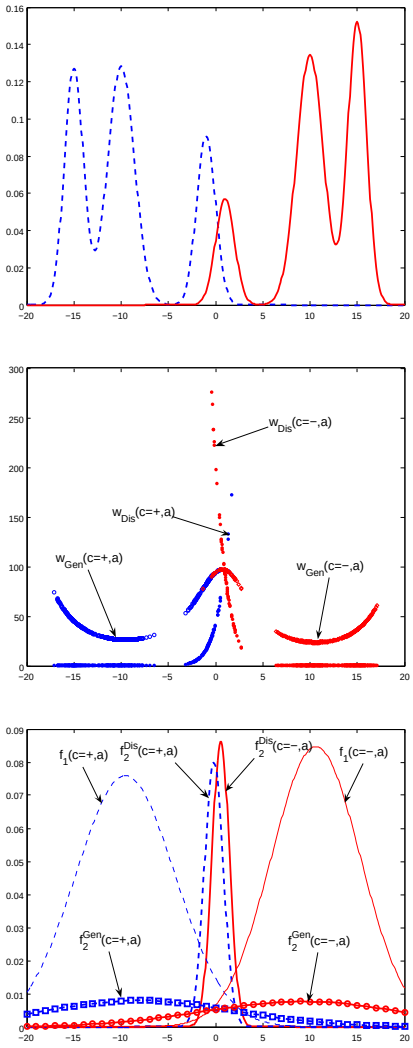

Figure 1. Data is generated by the distributions in the top panel (+ class in blue/dashed and − class in red/solid). The middle panel shows weights for the second component, both discriminative $w_{Dis}(c,a)$ and generative $w_{Gen}(c,a)$. The bottom panel displays the individual mixture components of the learned models. Generatively learned component $f_2^{Gen}(c,a)$ is contrasted to the discriminatively learned one, $f_2^{Dis}(c,a)$.

plained points away from the boundary are not considered because of their irrelevance for classification. The new mixture components will now be added close to the decision boundary (f_2^{Dis} in the bottom). On the other hand, in the generative learning, higher weights are assigned to unexplained samples ($w_{Gen}(c,a)$ in the middle), which selects the component corresponding to the main lobes, away from the boundary, hence obtaining a less discriminative mixture (f_2^{Gen} in the bottom).

4. Related Work

In automatic speech recognition, the discriminative parameter learning of HMMs and its benefit have been studied extensively (Woodland & Povey, 2002). Most methods adopt minimum classification error or CML (also called maximum mutual information) objectives, however, their optimization algorithms are based on gradient search or complex update schemes. Recently, Sha and Saul (2007) have introduced an alternative discriminative density estimator based on max-margin, formulating a convex program.

Prior approaches to estimation of mixtures of BNs have emerged in recent years (Thiesson et al., 1998; Rosset & Segal, 2002; Meek et al., 2002). Our recursive boosting algorithm for discriminative mixture learning is based on the functional gradient optimization of convex additive models. While similar gradient approaches have been introduced in the past (Friedman, 1999; Mason et al., 1999), they only provided heuristic methods for the component search or did not focus on mixtures of generative models. In (Pavlovic, 2004), a mixture fitting problem, reduced to the joint log-likelihood cost functional optimization in the supervised setting, was solved in a non-heuristic way. Our algorithm generalizes its framework to the classification setting with an appropriate data weighting schemes for the discriminative cost functional.

5. Experiments

To evaluate the utility of the proposed mixture-based classification method we conduct experiments on synthetic and real data. Here we focus on the task of classifying structured measurements (i.e., sequences). This is, in general, more difficult than the static multivariate data classification where the standard gradient-based methods such as CML are not preferred due to the complex model structures.

In the following experiments we use Gaussian-emission HMMs (GHMMs) to model the class conditional densities $f_c(a)$ for the real multivariate sequence a. The competing methods are denoted as: (1) **ML** (ML

learning for $f(c, a)$), (2) **CML** (Sec. 2.1), (3) **BBN** (Sec. 2.2), (4) **BxML** (generative mixture learning), and (5) **BxCML** (discriminative mixture learning).

5.1. Synthetic Experiment

2D sequences are generated by the following process: The class-1 is composed of two GHMMs, f_{1E} and f_{1H}, and the class-2 is another mixture of two GHMMs, f_{2E} and f_{2H}. The parameters of f_{1E} and f_{2E} are chosen in a way that they generate sequences looking very different (*easy* to distinguish). On the other hand, f_{1H} and f_{2H} generate sequences similar to each other (*hard* to classify), thus lying on the classification boundary. The example sequences are depicted in Figure 2. Note that our base model $f(c, a)$ has a *sub-optimal* structure since it has a single GHMM for each class.

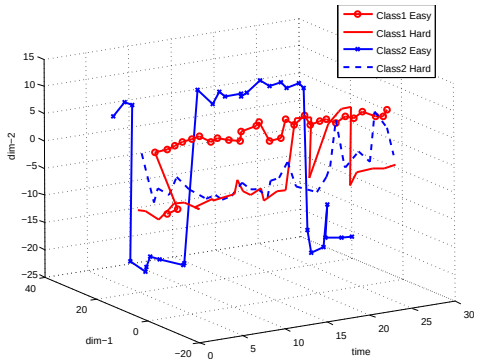

Figure 2. Example sequences generated by true model.

The first component of the mixture models (**BxML** and **BxCML**) is chosen as the **ML** model. The maximum number of iterations of **BxML** and **BxCML** is set as 4, however, **BxCML** often stops earlier when the CLL score reaches a value close to 0. We also run **BBN** for 10 iterations, sufficient for convergence.

The average test errors and the joint/conditional log-likelihood scores on test data are shown in Table 1. **BxCML** has the lowest error, *boosting* the incorrect base model structure effectively. Overall, the methods that utilize a discriminative objective tend to perform better than generative counterparts. **BxCML** also improves the joint log-likelihood score over that of **ML**, implying that the discriminative mixture model can still enjoy the benefits of generative models, such as the richness in synthesis.

5.2. Experiments on Real Data

We next demonstrate the benefits of the proposed method in a comprehensive set of evaluations on real-world time-series sequence classification problems.

Table 1. Average test errors (%), log-likelihoods (LL), and conditional log-likelihoods (CLL) on test data. **BBN**, a non-generative classifier, does not have LL or CLL.

	Test Error	LL on Test	CLL on Test
ML	19.60	-165.21	-1.97
CML	9.80	-174.99	-1.11
BBN	6.40	N/A	N/A
BxML	4.20	-139.12	-0.44
BxCML	0.60	-154.62	-0.02

The 5 classification problems from 4 datasets are described in Sec. 5.2.3. All the experimental results are summarized in Table 2 and Figure 3 with the discussion in Sec. 5.2.4. In the next section we briefly review two competing discriminative approaches (SVMs and Nearest Neighbors (NN)) that we compare against in our experiments. Sec. 5.2.2 outlines our treatment of multi-class problems.

5.2.1. FSVM and 1-NN/DTW

One way to approach the sequence classification problem relies on between-sequence distance measures. A central issue is the task of defining the distance measure (kernel for SVM and Euclidean distance for NN) between pairs of possibly *unequal-length* sequences.

SVM with Fisher kernel (FSVM): The Fisher kernel between two sequences a and a' is defined as the RBF evaluated on the distance between their Fisher scores with respect to the underlying generative model. More specifically, assuming binary classification[3], $k(a, a') = e^{-||U_a - U_{a'}||^2/(2\sigma^2)}$, where $U_a = \nabla_\theta \log f_{c=+}(a)$. $f_{c=+}(a)$ is usually learned by ML with the examples of the positive class only. The RBF scale σ^2 is determined as a median distance between the Fisher scores corresponding to the training sequences in the positive class and the closest Fisher score from the negative class in the train data (Jaakkola et al., 1999). We use SVM with the Fisher kernel. The SVM hyperparameters are selected by a cross validation.

NN with dynamic time warping (1-NN/DTW): For two unequal-length sequences, the dynamic time warping (DTW) finds the best warping path that minimize the Euclidean distance of aligned sequences using dynamic programming. With the warped Euclidean distance measure, we employ 1-NN to classify new sequences. We include 1-NN only since we have verified that the choice of $k \geq 2$ in k-NN rarely impacts on the classification performance in our experiments.

[3]The multi-class problems can be reduced to many binary problems. See Sec. 5.2.2.

5.2.2. Treating Multi-class Problems

In multi-class settings we apply both direct multi-class solutions and binarization. For **FSVM** we will ignore direct multi-class solutions due to difficulties in direct treatment. The binarization is usually done in either one-vs-others or one-vs-one manner. In the one-vs-others setting multi-class labels are predicted using the *winner-takes-all* (WTA) strategy from the outputs of the binarized problems. In the one-vs-one setting we employ the *pairwise coupling* (PWC) of (Hastie & Tibshirani, 1998). Note that for **FSVM**, the SVM outputs have to be transformed to Platt's probabilistic outputs (Platt, 1999) before we apply PWC[4] (Duan & Keerthi, 2003). In the notation, we denote SVM one-vs-one PWC by **FSVM(PWC)**, while **FSVM(WTA)** for one-vs-others.

For the generative models, we evaluate both direct multi-class solutions and the PWC for one-vs-one. For instance, for CML, we denote the former by **CML**, while the latter by **CML(PWC)**. For **BBN**, we used (1) direct multi-class treatment (AdaBoost.M1) denoted by **BBN**, and (2) one-vs-one binarization and *max-win-vote* denoted by **BBN(MWV)**.

5.2.3. Datasets

A. Gun/Point: The task is to distinguish whether *gun is drawn* or *finger is pointed* (Keogh & Folias, 2002). The motions are represented by 1D sequences recording the x-coordinates of the centroid of the right hand of a subject. This is the only dataset of the equi-length (150) sequences. The evaluation is performed by 10-fold cross validation.

B. Australian Sign Language (ASL): This dataset contains about 100 signs generated by 5 signers with different levels of skills (Hettich & Bay, 1999). In this experiment, we consider only 10 selected signs (e.g., "hello", "sorry", etc.). The sequences have features, corresponding to the hand position, hand orientation, finger flexion, and more. The sequence lengths are very diverse ranging from 17 to 196. We formulate binary classification problems distinguishing one sign from another, facing 45 ($= \binom{10}{2}$) problems. For each problem, 40 samples (20 from each sign) are gathered and the leave-1-out test is performed.

C. Georgia-Tech Gait (GT Gait): We also test the proposed method on the human gait recognition problem. From the database (Tanawongsuwan & Bobick, 2003), we take sequences of 5 subjects with 4 different walking speeds. The goal is to recognize *subjects* regardless of their walking speeds, that is, a 5-class problem. The original dataset provides high-quality 3D motion capture features on which most of competing models perform equally well. To make the classification task more difficult we consider two modification: (1) From the original 1-cycle sequences, we take sub-sequences randomly. (2) The features related only to the *lower* body part are used. The evaluation is done by 10-fold cross validation.

D. USF Human ID Gait Data: The database[5] consists of about 100 subjects walking in the elliptical paths periodically in front of the cameras. We focus on the task of motion-based subject identification. From the processed human silhouette video frames we computed the 7^{th} order Hu moments which are translation and rotation invariant descriptors of binary images. We randomly choose 7 humans from the database represented by 16 sequences. We consider the following two problem settings: **(1) Set1 (Distinguish two subjects)**: We select sequences of *only two* humans, and distinguish the two subjects. Thus we have $21(= \binom{7}{2})$ binary classification problems. **(2) Set2 (Recognize all subjects)**: We classify all 7 human IDs. This is a more difficult 7-class problem. Both sets are evaluated using leave-1-out validation.

5.2.4. Discussion

Results of our experiments on the four datasets are summarized in Table 2. The results suggest that the discriminatively trained mixture model, **BxCML**, is among the class of best-performing models, performing on par or better than state-of-the-art methods such as **FSVM**. This points to the critical benefit of **BxCML** that couples an increased modeling capacity of mixture models with the discriminative learning objective.

For the **Gun/Point** dataset, with binary class equi-length sequences, the purely discriminative classifiers (**FSVM** and **1-NN/DTW**) outperform traditional generative models trained both generatively and discriminatively (**ML** and **CML**). On the other hand, for the **ASL** dataset which contains diverse-length sequences, all generative models yield superior performance to example-based classifiers. This is possibly due to the sensitivity of kernel methods (**FSVM** and **1-NN/DTW**) to the choice of kernel parameters, which becomes a critical but difficult-to-solve problem for datasets with diverse-length sequences.

Generative models, on the other hand, naturally account for varying-length sequences. However, their representational power may need to be increased via

[4] This transformation is not necessary for the generative probabilistic models.

[5] Available at http://figment.csee.usf.edu/GaitBaseline.

the mixture modeling formalism in order to account for variability not captured by traditional HMMs, as suggested by the good performance of mixture models in the **Gun/Point**.

It is important to note, however, that despite the representational capacity of mixtures the role of proper optimization objective can be crucial. For the **GT Gait** dataset, we can see that generative models with discriminative objectives (**CML** and **BxCML**) are significantly better than those with generative objectives (**ML** and **BxML**).

Table 2. Test errors (%): For the datasets evaluated with cross validation, the averages and the standard deviations are included. The others contain average leave-1-out test errors. "–" means *redundant* since a multi-class method is to be applied for binary class data. The boldfaced numbers indicate the lowest, within the margin of significance, test errors for a given dataset. The mixture orders for the boosting algorithms are determined by cross validation, where they are usually small (fewer than 10 components).

	Gun/ Point	ASL	GT Gait	USF Set1	USF Set2
ML	36.22 ± 9.62	8.67	11.50 ± 4.78	20.24	55.36
ML (PWC)	–	–	11.50 ± 4.78	–	55.36
CML	26.06 ± 5.23	**5.45**	**3.38 ± 3.68**	17.11	50.89
CML (PWC)	–	–	**3.63 ± 3.51**	–	**39.29**
BBN	28.78 ± 13.75	**4.90**	10.13 ± 3.61	17.11	55.36
BBN (MWV)	–	–	**3.50 ± 3.05**	–	42.86
BxML	**19.28 ± 6.15**	6.33	11.87 ± 5.11	19.35	48.21
BxML (PWC)	–	–	14.25 ± 4.90	–	50.00
BxCML	**17.28 ± 5.67**	**5.18**	5.75 ± 2.78	**13.84**	54.46
BxCML (PWC)	–	–	6.87 ± 4.09	–	**35.71**
FSVM (PWC)	22.67 ± 6.58	10.90	7.12 ± 4.17	**12.95**	**39.29**
FSVM (WTA)	–	–	**2.87 ± 2.29**	–	44.64
1-NN/ DTW	22.33 ± 5.75	12.06	8.38 ± 3.68	22.17	54.46
Avg	24.66 ± 7.54	7.64	7.75 ± 3.88	17.54	48.15

Improved performance of **CML** compared to **BxML** implies that the discriminative learning of models with even inferior structures can yield superior classifiers. Overall, comparison of **BxCML** and **BxML** suggests that the impact of discriminative learning of the mixtures can be significant.

BBN has the potential similar to **BxCML** to focus on the decision boundary modeling. In our experiments, **BxCML** is never inferior to **BBN**, perhaps pointing to deficiencies in the approximation step of the weighted ML optimization in **BBN**. On the other hand, the weighted ML training in the **BxCML** approach does not involve a similar approximation assumption. Additionally, **BxCML** results in a completely generative model $F(c, a)$ that could possess attractive data-synthesis properties, as indicated by our result on the synthetic data.

While the discriminative mixture model outperforms other approaches, multi-class problems, as indicated by **USF Set2**, raise an important modeling issue. In particular, **USF Set2** suggests that the binarization yields better performance than direct multi-class treatment for some models. This issue does not daunt generatively learned generative models (**ML**/**BxML**) as the optimization of the *joint* likelihood implies no discrimination between the true and the competing class variables. **BxCML**, due to discriminative learning, may exhibit a large difference in binarization and direct multi-class treatment. This behavior is due to the numerator of the weight, $(1 - F(c|a))$, which penalizes the complement classes $(\neg c)$ *equally* for the incorrectly predicted point (c, a). Similar issues have been observed in binary class AdaBoost approaches and will be addressed in our future work.

6. Conclusions

We introduced a novel discriminative method for learning mixtures of generative models. Unlike traditional approaches to discriminative learning of generative models, the proposed method is highly computationally efficient. This makes the approach suitable for domains described by complex generative models and settings such as the spaces of time-series sequences.

We have shown, through a comprehensive set of evaluations, that a mixture learned discriminatively is not only superior to a single generative model, but also comparable in performance to ensembles of discriminative models. Moreover, the mixture model continues to enjoy other benefits of generative models such as the potential for powerful synthesis of data which will be explored in our future work.

In this paper, although we focused on the parameter learning to avoid expensive structure search, the discriminative mixture learning may include and benefit from the structure learning. Methods such as the structural EM algorithm can readily become a part of our framework, adjusting the model structure at each boosting stage.

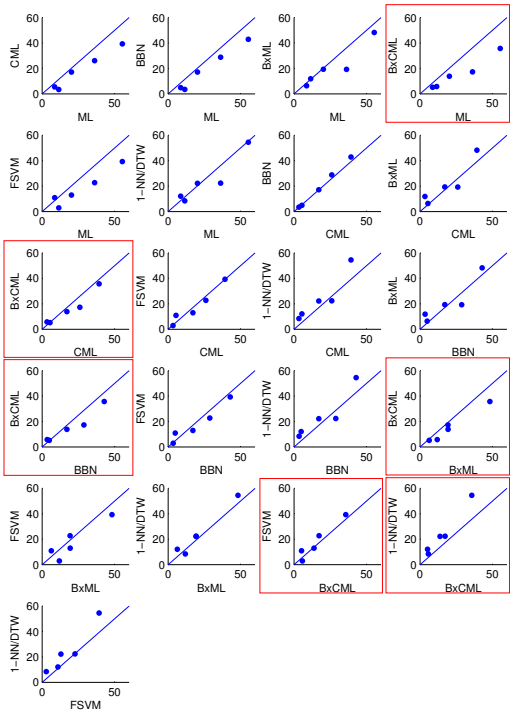

Figure 3. Test error scatter plots comparing 7 models from Table 2. Each point corresponds to one of the 5 classification problems. For instance, congregation of points below the main diagonal in the **BxCML** vs. **ML** case suggests that **BxCML** outperforms **ML** in most of the experimental evaluations. The (red) rectangles indicate the plots comparing **BxCML** with others.

References

Beaufays, F., Weintraub, M., & Konig, Y. (1999). Discriminative mixture weight estimation for large Gaussian mixture models. *International Conference on Acoustics, Speech, and Signal Processing.*

Duan, K., & Keerthi, S. (2003). Which is the best multiclass SVM method? An empirical study. *Neural Information Processing Systems.*

Freund, Y., & Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. *European Conference on Computational Learning Theory.*

Friedman, J. (1999). Greedy function approximation: a gradient boosting machine. Technical Report, Dept. of Statistics, Stanford University.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning.*

Greiner, R., & Zhou, W. (2002). Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. *Annual Meeting of the American Association for Artificial Intelligence.*

Hastie, T., & Tibshirani, R. (1998). Classification by pairwise coupling. *Neural Information Processing Systems.*

Hettich, S., & Bay, S. D. (1999). The UCI KDD Archive (http://kdd.ics.uci.edu). Irvine, University of California, Information and Computer Science.

Jaakkola, T., Diekhans, M., & Haussler, D. (1999). Using the Fisher kernel method to detect remote protein homologies. *International Conference on Intelligent Systems for Molecular Biology.*

Jing, Y., Pavlovic, V., & Rehg, J. M. (2005). Efficient discriminative learning of Bayesian Network Classifier via boosted augmented Naive Bayes. *International Conference on Machine Learning.*

Keogh, E., & Folias, T. (2002). University of California at Riverside Time Series Data Mining Archive (http://www.cs.ucr.edu/∼eamonn/TSDMA).

Mason, L., Baxter, J., Bartlett, P., & Frean, M. (1999). Functional gradient techniques for combining hypotheses. *In Advances in Large Margin Classifiers, MIT Press.*

Meek, C., Thiesson, B., & Heckerman, D. (2002). Staged mixture modelling and boosting. *Uncertainty in Artificial Intelligence.*

Pavlovic, V. (2004). Model-based motion clustering using boosted mixture modeling. *IEEE Conference on Computer Vision and Pattern Recognition.*

Pernkopf, F., & Bilmes, J. (2005). Discriminative versus generative parameter and structure learning of Bayesian Network Classifiers. *International Conference on Machine Learning.*

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *In Advances in Large Margin Classifiers, MIT Press.*

Rosset, S., & Segal, E. (2002). Boosting density estimation. *Neural Information Processing Systems.*

Sha, F., & Saul, L. K. (2007). Large margin hidden Markov models for automatic speech recognition. *Neural Information Processing Systems.*

Tanawongsuwan, R., & Bobick, A. (2003). Performance analysis of time-distance gait parameters under different speeds. *International Conference on Audio and Video Based Biometric Person Authentication.*

Thiesson, B., Meek, C., Chickering, D. M., & Heckerman, D. (1998). Learning mixtures of DAG models. *Uncertainty in Artificial Intelligence.*

Woodland, P., & Povey, D. (2002). Large scale discriminative training of hidden Markov models for speech recognition. *Computer Speech and Language.*