PAC-BAYESIAN APPROACH FOR MINIMIZATION OF PHONEME ERROR RATE

Joseph Keshet *David McAllester* *Tamir Hazan*

Toyota Technological Institute at Chicago
Chicago, IL

ABSTRACT

We describe a new approach for phoneme recognition which aims at minimizing the phoneme error rate. Building on structured prediction techniques, we formulate the phoneme recognizer as a linear combination of feature functions. We state a PAC-Bayesian generalization bound, which gives an upper-bound on the expected phoneme error rate in terms of the empirical phoneme error rate. Our algorithm is derived by finding the gradient of the PAC-Bayesian bound and minimizing it by stochastic gradient descent. The resulting algorithm is iterative and easy to implement. Experiments on the TIMIT corpus show that our method achieves the lowest phoneme error rate compared to other discriminative and generative models with the same expressive power.

Index Terms— PAC-Bayesian theorem, phoneme recognition, structured prediction, discriminative training, kernels

1. INTRODUCTION

Phoneme recognition is the task of predicting the phonetic content of a given speech signal. The performance of this task is measured by the phoneme error rate, which is defined as the minimum number of edits needed to transform the predicted phoneme sequence into the correct one. Unlike previous approaches, in this work we propose a new algorithm that aims at minimizing the phoneme error rate. Out algorithm is derived by finding the gradient of a generalizing bound, which gives an upper-bound on the expected phoneme error rate in terms of the empirical phoneme error rate. This gradient is used with a stochastic gradient descent approach to get an efficient iterative phoneme recognition algorithm.

Most previous work on phoneme recognition has focused on Hidden Markov Models (HMMs). Classically these models are trained to estimate the joint likelihood of the acoustic signal and the underlying phonetic representation [1], and do not aim at minimizing the phoneme error rate. Over the years, several discriminative training criteria for HMMs have been proposed, including Maximum Mutual Information (MMI) [2], Minimum Classification Error (MCE) [3], and Large Margin (LM) [4], none of which minimizes the phoneme error rate directly, with the exception of Minimum Phone Error (MPE) training criterion [5], which tries to minimize the smoothed phone error rate, but did not present results on the standard TIMIT phoneme recognition benchmark. Generative HMM-based approaches have several drawbacks: they do not faithfully reflect the underlying structure of speech signals as they assume conditional independence of observations given the state sequence [6] and they require uncorrelated acoustic features [7].

Our method builds upon recent advances in discriminative supervised learning for structured labels, such as the structured Support Vector Machines (SVMs) [8, 9] and Conditional Random Fields (CRFs) [10]. Structured SVMs generalize binary SVMs to deal with structured labels (such as sequences) with any cost function. They minimize a hinge surrogate to the cost with no guarantee for the actual cost on unseen data. CRFs minimize the regularized log loss on the training set and is independent on the cost whatsoever.

In this paper we present a new algorithm which aims at minimizing the phoneme error rate. The algorithm is derived by minimizing a PAC-Bayesian generalization bound using stochastic gradient descent. Despite minimizing a non-convex function, experiments with the TIMIT corpus show that our approach achieves the lowest phoneme error rate compared to other discriminative and generative models with the same expressive power.

The paper is organized as follows. In Section 2 we formally introduce the phoneme recognition problem. Next, in Section 3 we describe the PAC-Bayesian framework. The derivation of the algorithm is presented in Section 4. We conclude the paper with experimental results on the TIMIT corpus in Section 5.

2. PROBLEM SETTING

In the problem of phoneme recognition we are given a spoken utterance and the goal is to predict its phonetic content. The spoken utterance is represented as a sequence of acoustic feature-vectors $\overline{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$, whose components $\mathbf{x}_t$ are vectors in $\mathrm{X} \subset \mathbb{R}^d$. Each utterance corresponds to a sequence of phoneme symbols. Formally, we denote each phoneme symbol by $p \in \mathrm{P}$, where P is a set of phoneme symbols, and we denote the sequence of phoneme symbols by $\overline{p} = (p_1, \ldots, p_K)$. Furthermore, we denote by $s_k \in \mathbb{N}$ the start time of phoneme p_k (in frame units) and we denote by $\overline{s} = (s_1, \ldots, s_K)$ the sequence of all phoneme start-times. The number of phonemes K and the number of frames T can be different for different inputs, although typically we have T significantly larger than K. Our goal is to learn a function f that predicts the correct phoneme sequence given an acoustic sequence. The phonetic decoder f is a function from the set of finite-length sequences over the domain of the acoustic features X^* to the set of finite-length sequences over the domain of phoneme symbols, P^*.

Denote by $L(\overline{p}, f(\overline{\mathbf{x}}))$ the loss (or the cost) of predicting the phoneme sequence $f(\overline{\mathbf{x}})$ where the correct sequence is $\overline{p}$. Formally, $L : \mathrm{P}^* \times \mathrm{P}^* \to \mathbb{R}_+$ is a function that gets two phoneme sequences (not necessarily of the same length) and returns a positive number which is the cost of predicting the second sequence where the desired sequence is the first. The loss of a phonetic decoder is usually the *phoneme error rate* (also called *Levenshtein distance* or *edit distance*). This loss measures the minimum number of substitutions, insertions and deletions needed to transform the predicted phoneme sequence into the correct phoneme sequence normalized by the number of phonemes in the correct sequence.

Following structured prediction scheme [8, 9, 11], our phonetic decoder utilizes a fixed mapping $\boldsymbol{\phi} : \mathrm{X}^* \times \mathrm{P}^* \times \mathbb{N}^* \to \mathbb{R}^n$ from the input acoustic representation, a candidate phoneme sequence and a

candidate start-time sequence to feature vectors of length n. This mapping is needed to have features of the same length. Intuitively, the vector-valued feature function $\boldsymbol{\phi}(\overline{\mathbf{x}}, \overline{p}, \overline{s})$ is a set of n confidences for the candidate phoneme sequence and start-time sequence. Our phonetic decoder is of the following form

$$f_{\mathbf{w}}(\overline{\mathbf{x}}) = \arg\max_{\overline{p}} \left(\max_{\overline{s}} \ \mathbf{w}^{\top} \boldsymbol{\phi}(\overline{\mathbf{x}}, \overline{p}, \overline{s}) \right), \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^n$ is a weight vector. In this paper we describe a discriminative supervised learning approach for learning the weight vector $\mathbf{w}$ from a training set of m examples, $\{(\overline{\mathbf{x}}_i, \overline{p}_i)\}_{i=1}^m$. We assume that the examples are drawn from a fixed but unknown distribution $\mathcal{D}$ over the domain of the examples, $\mathrm{X}^* \times \mathrm{P}^*$. The ultimate objective is to set the weight vector $\mathbf{w}$ so as to minimize the expected phone error rate between the desired output $\overline{p}$ and the predicted output $f_{\mathbf{w}}(\overline{\mathbf{x}})$, that is

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \ \mathbb{E}_{(\overline{\mathbf{x}}, \overline{p}) \sim \mathcal{D}} \left[L(\overline{p}, f_{\mathbf{w}}(\overline{\mathbf{x}})) \right]. \tag{2}$$

To do so, we assume that the examples are sampled i.i.d. from $\mathcal{D}$. Note, however, that we cannot evaluate Eq. (2), since $\mathcal{D}$ is unknown. Instead, we use the empirical loss computed over the training set, and a regularization term is added in order to prevent overfitting when there is large number of features.

3. THE PAC-BAYESIAN FRAMEWORK

For any weight vector $\mathbf{w}$ and a pair $(\overline{\mathbf{x}}, \overline{p})$ we defined the generalized probit surrogate loss as $\hat{L}_p(\mathbf{w}, \overline{\mathbf{x}}, \overline{p})$ as follows where the expectation is over drawing the "noise" vector $\boldsymbol{\epsilon}$ from a unit-variance isotropic Gaussian.

$$L_p(\mathbf{w}, \overline{\mathbf{x}}, \overline{p}) = \mathbb{E}_{\boldsymbol{\epsilon}}[L(\overline{p}, f_{\mathbf{w}+\boldsymbol{\epsilon}}(\overline{\mathbf{x}}))]$$

We define the true probit loss $L_p(\mathbf{w})$ as follows.

$$L_p(\mathbf{w}) = \mathbb{E}_{(\overline{\mathbf{x}}, \overline{p}) \sim \mathcal{D}} \left[L_p(\mathbf{w}, \overline{\mathbf{x}}, \overline{p}) \right].$$

and similarly, we define the empirical probit loss $\widehat{L}_p(\mathbf{w})$ as

$$\widehat{L}_p(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} L_p(\mathbf{w}, \overline{\mathbf{x}}_i, \overline{p}_i).$$

Note that the probit loss corresponds to the phoneme error rate of the stochastic decoder $f_{\mathbf{w}+\boldsymbol{\epsilon}}(\overline{\mathbf{x}})$ where $\boldsymbol{\epsilon}$ is Gaussian noise vector from a unit-variance isotropic Gaussian. Hence any theoretical guarantee on the probit loss $L_p(\mathbf{w})$ establishes that same guarantee for the phoneme error rate of an actual (stochastic) decoder. Also note that the prediction $f_{\mathbf{w}}(\overline{\mathbf{x}})$ depends only on the direction of $\mathbf{w}$. As we scale $\mathbf{w}$ to be arbitrarily large the Gaussian noise has a vanishing effect on the prediction and the probit loss equals the phoneme error rate.

We state now a nonstandard version of PAC-Bayesian theorem. In Appendix A we show that this nonstandard statement is equivalent to a standard PAC-Bayesian bound for linear decoders under a Gaussian prior and posterior [12].

Theorem 1. *With probability of at least $1 - \delta$ over the draw of the training set the following holds simultaneously for all vectors $\mathbf{w}$*

$$L_p(\mathbf{w}) \leq \inf_{\lambda > 0} \frac{1}{1 - \frac{1}{2\lambda}} \left[\widehat{L}_p(\mathbf{w}) + \frac{\lambda}{2(m-1)} \|\mathbf{w}\|^2 + \frac{\lambda}{m-1} \ln \frac{m}{\delta} \right].$$

4. PAC-BAYESIAN STOCHASTIC GRADIENT DESCENT

In this section we describe an algorithm which minimizes the right-hand side of the PAC-Bayesian bound for a fixed value of λ. The minimization is carried out by stochastic gradient descent.

We start by finding the gradient of the right hand side of the PAC-Bayesian bound. Denoting $\lambda' = \lambda/(m-1)$, we have

$$\begin{aligned}
&\nabla_{\mathbf{w}} \left[\widehat{L}_p(\mathbf{w}) + \frac{\lambda}{2(m-1)} \|\mathbf{w}\|^2 + \frac{\lambda}{m-1} \ln \frac{m}{\delta} \right] = \\
&= \nabla_{\mathbf{w}} \left[\frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{\boldsymbol{\epsilon}} \left[L(\overline{p}_i, f_{\mathbf{w}+\boldsymbol{\epsilon}}(\overline{\mathbf{x}}_i)) \right] + \frac{\lambda'}{2} \|\mathbf{w}\|^2 \right] \\
&= \nabla_{\mathbf{w}} \left[\frac{1}{m} \sum_{i=1}^{m} \int (2\pi)^{-d/2} e^{-\frac{1}{2}\|\boldsymbol{\epsilon}\|^2} L(\overline{p}_i, f_{\mathbf{w}+\boldsymbol{\epsilon}}(\overline{\mathbf{x}}_i)) d\boldsymbol{\epsilon} + \frac{\lambda'}{2} \|\mathbf{w}\|^2 \right] \\
&= \frac{1}{m} \sum_{i=1}^{m} \int \boldsymbol{\epsilon} \cdot (2\pi)^{-d/2} e^{-\frac{1}{2}\|\boldsymbol{\epsilon}\|^2} L(\overline{p}_i, f_{\mathbf{w}+\boldsymbol{\epsilon}}(\overline{\mathbf{x}}_i)) d\boldsymbol{\epsilon} + \lambda' \mathbf{w} \\
&= \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{\boldsymbol{\epsilon}} \left[\boldsymbol{\epsilon} \, L(\overline{p}_i, f_{\mathbf{w}+\boldsymbol{\epsilon}}(\overline{\mathbf{x}}_i)) \right] + \lambda' \mathbf{w}. \qquad (3)
\end{aligned}$$

Stochastic (or "on-line") gradient descent learning takes place in rounds. The algorithm starts at a point $\mathbf{w}^0$. At each round the algorithm moves from $\mathbf{w}^t$ to $\mathbf{w}^{t+1}$ by minimizing along the line extending from $\mathbf{w}^t$ in the direction of the local downhill gradient. The true gradient is approximated by the gradient at a single example, $(\overline{\mathbf{x}}_i, \overline{p}_i)$, and resulting the following update rule

$$\mathbf{w}^{t+1} = (1 - \lambda' \eta^t) \mathbf{w}^t - \eta^t \mathbb{E}_{\boldsymbol{\epsilon}} \left[\boldsymbol{\epsilon} L(\overline{p}_i, f_{\mathbf{w}+\boldsymbol{\epsilon}}(\overline{\mathbf{x}}_i)) \right] \tag{4}$$

where η^t is the learning rate. We compute the expectation by sampling. When $\mathbf{w}$ has a large norm the sampling will not be effective, since with high probability any sample of $\boldsymbol{\epsilon}$ drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ is orthogonal to the gradient. We solve this issue by *importance sampling*, and steer the direction of the sampled vector $\boldsymbol{\epsilon}$ to the direction of the gradient. Formally, let $\hat{\mathbf{g}}_i$ be a unit vector in the direction of the gradient:

$$\hat{\mathbf{g}}_i = \frac{\boldsymbol{\phi}(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{s}_i) - \boldsymbol{\phi}(\overline{\mathbf{x}}_i, \overline{p}', \overline{s}')}{\|\boldsymbol{\phi}(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{s}_i) - \boldsymbol{\phi}(\overline{\mathbf{x}}_i, \overline{p}', \overline{s}')\|}, \tag{5}$$

with $(\overline{p}', \overline{s}')$ the phoneme sequence and its start-time sequence found by the weight vector $\mathbf{w}^t$ using Eq. (1), and $(\overline{p}_i, \overline{s}_i)$ are the correct phoneme sequence and start-time sequence. We note in passing that for the importance sampling and for the initialization we use the correct start-time sequence $\overline{s}_i$. But the the sequences $\overline{s}_i$ are not used in write hand side of Theorem 1. In order to perform the importance sampling we define a Gaussian distribution with mean $\hat{\mathbf{g}}_i$ and the identity covariance matrix, and normalize the expectation by $\mathcal{N}(\mathbf{0}, \mathbf{I}) / \mathcal{N}(\hat{\mathbf{g}}_i, \mathbf{I}) = e^{-\|\boldsymbol{\epsilon}\|^2/2} / e^{-\|\boldsymbol{\epsilon} - \hat{\mathbf{g}}_i\|^2/2} = k e^{-2\boldsymbol{\epsilon} \cdot \hat{\mathbf{g}}_i}$, where k is a constant that does not depend on $\boldsymbol{\epsilon}$, and can be absorbed into the learning parameter. Eq. (4) becomes

$$\mathbf{w}^{t+1} = (1 - \lambda' \eta^t) \mathbf{w}^t - \eta^t \mathbb{E}_{\boldsymbol{\epsilon}} \left[\boldsymbol{\epsilon} L(\overline{p}_i, f_{\mathbf{w}+\hat{\boldsymbol{\epsilon}}}(\overline{\mathbf{x}}_i)) e^{-2\boldsymbol{\epsilon} \cdot \hat{\mathbf{g}}_i} \right], \tag{6}$$

where the expectation is now with respect to $\boldsymbol{\epsilon} \sim \mathcal{N}(\hat{\mathbf{g}}_i, \mathbf{I})$. For efficiency, we can assume sampling of both $\boldsymbol{\epsilon}$ and $-\boldsymbol{\epsilon}$ simultaneously and we consider only those directions $\boldsymbol{\epsilon}$ that change the expected loss, namely

$$\begin{aligned}
\mathbf{w}^{t+1} &= (1 - \lambda' \eta^t) \mathbf{w}^t \\
&\quad - \eta^t \mathbb{E}_{\boldsymbol{\epsilon}} \left[\left(L(\overline{p}_i, f_{\mathbf{w}+\boldsymbol{\epsilon}}(\overline{\mathbf{x}}_i)) - L(\overline{p}_i, f_{\mathbf{w}-\boldsymbol{\epsilon}}(\overline{\mathbf{x}}_i)) \right) e^{-2\boldsymbol{\epsilon} \cdot \hat{\mathbf{g}}_i} \right].
\end{aligned}$$

The overall algorithm is described in Figure 1.

INPUT: training set $\{(\overline{\mathbf{x}}_i, \overline{p}_i)\}_{i=1}^m$;
parameters: λ, η_0, and J

INITIALIZATION: $\mathbf{w}^0$

FOR $t = 1, \ldots, T$

Pick example $(\overline{\mathbf{x}}_i, \overline{p}_i) \in \mathrm{S}$

Predict $(\overline{p}', \overline{s}') = \arg\max_{(\overline{p},\overline{s})} \mathbf{w}^t \cdot \boldsymbol{\phi}(\overline{\mathbf{x}}_i, \overline{p}, \overline{s})$

Set $\hat{\mathbf{g}}_i = \dfrac{\boldsymbol{\phi}(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{s}_i) - \boldsymbol{\phi}(\overline{\mathbf{x}}_i, \overline{p}', \overline{s}')}{\|\boldsymbol{\phi}(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{s}_i) - \boldsymbol{\phi}(\overline{\mathbf{x}}_i, \overline{p}', \overline{s}')\|}$

Draw $\boldsymbol{\epsilon}_j$ from $\mathcal{N}(\hat{\mathbf{g}}_i, \mathbf{I})$, $1 \le j \le J$

Set $\widehat{\mathrm{E}}_i^t = \frac{1}{J} \sum_{j=1}^J \boldsymbol{\epsilon}_j \left[L(\overline{p}_i, f_{\mathbf{w}^t + \boldsymbol{\epsilon}_j}(\overline{\mathbf{x}}_i)) - L(\overline{p}_i, f_{\mathbf{w}^t - \boldsymbol{\epsilon}_j}(\overline{\mathbf{x}}_i)) \right] e^{-2\boldsymbol{\epsilon}_j \cdot \hat{\mathbf{g}}_i}$

Set $\mathbf{w}^{t+1} = (1 - \frac{\lambda}{m-1} \frac{\eta_0}{t}) \mathbf{w}^t - \frac{\eta_0}{t} \widehat{\mathrm{E}}_i^t$

OUTPUT: $\mathbf{w}^{T+1}$

Fig. 1. The PAC-Bayesian Theorem Minimization algorithm. The initialization and importance sampling use the sequences $\overline{s}_i$ that are available in the training data.

5. EXPERIMENTAL RESULTS

We evaluated the proposed method on the TIMIT acoustic-phonetic continuous speech corpus [13]. The training set contains 462 speakers and 3696 utterances. We used the core test set of 24 speakers and 192 utterances and a development set of 50 speakers and 400 utterances as defined in [4] for tuning the parameters. Following the common practice [14], we mappped the 61 TIMIT phonemes into 48 phonemes for training, and further collapsed from 48 phonemes to 39 phonemes for evaluation. We extracted standard 12 MFCC features and log energy with their deltas and double deltas to form 39-dimensional acoustic feature vectors. The window size and the frame size were 25 msec and 10 msec, respectively.

Similarly to the output and transition probabilities in HMMs, our implementation has two sets of feature functions. The first feature function set captures the confidence of a phoneme based on the acoustic. For each phoneme $p \in \mathcal{P}$, we define the feature map as the sum over all acoustic features correspond to phoneme p,

$$\phi_p^I(\overline{\mathbf{x}}, \overline{p}, \overline{s}) = \sum_{t:p_t=p} \widetilde{\boldsymbol{\psi}}_\sigma(\mathbf{x}_t), \ \ \forall p \in \mathrm{P},$$

where p_t is the phoneme at frame t. The mapping $\widetilde{\psi}_\sigma$ is an approximation to the RBF kernel of order 3 with parameter σ as described in Appendix B. Below we report results with a context window of 1 frame and a context window of 9 frames, i.e., $\hat{\psi}_\sigma$ is a concatenation of 9 frames, and in each frame the acoustic feature vectors are mapped using the approximated RBF kernel.

The second set of feature functions captures both the duration of each phoneme and the transition between phonemes. For each pair of phonemes $p, q \in \mathcal{P}$ we define the feature map as a sum over all transitions between phoneme p and q:

$$\phi_{p,q}^{II}(\overline{\mathbf{x}}, \overline{p}, \overline{s}) = \sum_{t:p_t=p, p_{t+1}=q} \beta, \ \ \forall p, q \in \mathrm{P},$$

where β is a parameter that scales the second feature set.

We applied the algorithm as discussed in Section 4 where the parameters $\sigma^2 = 19$, $\lambda = 0.05$, $\beta = 0.4$, and $J = 1000$ were found on the development set. The initial weight vector $\mathbf{w}^0$ was set to averaged weight vector of the Passive-Aggressive (PA) algorithm [15], which was trained with the same set of parameters and with 100 epochs as described in [16].

Table 1. Reported results on TIMIT core test set.

Method	Frame error rate	Phoneme error rate
HMM [20]	39.3	42.0
HMM [11]	35.1	40.9
KSBSC [11]	-	45.1
PA [16]	30.0	33.4
DROP [16]	29.2	31.1
PAC-Bayes 1-frame	**27.7**	**30.2**
Online LM-HMM [20]	25.0	30.2
Batch LM-HMM [4]	-	28.2
CRFs (9-frames MLP) [21]	-	29.3
PAC-Bayes 9-frames	**26.5**	**28.6**

Table 1 summarizes the results and compare the performance of the proposed algorithm to other algorithms for phoneme recognition. Although the algorithm aims at minimizing the phoneme error rate, we also report the frame error rate, which is the fraction of misclassified frames. A common practice is to split each phoneme segment into three (or more) states. Using such a technique usually improves performance (see for example [17, 18, 19]). Here we report results on approaches which treat the phoneme as a whole, and defer the issues of splitting into states in our algorithm for future work. In the upper part of the table (above the line), we report results on approaches which make use of context window of 1 frame. The first two rows are two HMM systems taken from [11] and [20] with a single state corresponding to our setting. KSBSC [11] is kernel-based recognizer trained with PA algorithm. PA and DROP [16] is are online algorithm, uses the same setup and feature functions described here. Online LM-HMM [20] and Batch LM-HMM [4] are algorithms for large margin training of continuous density HMMs. Below the line, at the bottom part of the table, we report the result with a context of 9 frames. CRFs [21] is based on the computation of local posteriors with MLPs, which was trained on a context of 9 frames. We can see the our algorithm outperform all algorithms but the large margin HMMs. The difference between our algorithm and the LM-HMM algorithm might be in the reacher expressive power of the latter. Using context of 9 frames the results of our algorithm comparable to LM-HMM.

6. REFERENCES

[1] L. Rabiner and B.H. Juang, *Fundamentals of speech recognition*, Prentice-Hall, Inc., 1993.

[2] L.R. Bahl, P.F. Brown, P.V. de Souza, and R.L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. ICASSP*, 1986.

[3] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 5, no. 3, pp. 257–265, 1997.

[4] F. Sha and L. K. Saul, "Comparison of large margin training to other discriminative methods for phonetic recognition by hidden Markov models," in *Proc. ICASSP*, 2007.

[5] D. Povey and P.C. Woodland, "Minimum phone error and I-Smoothing for improved discriminative training," in *Proc. ICASSP*, 2002.

[6] M. Ostendorf, V.V. Digalakis, and O.A. Kimball, "From HMM's to segment models: A unified view of stochastic modeling for speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 5, 1996.

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

[8] B. Taskar, C. Guestrin, and D. Koller, "Max-margin markov networks," in *Proc. NIPS 17*, 2003.

[9] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *Journal of Machine Learning Research*, vol. 6, pp. 1453–1484, 2005.

[10] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. ICML*, 2001, pp. 282–289.

[11] J. Keshet, S. Shalev-Shwartz, S. Bengio, Y. Singer, and D. Chazan, "Discriminative kernel-based phoneme sequence recognition," in *Interspeech*, 2006.

[12] D. McAllester, "Simplified PAC-Bayesian margin bounds," in *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory*, 2003.

[13] L. Lamel, R. Kassel, and S. Seneff, "Speech database development: Design an analysis of the acoustic-phonetic corpus," in *DARPA Speech Recognition Workshop*, 1986.

[14] K.-F. Lee and H.-W. Hon, "Speaker independent phone recognition using hidden markov models," *IEEE Trans. Acoustic, Speech and Signal Proc.*, vol. 37, no. 2, pp. 1641–1648, 1989.

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

[16] K. Crammer, "Efficient online learning with individual learning-rates for phoneme sequence recognition," in *Proc. ICASSP*, 2010.

[17] A. Mohamed and G.E. Hinton, "Phone recognition using restricted boltzmann machines," in *Proc. ICASSP*, 2010.

[18] Y.-H. Sung and D. Jurafsky, "Hidden conditional random fields for phone recognition," in *Proc. ASRU*, 2010.

[19] P. Schwartz, P. Matejka, and J. Cernocky, "Hierarchical structures of neural networks for phoneme recognition," in *Proc. ICASSP*, 2006.

[20] C.-C. Cheng, F. Sha, and L. K. Saul, "A fast online algorithm for large margin training of continuous-density hidden Markov models," in *Interspeech*, 2009.

[21] J. Morris and E. Fosler-Lussier, "Conditional random fields for integrating local discriminative classifiers," *IEEE Trans. on Acoustics, Speech, and Language Processing*, vol. 16, no. 3, pp. 617–628, 2008.

A. PROOF SKETCH OF THEOREM 1

The statement of Theorem 1 is nonstandard. Here we show that this nonstandard statement is equivalent to a standard PAC-Bayesian bound. The following is a standard variant of the PAC-Bayesian theorem for linear decoders under a Gaussian prior and posterior [12].

$$L_p(\mathbf{w}) \leq \sup\left\{ L_p : \frac{(\widehat{L}_p(\mathbf{w}) - L_p)^2}{2L_p} \leq \frac{\frac{1}{2}||\mathbf{w}||^2 + \ln\frac{m}{\delta}}{m-1} \right\}$$

Theorem 1 now follows from the following observation which uses the observation that for $x, y \geq 0$ we have $\sqrt{xy} = \inf_{\lambda>0} \frac{1}{2}(\frac{x}{\lambda} + \lambda y)$.

$$\begin{aligned}
& \sup\left\{ L_p : \frac{(L_p - \widehat{L}_p)^2}{2L_p} \leq c \right\} \\
= & \sup\left\{ L_p : L_p - \widehat{L}_p \leq \sqrt{2L_p c} \right\} \\
= & \sup\left\{ L_p : \forall\, \lambda > 0\ L_p - \widehat{L}_p \leq \frac{L_p}{2\lambda} + \lambda c \right\} \\
= & \sup\left\{ L_p : \forall\, \lambda > 0\ L_p \leq \left(\frac{1}{1-\frac{1}{2\lambda}}\right)\left(\widehat{L_p} + \lambda c\right) \right\} \\
= & \inf_{\lambda>0}\left(\frac{1}{1-\frac{1}{2\lambda}}\right)\left(\widehat{L}_p + \lambda c\right).
\end{aligned}$$

B. RBF KERNEL APPROXIMATION

Let us start with the Gaussian Radial Basis Function (RBF) kernel definition

$$\begin{aligned}
k(\mathbf{x}, \mathbf{z}) &= e^{-\|\mathbf{x}-\mathbf{z}\|^2/2\sigma^2} \\
&= e^{-\|\mathbf{x}\|^2/2\sigma^2} e^{-\|\mathbf{z}\|^2/2\sigma^2} e^{<\mathbf{x},\mathbf{z}>/\sigma^2}.
\end{aligned}$$

Since the last term $e^{<\mathbf{x},\mathbf{z}>/\sigma^2}$ is a real number, it can be expanded using Taylor Expansion as follows:

$$\begin{aligned}
e^{<\mathbf{x},\mathbf{z}>/\sigma^2} &= \sum_{n=1}^{N} \frac{1}{n!}\left(\frac{<\mathbf{x},\mathbf{z}>}{\sigma^2}\right)^n \\
&= \left(1 + \frac{\mathbf{x}}{\sigma} + \frac{\mathbf{x}'}{\sqrt{2}\sigma^2} + \ldots\right)^\top \cdot \left(1 + \frac{\mathbf{z}}{\sigma} + \frac{\mathbf{z}'}{\sqrt{2}\sigma^2} + \ldots\right)^\top,
\end{aligned}$$

where $\mathbf{x}'$ (and similarly $\mathbf{z}'$) is defined as

$$\mathbf{x}' = \mathbf{x}_1^2 + \mathbf{x}_2^2 + \ldots + \mathbf{x}_d^2 + \sqrt{2}\mathbf{x}_1\mathbf{x}_2 + \ldots + \sqrt{2}\mathbf{x}_{d-1}\mathbf{x}_d.$$

Overall, the kernel can be written as $k(\mathbf{x}, \mathbf{z}) = \boldsymbol{\psi}(\mathbf{x}) \cdot \boldsymbol{\psi}(\mathbf{z})$, where

$$\boldsymbol{\psi}_\sigma(\mathbf{x}) = e^{-\|\mathbf{x}\|^2/2\sigma^2}\left(1 + \frac{\mathbf{x}}{\sigma} + \frac{\mathbf{x}'}{\sqrt{2}\sigma^2} + \ldots\right)^\top.$$

Now, instead of using the sum we can use the Taylor expansion

$$f(\mathbf{x}) = \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) = \mathbf{w} \cdot \boldsymbol{\psi}_\sigma(\mathbf{x}). \tag{7}$$

Using only several Taylor Expansion's terms we have an approximation to the Gaussian kernel

$$\hat{f}(\mathbf{x}) = \mathbf{w} \cdot \widetilde{\boldsymbol{\psi}}_\sigma(\mathbf{x}).$$

The length of the approximation $\widetilde{\boldsymbol{\psi}}_\sigma$ depends on the size of the input vector $\mathbf{x}$ and the size of the approximation. Recall that $\mathbf{x} \in \mathbb{R}^d$, then the zeroth approximation is of length one, the first approximation is of length d, the second approximation is of length $d + d(d-1)/2$, and in general the length of the p-th approximation is of order d^p. In our experiments we used approximation of order 3, which give approximation error (compared to the true RBF kernel) of less than 0.0001.