Discriminative Kernel-Based Phoneme Sequence Recognition

Joseph Keshet [1] Samy Bengio [2]
Dan Chazan [3] Shai Shalev-Shwartz [1]
Yoram Singer [4]

IDIAP–RR 06-14

April 2006

[1] School of Computer Science & Engineering, The Hebrew University, Jerusalem, Israel

[2] IDIAP Research Institute, 1920 Martigny, Switzerland

[3] IBM Research Labs, Haifa, Israel

[4] Google Labs, Mountain View, CA

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 06-14

Discriminative Kernel-Based Phoneme Sequence Recognition

Joseph Keshet Samy Bengio Dan Chazan Shai Shalev-Shwartz
Yoram Singer

April 2006

Abstract. We describe a new method for phoneme sequence recognition given a speech utterance. In contrast to HMM-based approaches, our method uses a kernel-based discriminative training procedure in which the learning process is tailored to the goal of minimizing the Levenshtein distance between the predicted phoneme sequence and the correct sequence. The phoneme sequence predictor is devised by mapping the speech utterance along with a proposed phoneme sequence to a vector-space endowed with an inner-product that is realized by a Mercer kernel. Building on large margin techniques for predicting whole sequences, we are able to devise a learning algorithm which distills to separating the correct phoneme sequence from all other sequences. We describe an iterative algorithm for learning the phoneme sequence recognizer and further describe an efficient implementation of it. We present initial encouraging experimental results with the TIMIT and compare the proposed method to an HMM-based approach.

1 Introduction

Most previous work on phoneme sequence recognition has focused on Hidden Markov Models (HMM). See for example [10, 7, 2] and the references therein. Despite their popularity, HMM-based approaches have several drawbacks such as convergence of the EM procedure to local maximum and overfitting effects due to the large number of parameters. Moreover, HMMs do not faithfully reflect the underlying structure of speech signals as they assume conditional independence of observations given the state sequence [11] and often require uncorrelated acoustic features [18]. Another problem with HMMs is that they do not directly address discriminative tasks. In particular, for the task of phoneme sequence prediction, HMMs as well as other generative models, are not trained to minimize the Levenshtein distance between the model-based predicted phoneme sequence and the correct one.

In this report we propose an alternative approach for phoneme sequence recognition that builds upon recent work on discriminative supervised learning and overcome the inherent problems of the HMM approaches. The advantage of discriminative learning algorithms stems from the fact that the objective function used during the learning phase is tightly coupled with the decision task one needs to perform. In addition, there is both theoretical and empirical evidence that discriminative learning algorithms are likely to outperform generative models for the same task (see for instance [17, 5]). One of the main goals of this work is to extend the notion of discriminative learning to the complex task of phoneme sequence prediction.

Our proposed method is based on recent advances in kernel machines and large margin classifiers for sequences [15, 14], which in turn build on the pioneering work of Vapnik and colleagues [17, 5]. The phoneme sequence recognizer we devise is based on mapping the speech signal along with the target phoneme sequence into a vector-space endowed with an inner-product that is defined by a kernel operator. One of the well-known discriminative learning algorithms is the support vector machine (SVM), which has already been successfully applied in speech applications [8, 13]. Building on techniques used for learning SVMs, our phoneme sequence recognizer distills to a classifier in this vector-space which is aimed at separating correct phoneme sequences from incorrect ones. The classical SVM algorithm is designed for simple decision tasks such as binary classification and regression. Hence, its exploitation in speech systems so far has also been restricted to simple decision tasks such as phoneme classification. The phoneme sequence recognition problem is more complex, since we need to predict a whole sequence rather than a single number. Previous kernel machine methods for sequence prediction [15, 16] introduce optimization problems which require long run-time and high memory resources, and are thus problematic for the large datasets that are typically encountered in speech processing. We propose an alternative approach which uses an efficient iterative algorithm for learning a discriminative phoneme sequence predictor by traversing the training set a single time.

This report is organized as follows. In Sec. 2 we formally introduce the phoneme sequence recognition problem. Next, our specific learning method is described in Sec. 3. Our method is based on non-linear phoneme recognition function using Mercer kernels. A specific kernel for our task is presented in Sec. 4. We present preliminary experimental results in Sec. 5 and conclude with a discussion in Sec. 6.

2 Problem Setting

In the problem of phoneme sequence recognition, we are given a speech utterance and our goal is to predict the phoneme sequence corresponding to it. We represent a speech signal as a sequence of acoustic feature-vectors $\bar{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$, where $\mathbf{x}_t \in \mathbb{R}^d$ for all $1 \leq t \leq T$. We denote the domain of the acoustic feature-vectors by $\mathcal{X} \subset \mathbb{R}^d$. Each utterance corresponds to a sequence of phoneme symbols. Formally, we denote each phoneme symbol by $p \in \mathcal{P}$, where $\mathcal{P}$ is a set of phoneme symbols, and we denote the sequence of phoneme symbols by $\bar{p} = (p_1, \ldots, p_K)$. Furthermore, we denote by $s_k \in \mathbb{N}$ the start time of phoneme p_k (in frame units) and we denote by $\bar{s} = (s_1, \ldots, s_K)$ the sequence of all phoneme start-times. Naturally, the length of the speech signal and hence the number of phonemes varies from one utterance to another and thus T and K are not fixed. We denote by $\mathcal{P}^*$ (and similarly $\mathcal{X}^*$ and $\mathbb{N}^*$) the set of all finite-length sequences over $\mathcal{P}$. Our goal is to learn a function f that predicts the correct phoneme sequence given an acoustic sequence. That is, f is a function from $\mathcal{X}^*$ to the set of finite-length sequences over the domain of phoneme symbols, $\mathcal{P}^*$. We also refer to f as a phoneme sequence recognizer or predictor.

Previous work typically employed HMMs for phoneme recognition. In HMMs, we assume that the speech signal $\bar{\mathbf{x}}$ is generated from a phoneme sequence $\bar{p}$ aligned by a start time sequence $\bar{s}$ according to probability density function $\Pr(\bar{\mathbf{x}}, \bar{p}, \bar{s}) = \Pr(\bar{\mathbf{x}}|\bar{p}, \bar{s}) \Pr(\bar{s}|\bar{p}) \Pr(\bar{p})$. The maximum a posteriori (MAP) prediction of the phoneme sequence is

$$\bar{p}' = \arg\max_{\bar{p}} \max_{\bar{s}} \Big[\log \Pr(\bar{\mathbf{x}}|\bar{p}, \bar{s}) + \log \Pr(\bar{s}|\bar{p}) + \log \Pr(\bar{p}) \Big] . \tag{1}$$

The likelihood value $\Pr(\bar{\mathbf{x}}|\bar{p}, \bar{s})$ is often referred to as the acoustic model, $\Pr(\bar{s}|\bar{p})$ is referred to as the phoneme duration model, and $\Pr(\bar{p})$ is termed the language model. To facilitate efficient calculation of $\bar{p}'$, practical generative models assume that the probability functions may be further decomposed into basic probability functions. It is often assumed that $\bar{p}$ and $\bar{s}$ generate a sequence of hidden states $\bar{q} = (q_1, \ldots, q_T)$, where each q_t is in a predefined set of hidden states, $\mathcal{Q}$, and the likelihood of the speech utterance given $\bar{q}$ is

$$\log \Pr(\bar{\mathbf{x}}|\bar{q}) = \sum_{t=1}^{T} \log \Pr(\mathbf{x}_t|q_t) + \sum_{t=2}^{T} \log \Pr(q_t|q_{t-1}) \ . \tag{2}$$

These simplifying assumptions lead to a model which is quite inadequate for the purpose of generating natural speech utterances. Nonetheless, the likelihood value $\Pr(\bar{\mathbf{x}}|\bar{q})$ is used as an assessment for the quality of the phoneme sequence, $\bar{p}$. The learning phase of the HMM aims at determining the basic probability functions from a training set of examples. The learning objective is to estimate the functions $\Pr(\mathbf{x}_t|q_t)$ and $\Pr(q_t|q_{t-1})$ so as to maximize the likelihood of the training set. Given these functions, the prediction $\bar{p}'$ is calculated in the so-called inference phase which can be performed efficiently using dynamic programming.

The ultimate goal of the phoneme sequence prediction is usually to minimize the Levenshtein distance between the predicted sequence and the correct one. The HMM, however, tries to maximize the likelihood of the parameters and there is no clear connection between the parameters' likelihood and the Levenshtein distance goal. Throughout this report, we denote by $\gamma(\bar{p}, \bar{p}')$ the Levenshtein distance between the predicted phoneme sequence $\bar{p}'$ and the true phoneme sequence $\bar{p}$. In the next section we present an algorithm which directly aims at minimizing the Levenshtein distance between the predicted phoneme sequence and the correct phoneme sequence.

3 The Learning Algorithm

In this section we describe a discriminative supervised learning algorithm for learning a phoneme sequence recognizer f from a training set of examples. Each example in the training set is composed of an acoustic signal $\bar{\mathbf{x}}$, a sequence of phonemes, $\bar{p}$, and a sequence of phoneme start-times, $\bar{s}$.

Our construction is based on a predefined set of feature functions $\{\phi_j\}_{j=1}^n$, each of which takes the form $\phi_j : \mathcal{X}^\star \times (\mathcal{P} \times \mathbb{N})^* \to \mathbb{R}$. Thus, the input of each function is an acoustic representation, $\bar{\mathbf{x}}$, together with a candidate phoneme symbol sequence $\bar{p}$ and a candidate phoneme start time sequence $\bar{s}$. Each feature function returns a scalar which, intuitively, represents the confidence in the suggested phoneme sequence. Similarly to HMMs, each feature function captures a local matching between the speech utterance and the suggested phoneme sequence. For example, one feature function can sum the number of times phoneme p comes after phoneme p', while other feature functions may extract properties of each acoustic feature vector $\mathbf{x}_t$ provided that phoneme p pronounced at time t. Those feature functions are reminiscent of the local probabilistic models appearing in Eq. (2). The description of the concrete form of each feature function is differed to Sec. 4. For brevity, we denote by $\boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}, \bar{s})$ the vector whose jth element is $\phi_j(\bar{\mathbf{x}}, \bar{p}, \bar{s})$.

Recall that our goal is to learn a phoneme sequence recognizer f, which takes as input a sequence of acoustic features $\bar{\mathbf{x}}$ and returns a sequence of phoneme symbols $\bar{p}$. The form of the function f we use is

$$f(\bar{\mathbf{x}}) = \arg\max_{\bar{p}} \max_{\bar{s}} \ \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}, \bar{s}) \ , \tag{3}$$

where $\mathbf{w} \in \mathbb{R}^n$ is a vector of importance weights that should be learned. In words, f returns a suggestion for a phoneme sequence by maximizing a weighted sum of the scores returned by the feature functions $\{\phi_j\}$. Learning the weight vector $\mathbf{w}$ is analogous to the estimation of the parameters of the local probability functions in HMMs (Eq. (2)). Our approach, however, does not require $\mathbf{w}$ to take a probabilistic form. The maximization defined by Eq. (3) is over an exponentially large number of all possible phoneme sequences. Nevertheless, as in HMMs, if each feature function, ϕ_j, is decomposable, the optimization in Eq. (3) can be efficiently calculated using a dynamic programming procedure.

We now describe a simple iterative algorithm for learning the weight vector $\mathbf{w}$. The algorithm receives as input a training set $S = \{(\bar{\mathbf{x}}_1, \bar{p}_1, \bar{s}_1), \ldots, (\bar{\mathbf{x}}_m, \bar{p}_m, \bar{s}_m)\}$ of examples. Initially we set $\mathbf{w} = \mathbf{0}$. At each iteration the algorithm updates $\mathbf{w}$ according to the ith example in S as we now describe. Denote by $\mathbf{w}_{i-1}$ the value of the weight vector before the ith iteration. Let $(\bar{p}'_i, \bar{s}'_i)$ be the predicted phoneme sequence for the ith example according to $\mathbf{w}_{i-1}$,

$$(\bar{p}'_i, \bar{s}'_i) = \arg\max_{(\bar{p},\bar{s})} \ \mathbf{w}_{i-1} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i, \bar{p}, \bar{s}) \ . \tag{4}$$

We now observe the Levenshtein distance, $\gamma(\bar{p}_i, \bar{p}'_i)$, between the predicted phoneme sequence $\bar{p}'_i$ and the correct phoneme sequence $\bar{p}_i$. If $\gamma(\bar{p}_i, \bar{p}'_i)$ is smaller than a predefined constant, γ_0, then we refer to our prediction as a correct prediction and thus keep the weight vector intact, $\mathbf{w}_i = \mathbf{w}_{i-1}$. Otherwise, we would like to update the weight vector $\mathbf{w}$ so that the weighted score of the correct phoneme sequence, $\mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{s}_i)$, increases while the weighted score of the (incorrectly) predicted phoneme sequence, $\mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i, \bar{p}'_i, \bar{s}'_i)$, decreases. This requirement is met by using the following update rule

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \alpha_i \Delta\boldsymbol{\phi}_i \ , \tag{5}$$

where $\Delta\boldsymbol{\phi}_i = \boldsymbol{\phi}(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{s}_i) - \boldsymbol{\phi}(\bar{\mathbf{x}}_i, \bar{p}'_i, \bar{s}'_i)$. The value of the scalar α_i is based on the Levenshtein distance $\gamma(\bar{p}_i, \bar{p}'_i)$, the different scores that $\bar{p}_i$ and $\bar{p}'_i$ received according to $\mathbf{w}_{i-1}$, and a parameter C. Formally,

$$\alpha_i = \min\left\{ C \ , \ \frac{\max\{\gamma(\bar{p}_i, \bar{p}'_i) - \mathbf{w}_{i-1} \cdot \Delta\boldsymbol{\phi}_i, 0\}}{\|\Delta\boldsymbol{\phi}_i\|^2} \right\} \ . \tag{6}$$

The parameter C serves as a complexity-accuracy trade-off parameter as in the SVM algorithm (see [5]).

The specific definition of α_i that we employ is based on an ongoing work on online learning algorithms appearing in [4, 14, 9]. These papers demonstrated that, under some mild technical conditions, the cumulative Levenshtein distance of the iterative procedure, $\sum_{i=1}^{m} \gamma(\bar{p}_i, \bar{p}'_i)$, is likely to be small. Moreover, it can be shown [1] that if the cumulative Levenshtein distance of the iterative procedure is small, there exists at least one weight vector among the vectors $\{\mathbf{w}_1, \ldots, \mathbf{w}_m\}$ which attains small averaged Levenshtein distance on unseen examples as well. To find this weight vector we simply calculate the averaged Levenshtein distance attained by each of the weight vectors on a validation set. A pseudo-code of our algorithm is given in Fig. 1.

To conclude this section, we extend the family of linear phoneme sequence recognizers given in Eq. (3) to non-linear recognition functions. This extension is based on Mercer kernels often used in SVM algorithms [17]. Recall that the update rule of the algorithm is $\mathbf{w}_i = \mathbf{w}_{i-1} + \alpha_i \Delta\boldsymbol{\phi}_i$ and that the initial weight vector is $\mathbf{w}_0 = \mathbf{0}$. Thus, $\mathbf{w}_i$ can be rewritten as, $\mathbf{w}_i = \sum_{j=1}^{i} \alpha_j \Delta\boldsymbol{\phi}_j$ and f can be

```
**Input:** training set $S = \{(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{s}_i)\}_{i=1}^{m}$ ; validation set $S_{\text{val}} = \{(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{s}_i)\}_{i=1}^{m_{\text{val}}}$ ;
    parameter $C$
**Initialize:** $\mathbf{w}_0 = \mathbf{0}$
**For** $i = 1, \ldots, m$
  **Predict:** $(\bar{p}'_i, \bar{s}'_i) = \arg\max_{\bar{p}, \bar{s}} \mathbf{w}_{i-1} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i, \bar{p}, \bar{s})$
  **Set:** $\Delta\boldsymbol{\phi}_i = \boldsymbol{\phi}(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{s}_i) - \boldsymbol{\phi}(\bar{\mathbf{x}}_i, \bar{p}'_i, \bar{s}'_i)$
  **Set:** $\ell(\mathbf{w}_{i-1}; \bar{\mathbf{x}}_i, \bar{p}_i, \bar{s}_i) = \max\{\gamma(\bar{p}_i, \bar{p}'_i) - \mathbf{w}_{i-1} \cdot \Delta\boldsymbol{\phi}_i, 0\}$
  **If** $\ell(\mathbf{w}_{i-1}; \bar{\mathbf{x}}_i, \bar{p}_i, \bar{s}_i) > 0$
    **Set:** $\alpha_i = \min\left\{C \ , \ \frac{\ell(\mathbf{w}_{i-1}; \bar{\mathbf{x}}_i, \bar{p}_i, \bar{s}_i)}{\|\Delta\boldsymbol{\phi}_i\|^2}\right\}$
    **Update:**   $\mathbf{w}_i = \mathbf{w}_{i-1} + \alpha_i \cdot \Delta\boldsymbol{\phi}_i$
**Output:** The weight vector $\mathbf{w}^*$ which achieves best performance on a validation
    set $S_{\text{val}}$:
      $\mathbf{w}^* = \arg \min_{\mathbf{w} \in \{\mathbf{w}_1, \ldots, \mathbf{w}_m\}} \sum_{j=1}^{m_{\text{val}}} \gamma\big(\bar{p}_j, f(\bar{\mathbf{x}}_j)\big)$
```

Figure 1: An iterative algorithm.

rewritten as

$$f(\bar{\mathbf{x}}) = \arg\max_{\bar{p}} \ \max_{\bar{s}} \ \sum_{j=1}^{i} \alpha_j \Big(\Delta\boldsymbol{\phi}_j \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}, \bar{s})\Big) \ . \tag{7}$$

By replacing the inner-product in Eq. (7) with a general kernel operator $\mathcal{K}(\cdot, \cdot)$ that satisfies Mercer's conditions [17], we obtain a non-linear phoneme recognition function. It is easy to verify that the definition of α_i given in Eq. (6) can also be rewritten using the kernel operator.

4 Non-Linear Feature Functions

In this section we describe a specific set of feature functions $\{\phi_j\}$. As mentioned in the previous section, we aim at designing non-linear phoneme sequence recognizers using Mercer kernels. Therefore, rather than describing the feature functions ϕ_j we describe a kernel operator which calculates implicitly the inner-product $\boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}, \bar{s}) \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}', \bar{p}', \bar{s}')$. To simplify our notation we denote by $\mathbf{z}$ the triplet $(\bar{\mathbf{x}}, \bar{p}, \bar{s})$ and similarly $\mathbf{z}'$ denotes $(\bar{\mathbf{x}}', \bar{p}', \bar{s}')$. The kernel operators $\mathcal{K}(\mathbf{z}, \mathbf{z}')$ we devise can be written as a weighted sum of three kernel operators $\mathcal{K}(\mathbf{z}, \mathbf{z}') = \sum_{i=1}^{3} \beta_i \mathcal{K}_i(\mathbf{z}, \mathbf{z}')$, where β_i are positive parameters. In the following we describe the three kernel operators we use.

The first kernel operator, $\mathcal{K}_1$, is reminiscent of the acoustic model appears in HMMs. First, for each phoneme $p \in \mathcal{P}$, let $T_p(\mathbf{z})$ be the set of all frame times in which the phoneme p is uttered. That is, $T_p(\mathbf{z}) = \{t : \exists k, \, p_k = p \wedge s_k \leq t \leq s_{k+1}\}$. Using this definition, the first kernel operator is defined to be

$$\mathcal{K}_1(\mathbf{z}, \mathbf{z}') \ = \ \sum_{p \in \mathcal{P}} \sum_{t \in T_p(\mathbf{z})} \sum_{\tau \in T_p(\mathbf{z}')} \exp\left(-\frac{\|\mathbf{x}_t - \mathbf{x}'_\tau\|^2}{2\sigma^2}\right) \quad ,$$

where σ is a predefined constant.

Recall that the inner-product of $\Delta\boldsymbol{\phi}_i$ with $\boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}, \bar{s})$ in Eq. (7) is replaced by $\mathcal{K}(\mathbf{z}_i, \mathbf{z}) - \mathcal{K}(\mathbf{z}'_i, \mathbf{z})$, where $\mathbf{z}_i = (\bar{\mathbf{x}}_i, \bar{p}_i, \bar{s}_i)$ and $\mathbf{z}'_i = (\bar{\mathbf{x}}'_i, \bar{p}'_i, \bar{s}'_i)$. Since $\mathcal{K}$ is the sum of $\mathcal{K}_1, \mathcal{K}_2$, and $\mathcal{K}_3$ we get that, $\mathcal{K}(\mathbf{z}_i, \mathbf{z}) - \mathcal{K}(\mathbf{z}'_i, \mathbf{z}) = \sum_{j=1}^{3} \beta_j (\mathcal{K}_j(\mathbf{z}_i, \mathbf{z}) - \mathcal{K}_j(\mathbf{z}'_i, \mathbf{z}))$. We would like to note in passing that the term

$\mathcal{K}_1(\mathbf{z}_i,\mathbf{z}) - \mathcal{K}_1(\mathbf{z}'_i,\mathbf{z})$ can be rewritten in the following more compact form,

$$\sum_{p\in\mathcal{P}}\sum_{\tau\in T_p(\mathbf{z})}\sum_{t=1}^{|\bar{\mathbf{x}}_i|}\psi(t;\mathbf{z}_i,\mathbf{z}'_i)\text{‘}\exp\left(-\frac{\|\mathbf{x}_{i,t}-\mathbf{x}_\tau\|^2}{2\sigma^2}\right) \quad ,$$

where

$$\psi(t;\mathbf{z}_i,\mathbf{z}'_i) \;=\; \begin{cases} 1 & t\in T_p(\mathbf{z}_i)\ \wedge\ t\notin T_p(\mathbf{z}'_i) \\ -1 & t\notin T_p(\mathbf{z}_i)\ \wedge\ t\in T_p(\mathbf{z}'_i) \\ 0 & \text{otherwise} \end{cases}$$

In particular, for all frames $\mathbf{x}_{i,t}$ such that $\mathbf{z}_i$ and $\mathbf{z}'_i$ agree on the uttered phoneme, the value of $\psi(t;\mathbf{z}_i,\mathbf{z}'_i)$ is zero, which means that frame $\mathbf{x}_{i,t}$ does not effect the prediction.

The second kernel operator $\mathcal{K}_2$ is reminiscent of a phoneme duration model and is thus oblivious to the speech signal itself and merely examines the duration of each phoneme. Let $\mathcal{D}$ denote a set of predefined thresholds. For each $p\in\mathcal{P}$ and $d\in\mathcal{D}$ let $N_{p,d}(\mathbf{z})$ denote the number of times the phoneme p appeared in $\bar{p}$ while its duration was at least d, that is, $N_{p,d}(\mathbf{z}) = |\{k : p_k = p\ \wedge\ (s_{k+1}-s_k)\ge d\}|$. Using this notation, $\mathcal{K}_2$ is defined to be

$$\mathcal{K}_2(\mathbf{z},\mathbf{z}') \;=\; \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}} N_{p,d}(\mathbf{z})\,N_{p,d}(\mathbf{z}') \quad .$$

The last kernel operator $\mathcal{K}_3$ is reminiscent of a phoneme transition model. Let $A(p',p)$ be an estimated transition probability matrix from phoneme p' to phoneme p. Additionally, let Θ be a set of threshold values. For each $\theta\in\Theta$ let $N_\theta(\mathbf{z})$ be the number of times we switch from phoneme p_{k-1} to phoneme p_k such that $A(p_{k-1},p_k)$ is at least θ, that is, $N_\theta(\mathbf{z}) = |\{k : A(p_{k-1},p_k)\ge\theta\}|$. Using this notation, $\mathcal{K}_3$ is defined to be

$$\mathcal{K}_3(\mathbf{z},\mathbf{z}') \;=\; \sum_{\theta\in\Theta} N_\theta(\mathbf{z})\,N_\theta(\mathbf{z}') \quad .$$

We conclude this section with a brief discussion on the practical evaluation of the function f. Recall that calculating f requires solving the optimization problem $f(\bar{\mathbf{x}}) = \arg\max_{\bar{p}}\max_{\bar{s}}\sum_{j=1}^{m}\alpha_j\left(\mathcal{K}(\mathbf{z}_i,\mathbf{z}) - \mathcal{K}(\mathbf{z}'_i,\mathbf{z})\right)$. A direct search for the maximizer is not feasible since the number of possible phoneme sequences is exponential in the length of the sequence. Fortunately, the kernel operator we have presented is decomposable and thus the best phoneme sequence can be found in polynomial time using dynamic programming (similarly to the Viterbi procedure often implemented in HMMs [12]).

5 Experimental Results

To validate the effectiveness of the proposed approach we performed experiments with the TIMIT corpus. All the experiments described here have followed the same methodology. We divided the training portion of TIMIT (excluding the SA1 and SA2 utterances) into two disjoint parts containing 3600 and 96 utterances. The first part is used as a training set and the second part is used as a validation set. Mel-frequency cepstrum coefficients (MFCC) along with their first and the second derivatives were extracted from the speech waveform in a standard way along with cepstral mean subtraction (CMS). Leading and trailing silences from each utterance were removed. The TIMIT original phoneme set of 61 phonemes was mapped to a 39 phoneme set as proposed by [10]. Performance was evaluated over the TIMIT core test set by calculating the Levenshtein distance between the predicted phoneme sequence and the correct one.

We applied our method as discussed in Sec. 3 and Sec. 4 where $\sigma^2 = 6$, $C = 80$, $\boldsymbol{\beta} = \{1, 0.02, 0.02\}$, $\mathcal{D} = \{5, 10, 15, \ldots, 40\}$ and $\Theta = \{0.1, 0.2, \ldots, 0.9\}$. We compared the results of our method to the HMM approach, where each phoneme was represented by a simple left-to-right HMM of 5 emitting states with 40 diagonal Gaussians. These models were enrolled as follows: first the HMMs were

	Correct	Accuracy	Ins.	Del.	Sub.
Kernel-based	62.6	41.8	20.8	3.7	33.7
HMM	62.7	59.1	3.6	10.5	26.8

Table 1: Phoneme recognition results comparing our kernel-based discriminative algorithm versus HMM.

initialized using K-means, and then enrolled independently using EM. The second step, often called embedded training, re-enrolls all the models by relaxing the segmentation constraints using a forced alignment. Minimum values of the variances for each Gaussian were set to 20% of the global variance of the data. All HMM experiments were done using the *Torch* package [3]. All hyper-parameters including number of states, number of Gaussians per state, variance flooring factor, were tuned using the validation set. The overall results are given in Table 1. We report the number of insertions (Ins.), deletions (Del.) and substitutions (Sub.), as calculated by the Levenshtein distance. The Levenshtein distance is defined as the sum of insertions, deletions, and substitutions. Accuracy stands for 100% minus the Levenshtein distance and Correct stands for Accuracy plus insertions. As can be seen, the HMM method outperforms our method in terms of accuracy, mainly due to the high level of insertions of our method, suggesting that a better duration model should be explored. Nevertheless, we believe that the potential of our method is larger than the results reported and we discuss some possible improvements in the next section. Source code of our method can be found in `http://www.cs.huji.ac.il/~jkeshet/`.

6 Discussion

To date, the most successful phoneme sequence recognizers have been based on HMMs. In this report, we propose an alternative learning scheme for phoneme recognition which is based on discriminative supervised learning and Mercer kernels. The work presented in this report is part of an ongoing research trying to apply discriminative kernel methods to speech processing problems [8, 6, 9]. So far, the experimental results we obtained with our method for the task of phoneme recognition are still inferior to state-of-the-art results obtained by HMMs. However, while there has been extensive continuous effort on using HMMs for phoneme sequence recognition, our method is rather innovative and the choice of features and kernel operators is by no means comprehensive. We intend to utilize the full power of kernel methods for phoneme recognition by experimenting with additional features and kernels for our task.

Acknowledgements

Part of this research was done while Joseph Keshet was visiting IDIAP Research Institute. The authors are in debt to Johnny Mariéthoz for making the HMM experiments using the *Torch* package. The authors also would like to thank David Grangier for his many suggestions and comments. This research was supported by the European PASCAL Network of Excellence.

References

[1] N. Cesa-Bianchi, A. Conconi, and C.Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

[2] R. Chengalvarayan and L. Deng. Speech trajectory discrimination using the minimum classification error learning. *IEEE Trans. Speech and Audio Proc.*, 6(6):505–515, 1998.

[3] R. Collobert, S. Bengio, and J. Mariéthoz. Torch: a modular machine learning software library. IDIAP-RR 46, IDIAP, 2002.

[4] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive aggressive algorithms. *Journal of Machine Learning Research*, 7, Mar 2006.

[5] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[6] O. Dekel, J. Keshet, and Y. Singer. Online algorithm for hierarchical phoneme classification. In *Workshop on Multimodal Interaction and Related Machine Learning Algorithms; Lecture Notes in Computer Science*, pages 146–159. Springer-Verlag, 2004.

[7] V.V. Digalakis, M. Ostendorf, and J.R. Rohlicek. Fast algorithms for phone classification and recognition using segment-based models. *IEEE Trans. on Signal Processing*, 40:2885–2896, 1992.

[8] J. Keshet, D. Chazan, and B.-Z. Bobrovsky. Plosive spotting with margin classifiers. In *Proceedings of the Seventh European Conference on Speech Communication and Technology*, pages 1637–1640, 2001.

[9] J. Keshet, S. Shalev-Shwartz, Y. Singer, and D. Chazan. Phoneme alignment based on discriminative learning. In *Interspeech*, 2005.

[10] K.-F. Lee and H.-W. Hon. Speaker independent phone recognition using hidden markov models. *IEEE Trans. Acoustic, Speech and Signal Proc.*, 37(2):1641–1648, 1989.

[11] M. Ostendorf, V.V. Digalakis, and O.A. Kimball. From hmm's to segment models: A unified view to stochastic modeling for speech recognition. *IEEE Trans. Speech and Audio Proc.*, 4(5):360–378, 1996.

[12] L. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[13] J. Salomon, S. King, and M. Osborne. Framewise phone classification using support vector machines. In *Proceedings of the Seventh International Conference on Spoken Language Processing*, pages 2645–2648, 2002.

[14] S. Shalev-Shwartz, J. Keshet, and Y. Singer. Learning to align polyphonic music. In *Proceedings of the 5th International Conference on Music Information Retrieval*, 2004.

[15] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Advances in Neural Information Processing Systems 17*, 2003.

[16] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the Twenty-First International Conference on Machine Learning*, 2004.

[17] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[18] S. Young. A review of large-vocabulary continuous speech recognition. *IEEE Signal Processing Mag.*, pages 45–57, September 1996.