Pattern Recognition Letters
journal homepage: www.elsevier.com

Maximum Margin Hidden Markov Models for Sequence Classification

Nikolaus Mutsam[a], Franz Pernkopf[a,**]

[a]*Graz University of Technology, Laboratory of Signal Processing and Speech Communication, Graz, Austria*

ABSTRACT

Discriminative learning methods are known to work well in pattern classification tasks and often show benefits compared to generative learning. This is particularly true in case of model mismatch, i.e. the model cannot represent the true data distribution. In this paper, we derive discriminative maximum margin learning for hidden Markov models (HMMs) with emission probabilities represented by Gaussian mixture models (GMMs). The focus is on single-label sequence classification where the margin objective is specified by the probabilistic gap between the true class and the most competing class. In particular, we use the extended Baum-Welch (EBW) framework to optimize this probabilistic margin embedded in a hinge loss function. Approximations of the margin objective and the derivatives are necessary. In the experiments, we compare maximum margin HMMs to generative maximum likelihood and discriminative conditional log-likelihood (CLL) HMM training. We present results of classifying trajectories of handwritten characters, Australian sign language data, digits of speech data and UCR time-series data. Maximum margin HMMs outperform in many cases CLL-HMMs. Furthermore, maximum margin HMMs achieve a significantly better performance that generative maximum likelihood HMMs.

1. Introduction

The analysis of time-series or sequential data covers a wide field of applications, such as for instance speech analysis Rabiner (1989); Pernkopf *et al.* (2014), financial mathematics Cao & Tay (2003) and weather forecasts Campbell & Diebold (2005). In a time-series the samples are dependent on previous samples in the sequence. Modelling the dependency on all previous samples is computationally intractable, therefore usually only the neighboring context is modeled. In the simplest case, only the most recent sample is considered leading to the first-order Markov model assumption. One of the simplest and most widely used models for time-series processing in the past decades is the HMM.

There are two classical learning paradigms in the machine learning community: generative learning and discriminative learning Bahl *et al.* (1986); Bishop (2006); Jebara (2001). Generative learning aims to recover the data distribution from a finite set of samples. This is achieved by optimizing the data-likelihood or the data-posterior-probability, in the case of a model-prior for regularization. Maximum likelihood estimation (MLE) is usually used to generatively learn the classifier. Generatively optimized models facilitate to generate samples by the model having the same statistical distribution as the training data. Discriminative learning methods such as conditional log-likelihood (CLL) Ng & Jordan (2002); Pernkopf & Bilmes (2010) or maximum margin learning Schölkopf & Smola (2001); Burges (1998); Pernkopf *et al.* (2012) more directly represent aspects that are important for classification accuracy, i.e. a prediction function is optimized, predicting output variables (classes) from a set of input variables (features). As shown in Ng & Jordan (2002) generative learning of naive Bayes models reach faster their asymptotic generalization performance with respect to samples size compared to discriminative training. If the model does not match the true underlying distribution, discriminatively learned models usually obtain better asymptotic performance for a sufficiently large set of training samples.

Discriminative training of HMMs celebrated success over the years in speech processing and in this context the extended

**Corresponding author: Tel.: +43-316-873-4436; fax: +43-316-873-104436;
e-mail: pernkopf@tugraz.at (Franz Pernkopf)

Baum-Welch optimization algorithm Gopalakrishnan *et al.* (1991) has been introduced. In the seminal work of Bahl *et al.* (1986), discriminative HMM parameter learning based on the maximum mutual information (MMI) criterion has been proposed.[1] The goal is to maximize the posterior probability of the transcriptions given the speech utterances. This results in significantly better recognition rates compared to conventional *generative* MLE learning. A decade later, the minimum classification error (MCE) has been proposed Juang *et al.* (1997) where the aim is to minimize the sentence classification error, i.e. risk, on the training set. The main advantage of MCE is that in MMI training only the posterior distribution is optimized which does not necessarily result in an optimal classification performance. MCE optimizes the empirical risk and is therefore susceptible to overfitting. The expected risk on unseen test data also depends on the generalization ability of the model. From the SVM literature Schölkopf & Smola (2001); Burges (1998) it is well-known that the optimization of the margin leads to good generalization properties cf. Vapnik-Chervonenkis (VC) dimension and PAC bounds Vapnik (1998). Hence, margin optimization has been proposed for HMMs in Taskar *et al.* (2004); Sha & Saul (2007b,a); Saon & Povey (2008); Heigold *et al.* (2008, 2010, 2012). Most of this work is in the area of speech processing, e.g. phoneme recognition, with focus on *multi-label* sequence classification, i.e. the observation sequence is assigned to *multiple* labels, i.e. a label sequence. This usually leads to a margin objective Sha & Saul (2007b) composed of two terms:

1. A probabilistic term measuring the gap between log-probabilities of the data of the target label sequence and the most competing sequence;
2. A term measuring the Hamming distance between two label sequences, the target label sequence and the competing label sequence.

In this objective, the gap between the log-probabilities in the first term should be larger or equal to the Hamming distance of the second term. In Heigold *et al.* (2010, 2012), the margin term has been incorporated in a unified training criterion. Special cases of this criterion are MMI or MCE amongst others.

In contrast, we perform *single-label* classification of sequences, i.e. the observation sequence is assigned to a *single* class label. We derive discriminative maximum margin learning for HMMs with emission probabilities represented by GMMs for single-label sequence classification. Related to the margin criterion for multi-label classification from above the distance measure between the competing and the target/true label sequences is neglected. Here, we use the probabilistic definition of the margin Guo *et al.* (2005); Pernkopf *et al.* (2012), i.e. the first term in the objective above. The log-probability of the sequence of the true class should lie at least some *distance* away from the log-probabilities of the competing classes, i.e. the second term in the objective above is set to a constant. Our probabilistic margin is embedded in a hinge-loss function and optimized by the EBW algorithm. The EBW algorithm uses the derivatives of the objective functions. Robust approximations to the derivatives are discussed. During optimization using the EBW algorithm the sum-to-one constraint of the probability distributions of the HMM is maintained. Hence, the parameters of the HMM are still 'normalized' probability distributions. This has the advantage that summing over missing variables is still possible as we show for Bayesian network classifiers in Pernkopf *et al.* (2012). This is in contrast to Kim & Pavlovic (2011) where also single-label classification is considered. Furthermore, they approximate the objective to obtain a convex optimization problem solved in a similar way as in Sha & Saul (2007b).

Generative pre-training is important in many discriminatively learned models Erhan *et al.* (2010); Bahl *et al.* (1986); Pernkopf *et al.* (2012). This can be seen as a form of regularization of the discriminative learning objective. Therefore, we initialize discriminative HMM training with the MLE solution. Furthermore, we use early stopping during discriminative optimization. Hence, the *discriminative* parameters are *partially* reflecting the MLE solution. Experimental results for frequently used time-series data such as handwritten characters, Australian sign language, digits of speech data and UCR time-series are provided. In all cases, maximum margin HMMs lead to competitive performance compared to CLL-HMMs and generative HMMs. Maximum margin HMMs mostly outperform the CLL-HMMs and the generatively optimized MLE-HMMs in terms of classification rate in all experiments.

The paper is organized as follows: In Section 2, we shortly review the Bayesian classifier for sequential data and introduce the HMM and the notation. In Section 3, conventional parameter estimation techniques for HMMs such as MLE and CLL optimization are summarized. Maximum margin parameter estimation for HMMs is derived in Section 4 using the EBW algorithm. This section also includes approximations of the derivatives necessary for the EBW method. In Section 5, we present experimental results. Section 6 concludes the paper.

2. Bayesian Classifier for Sequential Data

The task of classification is to assign a given observation sequence $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ to a class $c \in \{1, \dots, C\}$, where C is the number of classes, $\mathbf{x}_t \in \mathbb{R}^{\mathcal{D}}$, $\mathcal{D}$ is the number of observations at t and T is the length of the sequence. According to Bayes' rule, the class posterior $p(c|\mathbf{x})$ is given by

$$p(c|\mathbf{x}) = \frac{p(\mathbf{x}|c)p(c)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|c)p(c)}{\sum_{c'=1}^{C} p(\mathbf{x}|c')p(c')}, \tag{1}$$

where the likelihood term is assumed to be a parametric model for class c, i.e. $p(\mathbf{x}|c) = p(\mathbf{x}|\boldsymbol{\Theta}_c)$. In particular, we use an HMM $\boldsymbol{\Theta}_c$ for each class. The class label can be determined by the maximum a-posteriori (MAP) estimate, i.e. the most likely class label c^* is determined using the class posteriors as

$$c^* = \arg\max_{1 \le c \le C} p(c|\mathbf{x}) = \arg\max_{1 \le c \le C} [p(\mathbf{x}|\boldsymbol{\Theta}_c)\rho_c], \tag{2}$$

[1] The MMI criterion is closely related to the CLL objective.

where the denominator of (1) can be neglected since it only scales $p(c|\mathbf{x})$. The term $\rho_c = p(c)$ is the class prior distribution. The probability $p(\mathbf{x}|\mathbf{\Theta}_c)$ can be efficiently determined by using either the forward or the backward procedure Rabiner (1989). If the most probable state sequence $Q^* = \{q_1^*, \ldots, q_T^*\}$ producing $\mathbf{x}$ is known or estimated by the Viterbi algorithm, $p(\mathbf{x}|\mathbf{\Theta}_c)$ can also be approximated by $p^*(\mathbf{x}|\mathbf{\Theta})$, i.e. the product of the prior, the observation, and transition probabilities along the most probable path Q^* of HMM $\mathbf{\Theta}_c$.

An HMM can be fully described by two stochastic processes. The first is a Markov-process that produces a sequence of not directly observable states $Q = \{q_1, \ldots, q_t, \ldots, q_T\}$ where $q_t \in \{1, 2, \ldots, S\}$ and S is the number of states. The second process produces an observation $\mathbf{x}_t$ at every time step t of the state sequence according to a state-dependent observation probability distribution $b_i(\mathbf{x}_t) = p(\mathbf{x}_t|q_t = i)$. In a first-order HMM, the state of variable q_t depends on the state of the previous variable q_{t-1}, i.e. the transition from state $q_{t-1} = i$ to state $q_t = j$ occurs with a certain probability denoted by $a_{i,j} = p(q_t = j|q_{t-1} = i)$ where $\sum_{j=1}^{S} a_{i,j} = 1, \forall i \in \{1, \ldots, S\}$. The transition matrix A collects all transition probabilities $a_{i,j}$. The probability of being in hidden state i at the beginning of a sequence $t = 1$ is modeled by the state prior distribution $\pi_i = p(q_1 = i)$. We use a multivariate GMM to model the observation probabilities, i.e. we have a sum of M weighted Gaussians $\mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_{i,m}, \mathbf{\Sigma}_{i,m})$,

$$b_i(\mathbf{x}_t) = p(\mathbf{x}_t|q_t = i) = \sum_{m=1}^{M} \alpha_{i,m} \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_{i,m}, \mathbf{\Sigma}_{i,m}), \tag{3}$$

where $\alpha_{i,m}$ are the weights of each Gaussian component, $0 \leq a_{i,m} \leq 1$ and $\sum_{m=1}^{M} \alpha_{i,m} = 1$, and $\boldsymbol{\mu}_{i,m} \in \mathbb{R}^{\mathcal{D}}$ is the $\mathcal{D}$-dimensional mean vector and $\mathbf{\Sigma}_{i,m}$ is the $\mathcal{D} \times \mathcal{D}$ covariance matrix. We assume a diagonal covariance throughout the paper. An HMM is fully specified by the state prior distribution π_i, the transition matrix A and the emission probability $b_i(\mathbf{x}_t)$. These parameters are collected for HMM of class c in $\mathbf{\Theta}_c = \{\pi_{c,i}, a_{c,i,j}, \alpha_{c,i,m}, \boldsymbol{\mu}_{c,i,m}, \mathbf{\Sigma}_{c,i,m}\}_{i,j \in \{1,\ldots,S\}, m \in \{1,\ldots M\}}$.

3. Conventional Parameter Estimation of HMMs

Commonly, HMM parameters are determined iteratively using MLE Rabiner (1989); Pernkopf *et al.* (2014). Discriminative parameter optimization using conditional log-likelihood learning, i.e. the class posterior of the model is maximized, has been introduced in Bahl *et al.* (1986) as MMI training. Both objectives for parameter estimation are introduced in the following.

Formally, MLE parameters for the model of class c are learned as

$$\mathbf{\Theta}_c^{MLE} = \arg\max_{\mathbf{\Theta}_c} p(\mathcal{X}_c|\mathbf{\Theta}_c) = \prod_{n=1}^{N_c} p(\mathbf{x}^n|\mathbf{\Theta}_c), \tag{4}$$

where $\mathcal{X}_c = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{N_c}\}$ is a set of N_c training sequences belonging to class c. MLE of the HMM parameters leads to an iterative scheme such as the expectation-maximization (EM) algorithm also known as Baum-Welch algorithm for HMMs Rabiner (1989).

In contrast to generative methods, discriminative training of an HMM $\mathbf{\Theta}_c$ of class c involves all training samples $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2 \ldots, \mathcal{X}_C\}$, where $N = \sum_{c=1}^{C} N_c$. The conditional log-likelihood (CLL) is given as

$$CLL(\mathcal{X}|\mathbf{\Theta}) = \log \prod_{n=1}^{N} p(c^n|\mathbf{x}^n) = \log \prod_{n=1}^{N} \frac{p(\mathbf{x}^n|\mathbf{\Theta}_{c^n})\rho_{c^n}}{\sum_{c'=1}^{C} p(\mathbf{x}^n|\mathbf{\Theta}_{c'})\rho_{c'}} \tag{5}$$

and the parameters are determined according to

$$\mathbf{\Theta}_c^{CLL} = \arg\max_{\mathbf{\Theta}_c} CLL(\mathcal{X}|\mathbf{\Theta}), \tag{6}$$

where $\mathbf{\Theta} = \{\mathbf{\Theta}_1, \rho_1, \ldots, \mathbf{\Theta}_C, \rho_C\}$ and c^n denotes the class of sequence $\mathbf{x}^n$. Maximizing the CLL criterion is closely related to MMI estimation Bahl *et al.* (1986); Normandin & Morgera (1991); Normandin *et al.* (1994); Woodland & Povey (2002). The CLL can be maximized by gradient-based optimization methods, e.g. the EBW algorithm.

4. Maximum Margin Parameter Estimation

The multi-class margin Guo *et al.* (2005); Pernkopf *et al.* (2012) of sample n is

$$\tilde{d}_{\mathbf{\Theta}}^n = \min_{c \neq c^n} \frac{p(c^n|\mathbf{x}^n, \mathbf{\Theta})}{p(c|\mathbf{x}^n, \mathbf{\Theta})} = \min_{c \neq c^n} \frac{p(c^n, \mathbf{x}^n|\mathbf{\Theta})}{p(c, \mathbf{x}^n|\mathbf{\Theta})} = \frac{p(\mathbf{x}^n|\mathbf{\Theta}_{c^n})\rho_{c^n}}{\max_{c \neq c^n} p(\mathbf{x}^n|\mathbf{\Theta}_c)\rho_c}. \tag{7}$$

If $\tilde{d}_{\mathbf{\Theta}}^n > 1$, then sample n is correctly classified and vice versa. We replace the max operator by the differentiable approximation $\max_x f(x) \approx \left[\sum_x (f(x))^\eta\right]^{\frac{1}{\eta}}$, where $\eta \geq 1$ and $f(x)$ is non-negative. In the limit of $\eta \to \infty$ the approximation converges to the maximum operator. Replacing the maximum with its approximation, we obtain

$$d_{\mathbf{\Theta}}^n = \frac{p(\mathbf{x}^n|\mathbf{\Theta}_{c^n})\rho_{c^n}}{\left[\sum_{c \neq c^n} (p(\mathbf{x}^n|\mathbf{\Theta}_c)\rho_c)^\eta\right]^{\frac{1}{\eta}}}. \tag{8}$$

Usually, the maximum margin approach maximizes the margin of the sample with the smallest margin, i.e. $\min_{n=1,\ldots,N} d_{\mathbf{\Theta}}^n$ for a separable classification problem Schölkopf & Smola (2001). We aim to relax this by introducing a soft margin, i.e. we focus on samples with a $d_{\mathbf{\Theta}}^n$ close to one. Therefore, we consider the *hinge* loss function according to

$$\tilde{J}(\mathcal{X}|\mathbf{\Theta}) = \prod_{n=1}^{N} \min\left[\kappa, d_{\mathbf{\Theta}}^n\right], \tag{9}$$

where parameter $\kappa > 1$ controls the influence of the margin $d_{\mathbf{\Theta}}^n$ in the hinge loss $\tilde{J}(\mathcal{X}|\mathbf{\Theta})$ and is set by cross-validation. Maximizing this function with respect to the parameters $\mathbf{\Theta}$ implicitly means to increase the margin $d_{\mathbf{\Theta}}^n$ whereas the emphasis is on

samples with a margin $d^n_{\mathbf{\Theta}} < \kappa$, i.e. samples with a large positive margin have no impact on the optimization. Maximizing $\tilde{J}(\mathcal{X}|\mathbf{\Theta})$ via EBW or gradient descent is not straight forward due to the discontinuity in the derivative at $d^n_{\mathbf{\Theta}} = \kappa$. Therefore, we propose to use for the *hinge* function $h(y) = \min[\kappa, y]$ a *smooth hinge* function which enables a smooth transition of the derivative and has a similar shape as $h(y)$. We propose the following function inspired by the Huber loss Huber (1964). In particular, we approximate the discontinuity by a circle segment as

$$h(y) = \begin{cases} y + \frac{1}{2}, & \text{if } y \leq \kappa - 1 \\ \kappa - \frac{1}{2}(y-\kappa)^2, & \text{if } \kappa - 1 < y < \kappa \\ \kappa, & \text{if } y \geq \kappa \end{cases} \tag{10}$$

which requires to divide the data $\mathcal{X}$ into three partitions depending on $y = d^n_{\mathbf{\Theta}}$, i.e. $\mathcal{X}^1$ contains samples where $d^n_{\mathbf{\Theta}} \leq \kappa - 1$, $\mathcal{X}^2$ consists of samples with a margin in the range $\kappa - 1 < d^n_{\mathbf{\Theta}} < \kappa$, and $\mathcal{X}^3 = \mathcal{X} \setminus \left\{\mathcal{X}^1 \cup \mathcal{X}^2\right\}$. The smooth hinge function is illustrated in Pernkopf *et al.* (2012).

Basically, there are other smoothing techniques available for non-smooth convex objectives, e.g. Nesterov (2005). In our case, smoothing of the objective function makes it amenable for gradient-based optimization methods while still approximating the original objective well. Experiments using a similar *parametrized* smooth hinge function show only a slight influence on performance for maximum margin Bayesian network classifiers Pernkopf *et al.* (2012). Furthermore, a similar approximation of the maximum margin objective outperforms a convex formulation (which requires relaxation of constraints) with respect to computational requirements, while the classification performance is almost identical.

Using the smooth hinge function in (10), our objective function for margin maximization is

$$\begin{aligned} J(\mathcal{X}|\mathbf{\Theta}) &= \prod_{n=1}^{N} h(d^n_{\mathbf{\Theta}}) \qquad (11) \\ &= \left\{\prod_{n\in\mathcal{X}^1}\left(d^n_{\mathbf{\Theta}} + \frac{1}{2}\right)\right\}\left\{\prod_{n\in\mathcal{X}^2}\left[\kappa - \frac{1}{2}\left(d^n_{\mathbf{\Theta}} - \kappa\right)^2\right]\right\}\kappa^{|\mathcal{X}^3|}. \end{aligned}$$

4.1. Optimization of the Margin Objective

The EBW algorithm (more details are given in Appendix A) is an iterative procedure which can be used to optimize rational functions Gopalakrishnan *et al.* (1991). We use the EBW framework to optimize the margin objective in (11) for the discrete model parameters $\rho_c, \pi_{c,i}, a_{c,i,j}, \alpha_{c,i,m}$. The parameter re-estimation equation of the form

$$\theta^j_i \leftarrow \frac{\theta^j_i\left(\frac{\partial \log J(\mathcal{X}|\mathbf{\Theta})}{\partial \theta^j_i} + D\right)}{\sum_l \theta^j_l\left(\frac{\partial \log J(\mathcal{X}|\mathbf{\Theta})}{\partial \theta^j_l} + D\right)}, \tag{12}$$

is used, where $\theta^j_i \geq 0$, $\sum_i \theta^j_i = 1$, and j indicates a particular discrete variable. EBW requires the partial derivative $\frac{\partial \log J(\mathcal{X}|\mathbf{\Theta})}{\partial \mathbf{\Theta}}$ and D. Both terms are provided in the sequel. Specifically, the derivative $\frac{\partial \log J(\mathcal{X}|\mathbf{\Theta})}{\partial \mathbf{\Theta}}$ for the re-estimation equation (12) of the EBW algorithm is

$$\frac{\partial \log J(\mathcal{X}|\mathbf{\Theta})}{\partial \mathbf{\Theta}} = \sum_{n=1}^{N} s^n \frac{\partial \log d^n_{\mathbf{\Theta}}}{\partial \mathbf{\Theta}} \tag{13}$$

where s^n denotes a sample dependent weight given as follows:

$$s^n = \begin{cases} \frac{d^n_{\mathbf{\Theta}}}{d^n_{\mathbf{\Theta}} + \frac{1}{2}}, & \text{if } n \in \mathcal{X}^1 \\ \frac{\kappa d^n_{\mathbf{\Theta}} - \left(d^n_{\mathbf{\Theta}}\right)^2}{\kappa - \frac{1}{2}(d^n_{\mathbf{\Theta}} - \kappa)^2}, & \text{if } n \in \mathcal{X}^2 \\ 0, & \text{if } n \in \mathcal{X}^3 \end{cases}. \tag{14}$$

Approximating $p(\mathbf{x}|\mathbf{\Theta}_c)$ with the probability of the most probable state sequence of the Viterbi algorithm, i.e.

$$p(\mathbf{x}|\mathbf{\Theta}_c) \approx p^*(\mathbf{x}|\mathbf{\Theta}_c) = \pi_{c,q^*_1} b_{c,q^*_1}(\mathbf{x}_1) \prod_{t=2}^{T} a_{c,q^*_{t-1},q^*_t} b_{c,q^*_t}(\mathbf{x}_t), \tag{15}$$

the log of the margin $d^n_{\mathbf{\Theta}}$ of sample $\mathbf{x}^n$ in Eq. (8) decomposes to

$$\begin{aligned} \log d^n_{\mathbf{\Theta}} &= \log(p(\mathbf{x}^n|\mathbf{\Theta}_{c^n})\rho_{c^n}) - \frac{1}{\eta}\log \sum_{c'\neq c^n}(p(c', \mathbf{x}^n|\mathbf{\Theta}_{c'})\rho_{c'})^\eta \\ &= \log\pi_{c^n, i^{*,n}_{c^n,1}} + \sum_{t=1}^{T^n}\log b_{c, i^{*,n}_{c^n,t}}(\mathbf{x}^n_t) + \sum_{t=2}^{T^n}\log a_{c, i^{*,n}_{c^n,t-1}, i^{*,n}_{c^n,t}} + \log\rho_{c^n} \\ &\quad - \frac{1}{\eta}\log\Big[\sum_{c'\neq c^n}^{C}\Big(\pi_{c', i^{*,n}_{c',1}}\prod_{t=1}^{T^n} b_{c', i^{*,n}_{c',t}}(\mathbf{x}^n_t)\prod_{t=2}^{T^n} a_{c', i^{*,n}_{c',t-1}, i^{*,n}_{c',t}}\rho_{c'}\Big)^\eta\Big], \end{aligned} \tag{16}$$

where $i^{*,n}_{c^n,t}$ is the most probable state of the HMM of class c^n for a sequence $\mathbf{x}^n$ at time t.

The derivative for ρ_c of $\frac{\partial \log d^n_{\mathbf{\Theta}}}{\partial \mathbf{\Theta}}$ in (13) is

$$\begin{aligned} \frac{\partial \log d^n_{\mathbf{\Theta}}}{\partial \rho_c} &= \frac{\mathbb{1}_{\{c=c^n\}}}{\rho_c} - \frac{\mathbb{1}_{\{c\neq c^n\}}(p(\mathbf{x}^n|\mathbf{\Theta}_c)\rho_c)^{\eta-1}p(\mathbf{x}^n|\mathbf{\Theta}_c)}{\sum_{c'\neq c^n}(p(\mathbf{x}^n|\mathbf{\Theta}_{c'})\rho_{c'})^\eta}\frac{\rho_c}{\rho_c} \\ &= \frac{1}{\rho_c}\left[\mathbb{1}_{\{c=c^n\}} - \mathbb{1}_{\{c\neq c^n\}}\frac{(p(\mathbf{x}^n|\mathbf{\Theta}_c)\rho_c)^\eta}{\sum_{c'\neq c^n}(p(\mathbf{x}^n|\mathbf{\Theta}_{c'})\rho_{c'})^\eta}\right] \\ &= \frac{1}{\rho_c}\left[z^n_c - \breve{z}^n_c \cdot r^{n,\eta}_c\right], \end{aligned} \tag{17}$$

where

$$r^{n,\eta}_c = \frac{(p(\mathbf{x}^n|\mathbf{\Theta}_c)\rho_c)^\eta}{\sum_{c'\neq c^n}(p(\mathbf{x}^n|\mathbf{\Theta}_{c'})\rho_{c'})^\eta}, \tag{18}$$

$$z^n_c = \mathbb{1}_{\{c=c^n\}} \quad \text{and} \tag{19}$$

$$\breve{z}^n_c = \mathbb{1}_{\{c\neq c^n\}}. \tag{20}$$

Symbol $\mathbb{1}_{\{i=j\}}$ denotes the indicator function (i.e. equals 1 if the Boolean expression $i = j$ is true and 0 otherwise).

Furthermore, the partial derivatives of $\log d^n_{\mathbf{\Theta}}$ with respect to π_i, $a_{c,i,j}$ and $\alpha_{c,i,m}$ are given as follows:

$$\frac{\partial \log d^n_{\mathbf{\Theta}}}{\partial \pi_{c,i}} = \frac{1}{\pi_{c,i}}\left[u^n_{c,i,1} - \breve{u}^n_{c,i,1}\cdot r^{n,\eta}_c\right] \tag{21}$$

$$\frac{\partial \log d^n_{\mathbf{\Theta}}}{\partial a_{c,i,j}} = \frac{1}{a_{c,i,j}}\left[y^n_{c,i,j} - \breve{y}^n_{c,i,j}\cdot r^{n,\eta}_c\right] \tag{22}$$

$$\frac{\partial \log d^n_{\mathbf{\Theta}}}{\partial \alpha_{c,i,m}} = \frac{1}{\alpha_{c,i,m}}\sum_{t=1}^{T^n}\left[\gamma^n_{c,i,m,t}\left(u^n_{c,i,t} - \breve{u}^n_{c,i,t}\cdot r^{n,\eta}_c\right)\right], \tag{23}$$

where

$$u^n_{c,i,t} = \mathbb{1}_{\{c=c^n, i=i^{*,n}_{c,t}\}} \quad (24)$$

$$\check{u}^n_{c,i,t} = \mathbb{1}_{\{c\neq c^n, i=i^{*,n}_{c,t}\}} \quad (25)$$

$$y^n_{c,i,j} = \sum_{t=2}^{T^n} \mathbb{1}_{\{c=c^n, i=i^{*,n}_{c,t-1}, j=i^{*,n}_{c,t}\}} \quad (26)$$

$$\check{y}^n_{c,i,j} = \sum_{t=2}^{T^n} \mathbb{1}_{\{c\neq c^n, i=i^{*,n}_{c,t-1}, j=i^{*,n}_{c,t}\}} \quad (27)$$

and

$$\gamma^n_{c,i,m,t} = \frac{\alpha_{c,i,m} \cdot \mathcal{N}(\mathbf{x}^n_t|\boldsymbol{\mu}_{c,i,m}, \boldsymbol{\Sigma}_{c,i,m})}{\sum_{m'=1}^{M} \alpha_{c,i,m'} \cdot \mathcal{N}(\mathbf{x}^n_t|\boldsymbol{\mu}_{c,i,m'}, \boldsymbol{\Sigma}_{c,i,m'})}. \quad (28)$$

4.2. Approximation of the Gradient

The derivatives (17), (21), (22) and (23) are sensitive to small parameter values. Merialdo Merialdo (1988) observed that low-valued parameters ρ_c, $\pi_{c,i}$, $a_{c,i,m}$ and $\alpha_{c,i,m}$ may cause a large magnitude of the gradient and the optimization concentrates on those parameters. However, small parameter values indicate that they are rarely used during the production of an observation sequence. Hence, there is not sufficiently training data available for reliably estimating very low probabilities and concentrating on low-valued parameters is unreliable. Therefore, he suggests to focus on modifying better estimated high-valued parameters during optimization by using an approximation of the gradients. In particular, for gradients of the form $\frac{\partial \log d^n_{\boldsymbol{\Theta}}}{\partial \theta^j_i} = \frac{1}{\theta^j_i}(c_{i,j} - c'_{i,j})$, as in our case, he suggests to concentrate on high-valued parameters by replacing the gradient by

$$\frac{\partial \log d^n_{\boldsymbol{\Theta}}}{\partial \theta^j_i} \approx \frac{c_{i,j}}{\sum_j c_{i,j}} - \frac{c'_{i,j}}{\sum_j c'_{i,j}}. \quad (29)$$

This approximation of the gradients has been used for CLL learning in Normandin & Morgera (1991); Normandin *et al.* (1994). Unfortunately, approximating the gradient by (29) cannot be applied to the derivatives of the margin, because the approximated gradient disappears for any HMM parameter. Therefore, we suggest an alternative approximation in order to obtain reliable parameter updates. Since the unreliability of the updates is caused by small parameter values due to high values of the gradients Merialdo (1988), normalizing the gradient by a sum-to-one constraint of the absolute gradient values keeps the updates reliable. For gradients of the form $\frac{\partial \log d^n_{\boldsymbol{\Theta}}}{\partial \theta^j_i} = \frac{1}{\theta^j_i}(c_{i,j} - c'_{i,j})$, we propose to approximate the gradient by

$$\frac{\partial \log d^n_{\boldsymbol{\Theta}}}{\partial \theta^j_i} \approx \frac{\frac{1}{\theta^j_i}(c_{i,j} - c'_{i,j})}{\sum_{i'=1}^{S} \left|\frac{1}{\theta^j_{i'}}(c_{i',j} - c'_{i',j})\right|}. \quad (30)$$

The resulting approximations of the derivatives in (17), (21), (22) and (23) are provided in the algorithm for maximum margin (MM) training of HMMs in Appendix B. As an alternative, Woodland and Povey Woodland & Povey (2002) proposed an alternative mixture weight update rule using an iterative procedure.

4.3. Approximation for the Gaussians

EBW has been formulated for discrete probability distributions. Normandin and Morgera Normandin & Morgera (1991) introduced a discrete approximation of the Gaussian distribution assuming diagonal covariance matrices. This leads to the re-estimation equation for $\boldsymbol{\mu}_{c,i,m}$ and $\boldsymbol{\Sigma}_{c,i,m}$ given as

$$\tilde{\mu}_{c,i,m} \leftarrow \frac{\sum_{n=1}^{N} s^n \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r^{n,\eta}_c)\mathbf{x}^n_t\right] + D\mu_{c,i,m}}{\sum_{n=1}^{N} s^n \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r^{n,\eta}_c)\right] + D} \quad (31)$$

and

$$\tilde{\boldsymbol{\Sigma}}_{c,i,m} \leftarrow \frac{g_{c,i,m} + D(\boldsymbol{\Sigma}_{c,i,m} + (\boldsymbol{\mu}_{c,i,m})^2)}{\sum_{n=1}^{N} s^n \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r^{n,\eta}_c)\right] + D} - (\tilde{\boldsymbol{\mu}}_{c,i,m})^2, \quad (32)$$

where $g_{c,i,m} = \sum_{n=1}^{N} s^n \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r^{n,\eta}_c)(\mathbf{x}^n_t)^2\right]$ and the squares of $\mathbf{x}^n_t$ and $\boldsymbol{\mu}_{c,i,m}$ are taken element-wise.

4.4. Implementation of the MM-HMM EBW Algorithm

The EBW algorithm converges to a local optimum of $J(\mathcal{X}|\boldsymbol{\Theta})$ providing a sufficiently large value for D. Setting the constant D is not trivial. If it is chosen too large then training is slow and if it is too small the update may fail to increase the objective function. In practical implementations heuristics have been suggested Woodland & Povey (2002); Klautau *et al.* (2003); Pernkopf & Wohlmayr (2010). In order to obtain positive covariances, the inequality

$$\frac{g_{c,i,m,d} + D(\sigma_{c,i,m,d} + \mu^2_{c,i,m,d})}{h_{c,i,m} + D} - \left(\frac{k_{c,i,m,d} + D\mu_{c,i,m,d}}{h_{c,i,m} + D}\right)^2 > 0 \quad (33)$$

must hold for any covariance $\sigma_{c,i,m,d}$ of dimension $d \in \mathcal{D}$, where

$$h_{c,i,m} = \sum_{n=1}^{N} s^n \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r^{n,\eta}_c)\right] \quad (34)$$

and

$$k_{c,i,m} = \sum_{n=1}^{N} s^n \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}(u^n_{c,i,t} - \check{u}^n_{c,i,t} \cdot r^{n,\eta}_c)\mathbf{x}^n_t\right]. \quad (35)$$

Rearranging (33) leads to a quadratic inequality with respect to D Valtchev *et al.* (1997):

$$\underbrace{\sigma_{c,i,m,d}}_{a} D^2 + \underbrace{(\sigma_{c,i,m,d}h + \mu^2_{c,i,m,d} + g_{c,i,m,d} - 2k_{c,i,m,d}\mu_{c,i,m,d})}_{b} D + \underbrace{g_{c,i,m,d}h - k^2_{c,i,m,d}}_{c} > 0 \quad (36)$$

We propose to set

$$D = F \cdot \max\{D_1, D_2, D_3\}, \quad (37)$$

Table 1. Classification rates of MLE-HMMs, CLL-HMMs, and MM-HMMs in [%] on data of the Pendigits characters.

	MLE-HMM			CLL-HMM			MM-HMM		
S	M			M			M		
	2	**3**	**4**	**2**	**3**	**4**	**2**	**3**	**4**
2	90.3 ± 0.98	94.3 ± 0.77	93.0 ± 0.84	96.4 ± 0.62	96.9 ± 0.57	97.7 ± 0.50	97.0 ± 0.57	97.4 ± 0.53	**97.9 ± 0.47**
3	92.5 ± 0.87	92.5 ± 0.87	93.9 ± 0.80	97.1 ± 0.56	96.7 ± 0.59	97.4 ± 0.52	97.8 ± 0.49	97.9 ± 0.48	**98.4 ± 0.42**
4	92.5 ± 0.87	93.9 ± 0.79	95.6 ± 0.68	97.5 ± 0.52	96.8 ± 0.58	98.2 ± 0.44	97.4 ± 0.53	97.9 ± 0.47	**98.6 ± 0.39**
5	93.7 ± 0.80	94.9 ± 0.73	94.2 ± 0.77	96.4 ± 0.62	97.1 ± 0.56	97.31 ± 0.53	98.2 ± 0.44	98.3 ± 0.43	**98.8 ± 0.36**
6	94.7 ± 0.74	94.3 ± 0.77	95.7 ± 0.67	97.9 ± 0.48	97.6 ± 0.50	98.1 ± 0.46	98.3 ± 0.43	97.9 ± 0.47	**98.5 ± 0.40**

Table 2. Classification rates of MLE-HMMs, CLL-HMMs, and MM-HMMs in [%] on data of the Auslan database.

	MLE-HMM			CLL-HMM			MM-HMM		
S	M			M			M		
	2	**3**	**4**	**2**	**3**	**4**	**2**	**3**	**4**
2	68.1 ± 7.72	72.9 ± 7.37	73.6 ± 7.30	68.3 ± 7.71	**87.1 ± 5.54**	80.0 ± 6.63	81.2 ± 6.47	85.5 ± 5.84	84.1 ± 6.07
3	71.2 ± 7.50	75.0 ± 7.17	77.9 ± 6.88	86.7 ± 5.63	86.4 ± 5.67	80.2 ± 6.60	84.8 ± 5.95	86.0 ± 5.76	**87.9 ± 5.41**
4	73.3 ± 7.33	77.4 ± 6.93	81.2 ± 6.47	86.2 ± 5.71	78.6 ± 6.80	80.7 ± 6.54	86.7 ± 5.63	78.8 ± 6.77	**87.4 ± 5.50**
5	76.2 ± 7.06	78.1 ± 6.85	80.5 ± 6.57	79.0 ± 6.74	79.5 ± 6.68	**80.7 ± 6.54**	80.5 ± 6.57	82.6 ± 6.28	**80.7 ± 6.54**

where

$$D_{1,2} = \frac{-(b) \pm \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \tag{38}$$

$$D_3 = 1 + \left| \min_{i,j} \frac{\partial \log J(\mathcal{X}|\boldsymbol{\Theta})}{\partial \theta_i^j} \right|. \tag{39}$$

D_3 guarantees a positive parameter after the update in (12) and $F > 1$ regulates the convergence speed of the algorithm.

The parameters $\boldsymbol{\Theta}$ for discriminative learning are initialized to the MLE of the HMM determined by the EM algorithm (see Section 3). Generative model pre-training can be seen as a form of regularization Erhan *et al.* (2010). The class prior is set to the normalized class frequency in $\mathcal{X}$, i.e. $\rho_c = \frac{N_c}{N}$. A detailed algorithm of maximum margin (MM) training for HMM is provided in Appendix B.

5. Experiments

The maximum margin HMM is compared to the MLE and CLL optimized HMM. We provide results for spoken digit classification using the TIMIT corpus, handwritten character data Australian sign language data, and UCR time-series data. We use the acronym *MLE-HMM* for generatively learned HMMs and *CLL-HMM* and *MM-HMM* for discriminative CLL and maximum margin HMM parameter estimation, respectively. For comparison, we used a 1-nearest neighbor classifier with a similarity measure obtained by dynamic time warping (DTW) Hiroaki & Chiba (1978); Berndt & Clifford (1994).

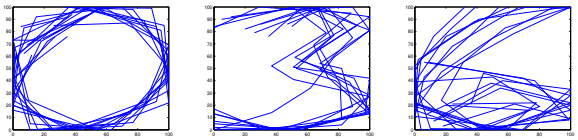
5.1. Experimental Setup

The HMM parameters trained by MLE have been used as initialization for the discriminative methods, i.e. CLL and MM parameter learning. We perform classification using HMMs with varying numbers of mixture components and states. We use up to $S = 6$ states and $M \in \{2, 3, 4\}$ mixture components. A large value of S and M leads to an HMM with many parameters. For a reliable estimation of many parameters a sufficiently large data set has to be provided. Discriminative training methods were unstable for too low values of the convergence-regulating constant F. The minimum value of F for convergence has been determined empirically for each data set. For MM-HMM, the margin scaling parameter $\kappa \in \{0.001, \ldots, 1\}$[2] is set by 3-fold cross-validation on the training set. Furthermore, the parameter η for the approximation of the maximum in Eq. (8) has limited influence on the performance, i.e. it is set to 2. The objective function does not necessarily increase at each iteration. Reasons are the approximation of the derivatives and a bad choice of D. We used early stopping for training the CLL-HMM and MM-HMM models, i.e. the best number of training iterations is determined on the training set. Numerical underflow might occur during the estimation of very small HMM parameters, leading to unreliable training results. To overcome this, we set the parameter values to a minimum value. In particular, the values of the covariance matrix and the transition probabilities are set to 0.0001 and 0.001 during optimization, respectively. For DTW, the warping window parameter $w \in \{1, \ldots, 10\}$ specifies the size of the local neighborhood in DTW, i.e. consider comparing two sequences $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_i \ldots, \mathbf{x}_{T_x}\}$ and $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_j \ldots, \mathbf{y}_{T_y}\}$, the distance between $\mathbf{x}_i$ and $\mathbf{y}_j$ is calculated only for indexes i and j such that $|i - j| \leq w$ Hiroaki & Chiba (1978); Berndt & Clifford (1994). w is selected by cross-validation and the $\ell 2$-norm is used as distance measure.

5.2. Handwritten Digit Classification on the Pendigits Database

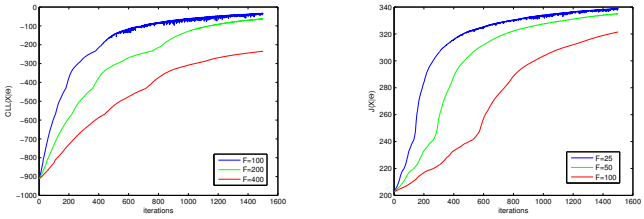
The Pendigits database contains trajectories of handwritten digits from 0 to 9 from 44 different writers. Some examples

[2] The selected κ values for the experiments in the following sections using Auslan, Pendigits, TIMIT, ECG200, OSULeaf, and SwedishLeaf are 0.26, 0.0215, 0.0023, 0.209, 0.209, and 0.209, respectively.



(a) digit '0' (b) digit '3' (c) digit '6'

Fig. 1. Digits from the Pendigits database.



(a) CLL-HMM (b) MM-HMM

Fig. 2. Convergence of CLL-HMM and MM-HMM on the SwedishLeaf dataset of the UCR database.

for digit '0', '3', and '6' are shown in Figure 1. The data is divided into a training set of 7494 samples from 30 writers and a test set of 3498 samples from 14 writers, respectively. The sample sequences have a uniform length of 8 time steps. The feature vectors consist of two elements: the absolute pen position in x- and y-direction. Values have been normalized to a range between 0 and 100. We rescaled the values to a range between -1 and 1 and added the first derivatives in x- and y-direction. The derivatives were obtained by numerical differentiation of the coordinates. The classification performances for MLE-HMM, CLL-HMM and MM-HMM of the Pendigits data are shown in Table 1. Best results for each number of states $S \in \{2, 3, \ldots, 6\}$ are bold.

In this task discriminative MM training clearly outperforms generative MLE-HMMs. MM-HMMs perform mostly slightly better than CLL-HMMs.

5.3. Australian Sign Language Classification

The Australian Sign Language (Auslan) data set consists of 6647 samples of 95 different signs from 5 writers. The feature vectors contain 15 values, including the position in x-, y- and z- direction, finger bend and more. We ignored the attributes 5, 6 and 11-14 as advised in the description of the data set. For the experiments, we selected 10 signs that were used in Kim & Pavlovic (2011) and applied a median filter to the data. Furthermore, we compressed the sequences to a fixed length of 10 time steps by taking the means of equally-sized partitions of each sequence. We split the data randomly into 80% of the samples for training and 20% for testing, respectively. The splitting was repeated three times and the average is reported. The classification performances for MLE-HMM, CLL-HMM and MM-HMM of the Auslan data are shown in Table 2. Each result is the mean of three runs with randomly selected partitions for training an testing. Best results for each number of states $S \in \{2, 3, \ldots, 5$ *rbrace* are bold. Discriminative MM training outperforms generative MLE-HMMs. Again, MM-HMMs perform mostly slightly better than CLL-HMMs.

5.4. Sequence Classification on the UCR Database

In this experiment, we evaluated HMM classification on three data sets, namely ECG200, OSULeaf, and SwedishLeaf, of the UCR database. The data sets vary in their number of classes, size of training and test set and sequence length. All provided data sets have one single attribute. Further information about specific data sets can be found at http://www.cs.ucr.edu/~eamonn/ time_series_data/. For the experiments, we selected three data sets with a sufficiently large number of training and test samples. We appended the first derivatives of the sequences to the feature vectors. Furthermore, we compressed the time series to approximately $1/10$ of their original length. Only results for the optimal number of states and components are reported.

Table 3. Classification rates of MLE-HMMs, CLL-HMMs and MM-HMMs in [%] on data of the UCR database.

Dataset	MLE-HMM	CLL-HMM	MM-HMM
ECG200	84.0 ± 7.19	86.0 ± 6.80	**89.0 ± 6.13**
OSULeaf	62.4 ± 6.10	63.2 ± 6.08	**65.7 ± 5.98**
SwedishLeaf	58.2 ± 3.87	74.9 ± 3.40	**77.1 ± 3.29**

The classification performances for MLE-HMM, CLL-HMM, MM-HMM and DTW of the UCR data are shown in Table 3. Best results for each dataset are bold. MM-HMMs outperform MLE-HMMs and CLL-HMMs. Fig. 2 shows the convergence of the objective function of CLL-HMMs and MM-HMMs for different values of F on the SwedishLeaf dataset of the UCR database.

5.5. Spoken Digit Classificaton

A set of spoken numbers from 'one' to 'ten' has been extracted from the TIMIT corpus Lamel *et al.* (1986). The utterances are recorded at a sampling rate of 16 kHz. For each digit a sequence of observation vectors $\mathbf{x}_t$ consisting of 13 mel-frequency cepstral coefficients (MFCCs) and their first and second derivatives Jurafsky & Martin (2009) are determined at a frame rate of 10ms and a window length of 25 ms. Thus, each observation $\mathbf{x}_t$ consists of 39 features. Additionally, principal component analysis (PCA) was applied to whiten the data. In total, 165 training and 64 test sequences of spoken digits are available. The classification rates are shown in Tables 4 for MLE, CLL and MM training of the HMM using $M = 2$ Gaussian components. Best results for each number of states S are bold.

In this task, MM training achieves the highest classification rate, i.e. generative MLE-HMMs are outperformed in each case while CLL-HMMs provides the same classification performance for $S = 5$ and $S = 2$. The classification performance of MLE, CLL and MM decreases with an increasing number of HMM states. Due to the little amount of training samples, these methods are presumably suffering from overfitting, i.e. the number of parameters is too large to be reliably estimated.

Table 4. Classification rates of MLE-HMMs, CLL-HMMs, and MM-HMMs in [%] on spoken digit classification.

S	$M = 2$		
	MLE-HMM	CLL-HMM	MM-HMM
2	90.6 ± 7.14	**95.3 ± 5.18**	**95.3 ± 5.18**
3	93.8 ± 5.93	90.6 ± 7.14	**96.9 ± 4.26**
4	82.8 ± 9.24	93.8 ± 5.93	**98.4 ± 3.04**
5	79.7 ± 9.86	**95.3 ± 5.18**	**95.3 ± 5.18**
6	82.8 ± 9.24	90.6 ± 7.14	**96.9 ± 4.26**

Table 5. Classification rates of DTW in [%].

Data	Warping window size w	Classification rate
Pendigits	0	97.6 ± 0.51
Auslan	1	88.8 ± 5.22
TIMIT	3	79.7 ± 9.86
ECG200	5	87.0 ± 6.59
OSULeaf	6	57.9 ± 6.22
SwedishLeaf	8	89.0 ± 2.46

5.6. Comparison to DTW and other Methods

Table 5 shows classification performance of DTW on the data sets stated above. For most data sets, these classifiers are outperformed by MM-HMM classification. For comparison reasons, state-of-the-art results from the literature on the data sets and the corresponding classification methods are summarized in Table 6. The methods are:

- Decision Templates Ebrahimpour & Hamedi (2009): Decision templates are extracted from multiple classifiers and an aggregation rule is applied. The decision profile $DP(\mathbf{x})$ for a sample $\mathbf{x}$ is a $L \times C$ matrix containing the output of L classifiers, e. g. multilayer perceptrons, for C classes. The decision Template DT_c for a class $c \in 1, \ldots, C$ is the average over all DPs of training samples labeled with c. A new sample $\mathbf{z}$ is classified by selecting the class with the best similarity between $DP(\mathbf{z})$ and DT_c.

- Large margin-HMM (LM-HMM) Kim & Pavlovic (2011): They approximate the large margin objective to obtain a convex optimization problem solved in a similar way as in Sha & Saul (2007b).

- Hub-based Selection Buza *et al.* (2011): A 1-nearest neighbor DTW classification technique is used on top of an instance selection technique keeping only the most informative samples. Samples are selected by using the property of hubness Radovanović *et al.* (2009).

- Triangle side length and angle representation (TSLA) Mouine *et al.* (2013): This approach has been tailored to leaf classification, i.e. it is a multiscale triangle area, side length and angle representation of the leaf. Essentially, this means that other features are used. Classification is based on a locality sensitive hashing method.

In all cases except SwedishLeaf our MM-HMM is providing the best recognition rates. For the SwedishLeaf classification problem MM-HMMs do not deliver good results. One reason is that the benchmark method in Buza *et al.* (2011) rely on more appropriate features.

Table 6. Classification rates in [%] for other state-of-the-methods from literature.

Data	Reference Method	Classification rate
Pendigits	Decision Templates Ebrahimpour & Hamedi (2009)	97.6
Auslan	LM-HMM Kim & Pavlovic (2011)	71.2
ECG200	Hub-based Selection Buza *et al.* (2011)	83.5
OSULeaf	LM-HMM Kim & Pavlovic (2011)	61.7
SwedishLeaf	TSLA Mouine *et al.* (2013)	96.5

6. Conclusion

A discriminative maximum margin learning method is derived for hidden Markov models and compared to conditional log-likelihood and maximum likelihood parameter estimation. We formulate the margin of a sample sequence as the ratio of the class posterior of the true class and the most competing class. This sample margin is embedded in a hinge loss function. The derivatives of a smooth approximation of this objective function are used in the extended Baum-Welch algorithm to discriminatively optimize the HMM parameters. In the experiments, we provide results for the tasks of classifying handwritten characters, Australian sign language data, digits of speech data and UCR time-series data. Discriminatively trained HMMs outperform the generative maximum likelihood approach. Maximum margin training outperforms conditional likelihood training in almost all cases. generative learning.

Appendix A: EBW Algorithm

In its original form Baum & Eagon (1967), the Baum-Eagon inequality has been formulated for domains of discrete probabilities. Consider a domain E of discrete probability values $\mathbf{\Phi} = \{\varphi_i^j\}$, with $\varphi_i^j \geq 0$, $\sum_i \varphi_i^j = 1$, and $j = 1, ..., J$. Given a homogeneous polynomial $Q(\mathbf{\Phi})$ with nonnegative coefficients over the domain E, the Baum-Eagon inequality offers an iterative method to find local maxima in Q. It provides a transformation, $T : E \rightarrow E$, such that $Q(T(\mathbf{\Phi})) > Q(\mathbf{\Phi})$, unless $T(\mathbf{\Phi}) = \mathbf{\Phi}$. This transformation, called *growth transform*, maps from $\hat{\mathbf{\Phi}} \in E$ to $T(\hat{\mathbf{\Phi}}) = \bar{\mathbf{\Phi}} \in E$, where

$$\bar{\varphi}_i^j = \frac{\hat{\varphi}_i^j \frac{\partial Q(\hat{\mathbf{\Phi}})}{\partial \varphi_i^j}}{\sum_{i'} \hat{\varphi}_{i'}^j \frac{\partial Q(\hat{\mathbf{\Phi}})}{\partial \varphi_{i'}^j}}. \tag{40}$$

For brevity, $\frac{\partial Q(\hat{\mathbf{\Phi}})}{\partial \varphi_i^j}$ denotes the partial derivative $\frac{\partial Q}{\partial \varphi_i^j}$ evaluated at point $\hat{\mathbf{\Phi}}$.

In Gopalakrishnan *et al.* (1991), the growth transform is extended[3] to rational functions $R(\mathbf{\Phi})$ over E:

$$R(\mathbf{\Phi}) = \frac{\text{Num}(\mathbf{\Phi})}{\text{Den}(\mathbf{\Phi})}.$$

[3] Additionally, they show that the growth transform in Eq. (40) can be applied to nonhomogeneous polynomials.

This is done by converting $R(\mathbf{\Phi})$ into a polynomial $Q_{\hat{\mathbf{\Phi}}}(\mathbf{\Phi})$ for a given $\hat{\mathbf{\Phi}}$ such that if $Q_{\hat{\mathbf{\Phi}}}\left(T\left(\hat{\mathbf{\Phi}}\right)\right) > Q_{\hat{\mathbf{\Phi}}}(\hat{\mathbf{\Phi}})$, then $R\left(T\left(\hat{\mathbf{\Phi}}\right)\right) > R\left(\hat{\mathbf{\Phi}}\right)$, except $T\left(\hat{\mathbf{\Phi}}\right) = \hat{\mathbf{\Phi}}$. The polynomial $Q_{\hat{\mathbf{\Phi}}}(\mathbf{\Phi})$ that fulfills this condition is given in Gopalakrishnan *et al.* (1991) as

$$Q_{\hat{\mathbf{\Phi}}}(\mathbf{\Phi}) = \mathrm{Num}(\mathbf{\Phi}) - R(\hat{\mathbf{\Phi}})\mathrm{Den}(\mathbf{\Phi}).$$

To see this, first note that $Q_{\hat{\mathbf{\Phi}}}(\hat{\mathbf{\Phi}}) = 0$. Thus, if $Q_{\hat{\mathbf{\Phi}}}(\bar{\mathbf{\Phi}}) > Q_{\hat{\mathbf{\Phi}}}(\hat{\mathbf{\Phi}})$, then $\mathrm{Num}(\bar{\mathbf{\Phi}}) > R(\hat{\mathbf{\Phi}})\mathrm{Den}(\bar{\mathbf{\Phi}})$, and hence $R(\bar{\mathbf{\Phi}}) > R(\hat{\mathbf{\Phi}})$.

Unfortunately, the growth transform can not be applied directly to $Q_{\hat{\mathbf{\Phi}}}(\mathbf{\Phi})$, as it might have negative coefficients. To ensure nonnegativity, the growth transform is instead applied to

$$S_{\hat{\mathbf{\Phi}}}(\mathbf{\Phi}) = Q_{\hat{\mathbf{\Phi}}}(\mathbf{\Phi}) + C(\mathbf{\Phi}),$$

where

$$C(\mathbf{\Phi}) = \kappa\left(\sum_{j,i}\varphi_i^j + 1\right)^r$$

has constant value over E, since $\sum_i \varphi_i^j = 1$, and r denotes the maximal order of $Q_{\hat{\mathbf{\Phi}}}(\mathbf{\Phi})$. Hence, $C(\mathbf{\Phi})$ adds a constant κ to every monomial in $Q_{\hat{\mathbf{\Phi}}}(\mathbf{\Phi})$. This constant κ must be chosen such that $S_{\hat{\mathbf{\Phi}}}(\mathbf{\Phi})$ has nonnegative coefficients for every $\hat{\mathbf{\Phi}}$. Thus, $S_{\hat{\mathbf{\Phi}}}(\mathbf{\Phi})$ has positive coefficients and still has the same important property as $Q_{\hat{\mathbf{\Phi}}}(\mathbf{\Phi})$. This polynomial with positive coefficients can now be considered for the growth transform in Eq. (40).

As easily can be verified, the partial derivative of $S_{\hat{\mathbf{\Phi}}}(\mathbf{\Phi})$ can be expressed in terms of $\frac{\partial \log R(\hat{\mathbf{\Phi}})}{\partial \varphi_i^j}$, according to

$$\frac{\partial S_{\hat{\mathbf{\Phi}}}(\hat{\mathbf{\Phi}})}{\partial \varphi_i^j} = \mathrm{Num}(\hat{\mathbf{\Phi}})\frac{\partial \log R(\hat{\mathbf{\Phi}})}{\partial \varphi_i^j} + D,$$

where $D = \kappa r(J+1)^{r-1}$ is the derivative of $C(\mathbf{\Phi})$. Plugging this result into Eq. (40), we finally obtain the extended Baum-Welch re-estimation equation for discrete probability distributions of the form

$$\bar{\varphi}_i^j \leftarrow \frac{\hat{\varphi}_i^j\left(\frac{\partial \log R(\hat{\mathbf{\Phi}})}{\partial \varphi_i^j} + D\right)}{\sum_{i'}\hat{\varphi}_{i'}^j\left(\frac{\partial \log R(\hat{\mathbf{\Phi}})}{\partial \varphi_{i'}^j} + D\right)}, \tag{41}$$

where the $\bar{\varphi}_i^j$ denotes the updated parameters, and constant D must be chosen to be sufficiently large.

Appendix B: MM-HMM EBW Algorithm

The implementation of the EBW algorithm for maximizing the margin, i.e. MM-HMM EBW algorithm, is stated in Algorithm 1.

The E-step of the MM-HMM EBW algorithm using the approximation of $\frac{\partial \log d_{\mathbf{\Theta}}^n}{\partial \mathbf{\Theta}}$ (see Eq. (30)) is depicted in Algorithm 2.

In Algorithm 3, the M-step of the MM-HMM EBW algorithm using parameter updates of Eq. (12) is illustrated.

7. Acknowledgements

This work was supported by the Austrian Science Fund (Project number P27803-N15).

Input: $\{X_1, \ldots X_C\}$
Output: $\rho_c, \pi_{c,i}, a_{c,i,j}, \alpha_{c,i,m}, \boldsymbol{\mu}_{c,i,m}, \boldsymbol{\Sigma}_{c,i,m} \quad \forall c \in \{1,\ldots,C\},\ \forall i,j \in \{1,\ldots,S\}; \forall m \in \{1,\ldots,M\}$
Initialization: For each c, $\mathbf{\Theta}_c = \{\pi_{c,i}, a_{c,i,j}, \alpha_{c,i,m}, \boldsymbol{\mu}_{c,i,m}, \boldsymbol{\Sigma}_{c,i,m}\}_{i,j\in\{1,\ldots,S\},m\in\{1,\ldots,M\}}$ is initialized by MLE using the EM-algorithm. The class prior is set to the normalized class frequency, i.e. $\rho_c = \frac{N_c}{N}$
while $J(\mathcal{X}|\mathbf{\Theta})$ *not converged* **do**
  Determine: $\mathcal{X}^1, \mathcal{X}^2, \mathcal{X}^3$ based on $(d_\Theta^n)^\lambda$
  Determine: $s^n \quad \forall n \in \{1,\ldots,N\}$ based on $\mathcal{X}^1, \mathcal{X}^2, \mathcal{X}^3$
  E-Step (see Algorithm 2)
  Determine D (see Section 4.4)
  M-Step (see Algorithm 3)
end

Algorithm 1: Discriminative Margin-based training of HMMs (MM-HMM EBW algorithm).

E-Step:
for $c \leftarrow 1$ **to** C **do**
  $r_c^{n,\eta} \leftarrow \frac{(p(\mathbf{x}^n|\mathbf{\Theta}_c)\rho_c)^\eta}{\sum_{c'\neq c^n}(p(\mathbf{x}^n|\mathbf{\Theta}_{c'})\rho_{c'})^\eta}$
  $\frac{\partial log d_{\mathbf{\Theta}}^n}{\partial \rho_c} \leftarrow \frac{\frac{1}{\rho_c}\left[z_c^n - z_c^n \cdot r_c^{n,\eta}\right]}{\sum_{c'=1}^C\left|\frac{1}{\rho_{c'}}\left[z_c^n - z_c^n \cdot r_c^{n,\eta}\right]\right|}$
  $\partial\rho_c \leftarrow \sum_{n=1}^N s^n \frac{\partial \log d_{\mathbf{\Theta}}^n}{\partial \rho_c}$
  for $i \leftarrow 1$ **to** S **do**
    $\frac{\partial \log d_{\mathbf{\Theta}}^n}{\partial \pi_{c,i}} \leftarrow \frac{\frac{1}{\pi_{c,i}}\left[u_{c,i,1}^n - \ddot{u}_{c,i,1}^n \cdot r_c^{n,\eta}\right]}{\sum_{i'=1}^S\left|\frac{1}{\pi_{c,i'}}\left[u_{c,i',1}^n - \ddot{u}_{c,i',1}^n \cdot r_c^{n,\eta}\right]\right|}$
    $\partial\pi_{c,i} \leftarrow \sum_{n=1}^N s^n \frac{\partial \log d_{\mathbf{\Theta}}^n}{\partial \pi_{c,i}}$
    for $j \leftarrow 1$ **to** S **do**
      $\frac{\partial \log d_{\mathbf{\Theta}}^n}{\partial a_{c,i,j}} \leftarrow \frac{\frac{1}{a_{c,i,j}}\left[y_{c,i,j}^n - \ddot{y}_{c,i,j}^n \cdot r_c^{n,\eta}\right]}{\sum_{j'=1}^S\left|\frac{1}{a_{c,i,j'}}\left[y_{c,i,j'}^n - \ddot{y}_{c,i,j'}^n \cdot r_c^{n,\eta}\right]\right|}$
      $\partial a_{c,i,j} \leftarrow \sum_{n=1}^N s^n \frac{\partial \log d_{\mathbf{\Theta}}^n}{\partial a_{c,i,j}}$
    end
    for $m \leftarrow 1$ **to** M **do**
      $\gamma_{c,i,m,t}^n \leftarrow \frac{\alpha_{c,i,m}\cdot\mathcal{N}(\mathbf{x}_t^n|\boldsymbol{\mu}_{c,i,m},\boldsymbol{\Sigma}_{c,i,m})}{b_{c,i}(\mathbf{x}_t^n)} \quad \forall n \in \{1,\ldots,C\}$
      $\frac{\partial \log d_{\mathbf{\Theta}}^n}{\partial \alpha_{c,i,m}} \leftarrow \frac{\frac{1}{\alpha_{c,i,m}}\sum_{t=1}^{T^n}\left[\gamma_{c,i,m,t}^n\left(u_{c,i,t}^n - \ddot{u}_{c,i,t}^n \cdot r_c^{n,\eta}\right)\right]}{\sum_{m'=1}^M\left|\frac{1}{\alpha_{c,i,m'}}\sum_{t=1}^{T^n}\left[\gamma_{c,i,m',t}^n\left(u_{c,i,t}^n - \ddot{u}_{c,i,t}^n \cdot r_c^{n,\eta}\right)\right]\right|}$
      $\partial\alpha_{c,i,m} \leftarrow \sum_{n=1}^N s^n \frac{\partial \log d_{\mathbf{\Theta}}^n}{\partial \alpha_{c,i,m}}$
    end
  end
end

Algorithm 2: E-step of the MM-HMM EBW algorithm.

References

Bahl, L.R., Brown, P.F., de Souza, P.V., & Mercer, R.L. 1986. Maximum Mutual Information Estimation of HMM Parameters for Speech Recognition. *Pages 49–52 of: IEEE Conf. on Acoustics, Speech, and Signal Proc.*

Baum, L.E., & Eagon, J.A. 1967. An inequality with applications to statistical prediction for functions of Markov processes and to a model of ecology. *Bull. Amer. Math. Soc.*, **73**, 360–363.

Berndt, D.J., & Clifford, J. 1994. Using Dynamic Time Warping to Find Patterns in Time Series. *Pages 359–370 of: KDD Workshop*.

Bishop, C.M. 2006. *Pattern recognition and machine learning*. Springer.

Burges, C.J.C. 1998. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, **2**(2), 121–167.

Buza, K., Nanopoulos, A., Schmidt-Thieme, L., & Koller, J. 2011. Fast Classification of Electrocardiograph Signals via Instance Selection. *Pages 9–16 of: Intern. Conf. on Healthcare Informatics, Imaging and Systems Biology (HISB)*.

Campbell, Sean D, & Diebold, Francis X. 2005. Weather forecasting for weather derivatives. *Journal of the American Statistical Association*, **100**(469), 6–16.

Cao, LJ, & Tay, FEH. 2003. Support vector machine with adaptive parameters

M-Step:
for $c \leftarrow 1$ **to** C **do**
  $\bar{\rho}_c \leftarrow \frac{\rho_c(\partial\rho_c + D)}{\sum_{c'=1}^{C} \rho_{c'}(\partial\rho_{c'} + D)}$
  for $i \leftarrow 1$ **to** S **do**
    $\bar{\pi}_{c,i} \leftarrow \frac{\pi_{c,i}(\partial\pi_{c,i} + D)}{\sum_{i'=1}^{S} \pi_{c',i}(\partial\pi_{c',i} + D)}$
    for $j \leftarrow 1$ **to** S **do**
      $\bar{a}_{c,i,j} \leftarrow \frac{a_{c,i,j}(\partial a_{c,i,j} + D)}{\sum_{j'=1}^{S} a_{c,i,j'}(\partial a_{c,i,j'} + D)}$
    end
    for $m \leftarrow 1$ **to** M **do**
      $\bar{\alpha}_{c,i,m} \leftarrow \frac{\alpha_{c,i,m}(\partial\alpha_{c,i,m} + D)}{\sum_{m'=1}^{M} \alpha_{c,i,m'}(\partial\alpha_{c,i,m'} + D)}$
      $\bar{\boldsymbol{\mu}}_{c,i,m} \leftarrow \frac{\sum_{n=1}^{N} s^n \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}\left(u^n_{c,i,t}\cdot\tilde{u}^n_{c,i,t}\cdot r_c^{n,\eta}\right)\mathbf{x}_t^n\right] + D\boldsymbol{\mu}_{c,i,m}}{\sum_{n=1}^{N} s^n \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}\left(u^n_{c,i,t}\cdot\tilde{u}^n_{c,i,t}\cdot r_c^{n,\eta}\right)\right] + D}$
      $\boldsymbol{\Sigma}_{c,i,m} \leftarrow \frac{g_{c,i,m} + D(\boldsymbol{\Sigma}_{c,i,m} + (\boldsymbol{\mu}_{c,i,m})^2)}{\sum_{n=1}^{N} s^n \sum_{t=1}^{T^n} \left[\gamma^n_{c,i,m,t}\left(u^n_{c,i,t}\cdot\tilde{u}^n_{c,i,t}\cdot r_c^{n,\eta}\right)\right] + D} - (\bar{\boldsymbol{\mu}}_{c,i,m})^2$
      $\boldsymbol{\mu}_{c,i,m} \leftarrow \bar{\boldsymbol{\mu}}_{c,i,m}$
    end
    $\alpha_{c,i,m} \leftarrow \bar{\alpha}_{c,i,m} \quad \forall m$
    $a_{c,i,j} \leftarrow \bar{a}_{c,i,j} \quad \forall j$
  end
  $\pi_{c,i} \leftarrow \bar{\pi}_{c,i} \quad \forall i$
end
$\rho_c \leftarrow \bar{\rho}_c \quad \forall c$

Algorithm 3: M-step of the MM-HMM EBW algorithm.

in financial time series forecasting. *IEEE Transactions on Neural Networks*, **14**(6), 1506–1518.

Ebrahimpour, R., & Hamedi, S. 2009. Hand written digit recognition by multiple classifier fusion based on decision templates approach. *Pages – of: International Conference on Computer, Electrical, Systems Science, and Engineering (CESSE)*.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., & Bengio, S. 2010. Why Does Unsupervised Pre-training Help Deep Learning? *Journal of Machine Learning Research*, **11**, 625–660.

Gopalakrishnan, O., Kanevsky, D., Nàdas, A., & Nahamoo, D. 1991. An inequality for rational functions with applications to some statistical estimation problems. *IEEE Transactions on Information Theory*, **37**(1), 107–113.

Guo, Y., Wilkinson, D.F., & Schuurmans, D. 2005. Maximum Margin Bayesian Networks. *Pages 233–242 of: International Conference on Uncertainty in Artificial Intelligence (UAI)*.

Heigold, G., Deselaers, T., Schlüter, R., & Ney, H. 2008. Modified MMI/MPE: A direct evaluation of the margin in speech recognition. *Pages 384–391 of: International Conference on Machine Learning (ICML)*.

Heigold, G., Dreuw, P., Hahn, S., Schlüter, R., & Ney, H. 2010. Margin-Based Discriminative Trainign for String Recognition. *Pages 917–925 of: IEEE Journal of Selected Topics in Signal Processing*, vol. 4.

Heigold, G., Ney, H., Schlüter, R., & Wiesler, S. 2012. Discriminative Training for Automatic Speech Recognition. *Pages 58–69 of: IEEE Signal Processing Magazine*, vol. 29.

Hiroaki, S., & Chiba, S. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **26**(1), 43–49.

Huber, P.J. 1964. Robust Estimation of a Location Parameter. *Annals of Statistics*, **53**, 73–101.

Jebara, T. 2001. *Discriminative, generative and imitative learning*. Ph.D. thesis, Media Laboratory, MIT.

Juang, B.-H., Chou, W., & Lee, C.-H. 1997. Minimum Classification Error Rate Methods for Speech Recognition. *IEEE Transactions on Speech and Ausio Processing*, **5**(3), 257–265.

Jurafsky, D., & Martin, J. H. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2ndt edn. Pearson Education International.

Kim, M., & Pavlovic, V. 2011. Sequence classification via large margin hidden Markov models. *Data Mining and Knowledge Discovery*, **23**(2), 322–344.

Klautau, A., Jevtić, N., & Orlitsky, A. 2003. Discriminative Gaussian Mixture models: A comparison with kernel classifiers. *Pages 353 – 360 of: Inter. Conf. on Machine Learning (ICML)*.

Lamel, L., Kassel, R., & Seneff, S. 1986. Speech database development: Design and analysis of the acoustic-phonetic corpus. *In: DARPA Speech Recognition Workshop, Report No. SAIC-86/1546*.

Merialdo, B. 1988. Phonetic recognition using hidden Markov models and maximum mutual information training. *Pages 111–114 of: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Mouine, S., Yahiaoui, I., & Verroust-Blondet, A. 2013. A shape-based approach for leaf classification using multiscaletriangular representation. *Pages – of: ACM Conference on Multimedia Retrieval*.

Nesterov, Y. 2005. Smooth minimization of non-smooth functions. *Mathematical programming*, **103**(1), 127–152.

Ng, A.Y., & Jordan, M. 2002. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and Naive Bayes. *In: NIPS 14*.

Normandin, Y., & Morgera, S.D. 1991. An improved MMIE training algorithm for speaker-independent small vocabulary, continuous speech recognition. *Pages 537–540 of: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Normandin, Y., Cardin, R., & De Mori, R. 1994. High-performance connected digit recognition using maximum mutual information estimation. *IEEE Trans. on Speech and Audio Proc.*, **2**(2), 299–311.

Pernkopf, F., & Bilmes, J. 2010. Efficient Heuristics for Discriminative Structure Learning of Bayesian Network Classifiers. *Journal of Machine Learning Research*, **11**, 2323–2360.

Pernkopf, F., & Wohlmayr, M. 2010. Large Margin Learning of Bayesian Classifiers based on Gaussian Mixture Models. *Pages 50–66 of: European Conference on Machine Learning (ECML)*.

Pernkopf, F., Wohlmayr, M., & Tschiatschek, S. 2012. Maximum Margin Bayesian Network Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**(3), 521–532.

Pernkopf, F., Peharz, R., & Tschiatschek, S. 2014. Introduction to Probabilistic graphical models. *Academic Press Library in Signal Processing*, **1**(Ch. 18), 989–1064.

Rabiner, L.R. 1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**(2), 257–286.

Radovanović, M., Nanopoulos, A., & Ivanović, M. 2009. Nearest neighbors in high-dimensional data: The emergence and influence of hubs. *Pages – of: International Conference on Machine Learning (ACM)*.

Saon, G., & Povey, D. 2008. Penalty function maximization for large margin HMM training. *Pages 920–923 of: INTERSPEECH*.

Schölkopf, B., & Smola, A.J. 2001. *Learning with kernels: Support Vector Machines, regularization, optimization, and beyond*. MIT Press.

Sha, F., & Saul, L. 2007a. Comparison of Large Margin Training to other Discriminative Methods for Phonetic Recognition by Hidden Markov Models. *Pages 313–316 of: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Sha, F., & Saul, L.K. 2007b. Large margin hidden Markov models for automatic speech recognition. *Pages 1249–1256 of: Advances in Neural Information Processing Systems 19*.

Taskar, B., Guestrin, C., & Koller, D. 2004. Max-Margin Markov Networks. *In: Advances in Neural Information Processing Systems (NIPS)*.

Valtchev, V., Odell, J. J., Woodland, P.C., & Young, S.J. 1997. MMIE training of large vocabulary recognition systems. *Pages 303–314 of: Speech Communication*, vol. 22.

Vapnik, V. 1998. *Statistical learning theory*. Wiley & Sons.

Woodland, P.C., & Povey, D. 2002. Large scale discriminative training of hidden Markov models for speech recognition. *Computer Speech and Language*, **16**, 25–47.