AN EVALUATION OF POSTERIOR MODELING TECHNIQUES FOR PHONETIC RECOGNITION

Rohit Prabhavalkar[1*] *Tara N. Sainath*[2] *David Nahamoo*[2] *Bhuvana Ramabhadran*[2] *Dimitri Kanevsky*[2]

[1] Department of CSE, The Ohio State University, Columbus, OH 43210, U.S.A
[2] IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, U.S.A
prabhava@cse.ohio-state.edu {tsainath, nahamoo, bhuvana, kanevsky}@us.ibm.com

ABSTRACT

Several methods have been proposed recently for modeling posterior representations derived from local classifiers [1, 2]. In recent work, Sainath et al. have proposed the use of a tied-mixture-based posterior modeling approach [3] to enhance exemplar-based posterior representations for phone recognition tasks. In this work, we conduct a detailed evaluation to determine the effectiveness of this technique on three representative posterior systems. In addition, we propose and evaluate an alternative discriminative formulation of the posterior modeling objective function that seeks to minimize frame-level errors. In experimental evaluations on the TIMIT corpus, we find that posterior modeling results in relative phone error rate (PER) reductions of between 1.1–5.5 % across the systems tested. In fact, using S_{pif}–NN [4, 3] posteriors, we are able to achieve a PER of 18.5; to the best of our knowledge, this is the best result reported in the literature to date.

Index Terms— posterior modeling, TIMIT, phone recognition, tied-mixture smoothing

1. INTRODUCTION

Local posterior estimates of phone class probabilities, derived using neural networks (NNs), for example, have long been used for phone and word recognition tasks in ASR. For instance, such posterior representations have been used, after suitable transformations, to estimate HMM output probabilities [5] in the 'hybrid' approach or as a replacement for standard MFCCs or PLPs in the 'tandem' approach [6]. Although these approaches can, in principle, be used to integrate local posterior probabilities generated from arbitrary underlying classifiers, better classification performance does not always translate into better recognition performance: It has been empirically observed with some classifiers, for example posteriors derived from conditional random fields (CRFs) [7] and exemplar-based methods [4], that the entropy of the resulting posterior distributions is extremely low. As a result, when the system prediction is incorrect, the probability of the incorrect class is boosted relative to the correct class which leads to errors in subsequent processing.

There have been a number of recent proposals, that we collectively refer to as posterior modeling, that have attempted to address the problem of posterior sharpness (low entropy of resulting posterior distributions). Aradilla [1] has proposed using Kullback-Leibler divergences between state distributions learned for phone states and posterior vectors corresponding to speech frames and applied these models for word recognition tasks. In [2], the authors propose modeling the output distribution of posterior vectors in an HMM using a Dirichlet mixture model, thus avoiding the need for 'tandem' processing. In [8], the authors consider techniques for reducing posterior sharpness in CRF-derived posterior representations by considering transformation of the CRF model parameters.

In this work, we conduct a detailed evaluation of the tied-mixture posterior modeling approach proposed in [3]. Our goal in this work is to determine whether these techniques can be effective at improving performance on posterior representations obtained from various posterior systems. For this purpose, we consider three representative posterior systems: posteriors generated from an exemplar-based approach (S_{pif}–NN [3, 4]), a deep belief network (DBN) trained on discriminative acoustic features and a state-of-the-art GMM-HMM system. We also consider a modification of the maximum-likelihood-based formulation of the objective function explored in [3], replacing it with a discriminative objective function that directly attempts to minimize frame-level errors. In experimental evaluations, we find that posterior modeling results in relative PER reductions of between 1.1–5.5%.

2. POSTERIOR MODELING USING TIED MIXTURE SMOOTHING

Posterior modeling was proposed recently [3] as a means for reducing posterior sharpness and was demonstrated to improve performance on TIMIT [9] phone recognition. Although the technique was introduced in the context of

*This work was produced as part of a summer internship at IBM research.

S_{pif}–NN (NN posteriors trained on S_{pif} features [4]), it is extremely general and can be utilized for modeling arbitrary posterior representations. We begin with a detailed review of the technique in this section.

Given a speech utterance $\overline{\mathbf{x}}$, let the acoustic vector corresponding to frame t be denoted as, $\mathbf{x}_t$. We assume the existence of an underlying classifier that produces posterior estimates of phone state probabilities, which we denote as $p(q = k|\mathbf{x}_t)$, where $k \in \mathcal{Q}$ represents a particular phone state. For example, the underlying posterior representation might correspond to a deep belief network (DBN) trained to predict context-dependent phone state probabilities. These posterior representation can be converted into scaled likelihoods using Bayes' rule (denoted by $a_t(k; \overline{\mathbf{x}})$), by dividing them by prior class probabilities, $p(q = k)$:

$$a_t(k; \overline{\mathbf{x}}) = \frac{p(q = k|\mathbf{x}_t)}{p(q = k)} \tag{1}$$

When underlying classifiers are trained to optimize frame error rates, the resultant posterior distributions often display very low entropy. As a result, when the classification decision is incorrect, the corresponding posterior distribution and hence the likelihoods in Equation 1 for the incorrect class are over-emphasized; this in turn degrades performance when these posteriors are utilized for phone or word recognition. This can be mitigated by using a tied-mixture smoothing technique [10] that models individual class likelihoods as mixtures of likelihoods from all classes. Denoting the mixing coefficients by $b(l, k)$, for states $l, k \in \mathcal{Q}$, we represent the smoothed class likelihoods, denoted as $c_t(l; \overline{\mathbf{x}})$, as

$$c_t(l; \overline{\mathbf{x}}) = \sum_{k \in \mathcal{Q}} b(l, k) a_t(k; \overline{\mathbf{x}}) \tag{2}$$

$$\text{where, } \forall l, k \in \mathcal{Q} : b(l, k) \geq 0 \text{ and } \forall l \in \mathcal{Q} : \sum_{k \in \mathcal{Q}} b(l, k) = 1 \tag{3}$$

Thus, in summary, the smoothed models are parameterized by a set of $|\mathcal{Q}|^2$ mixing coefficients that are the parameters of the model. The simplicity of the form of the model in Equation 2 makes it amenable for parameter learning in a maximum-likelihood framework (Section 2.1) or alternatively in a discriminative framework where the model can optimize a task-specific cost function as described in Section 2.2.

2.1. Parameter Learning: Maximum-Likelihood Formulation

In this section, we present a technique for learning the parameters of the model beased on a maximum-likelihood (ML) formulation. Given a training corpus $\{\overline{\mathbf{x}}_i\}_{i=1}^N$ of speech utterances, let $\hat{q}_t(\overline{\mathbf{x}}) \in \mathcal{Q}$ denote the true phone state label corresponding to frame t in the utterance $\overline{\mathbf{x}}$. Denote by the set $\mathcal{T}^l$, the set of frames in the corpus that correspond to the class $l \in \mathcal{Q}$, i.e. $\mathcal{T}^l = \{(t, \overline{\mathbf{x}}_i) \mid \hat{q}_t(\overline{\mathbf{x}}_i) = l\}$. The data-likelihood, $\mathcal{F}$, under the model in Equation 2 is given by,

$$\mathcal{F} = \prod_{l \in \mathcal{Q}} f_l(b) = \prod_{l \in \mathcal{Q}} \prod_{(t, \overline{\mathbf{x}}_i) \in \mathcal{T}^l} c_t(l; \overline{\mathbf{x}}_i) \tag{4}$$

The maximum likelihood estimate of the parameters $b(l, k)$ can be obtained by maximizing Equation 4, subject to the constraints in Equation 3. Since Equation 4 is a polynomial with positive coefficients, it can be solved iteratively using the Baum-Welch update equations, where we exploit the fact that the optimization can be done independently for each $l \in \mathcal{Q}$,

$$b(l, k) \leftarrow \frac{b(l, k) \nabla_{b(l,k)} f_l(b)}{\sum_{j \in \mathcal{Q}} b(l, j) \nabla_{b(l,j)} f_l(b)} \tag{5}$$

The gradients required in Equation 5 can be computed as,

$$\nabla_{b(l,k)} f_l(b) = \sum_{(t, \overline{\mathbf{x}}_i) \in \mathcal{T}^l} f_l(b) \frac{a_t(k; \overline{\mathbf{x}}_i)}{\sum_{j \in \mathcal{Q}} b(l, i) a_t(j; \overline{\mathbf{x}}_i)} \tag{6}$$

Substituting the gradients from Equation 6 in Equation 5, we derive the update equation as,

$$b(l, k) \leftarrow \frac{1}{|\mathcal{T}^l|} \sum_{(t, \overline{\mathbf{x}}_i) \in \mathcal{T}^l} \frac{b(l, k) a_t(k; \overline{\mathbf{x}}_i)}{\sum_{j \in \mathcal{Q}} b(l, j) a_t(j; \overline{\mathbf{x}}_i)} \tag{7}$$

2.2. Discriminative Parameter Learning

As we had mentioned briefly in Section 2.1, the mixing coefficients $b(l, k)$ in Equation 2 can be learned discriminatively to optimize a task-specific loss function. Note that the optimization function in Equation 4, is a product of independent functions $f_l(b)$, and can thus be optimized independently for each class $l \in \mathcal{Q}$. Thus, the final update formula in Equation 7 for $b(l, k)$ is independent of $b(l', k')$ for $l \neq l'$. The ML solution boosts the score $c_t(l; \overline{\mathbf{x}}_i)$ of each class independently, which might increase inter-class confusability. We therefore consider a frame discriminative objective function, $\mathcal{G}$, that attempts to boost the score for the correct class, while simultaneously reducing the score for competing classes as follows,

$$\mathcal{G} = \prod_{l \in \mathcal{Q}} \prod_{(t, \overline{\mathbf{x}}_i) \in \mathcal{T}^l} \frac{c_t(l; \overline{\mathbf{x}}_i)}{\sum_{j \in \mathcal{Q}} c_t(j; \overline{\mathbf{x}}_i)} = \prod_{l \in \mathcal{Q}} \prod_{(t, \overline{\mathbf{x}}_i) \in \mathcal{T}^l} \frac{c_t(l; \overline{\mathbf{x}}_i)}{C_t(\overline{\mathbf{x}}_i)} \tag{8}$$

where, we set $C_t(\overline{\mathbf{x}}_i) = \sum_{j \in \mathcal{Q}} c_t(j; \overline{\mathbf{x}}_i)$. Unlike the ML objective function in Equation 4, the frame discriminative objective function attempts to increase the likelihood of the correct class relative to all the other classes at that frame. The objective function in Equation 8 can be optimized iteratively using the Extended Baum-Welch (EBW) [11] updates,

$$b(l, k) \leftarrow \frac{b(l, k)(\nabla_{b(l,k)} \mathcal{G} + D)}{\sum_{j \in \mathcal{Q}} b(l, j)(\nabla_{b(l,j)} \mathcal{G} + D)} \tag{9}$$

where, D is a sufficiently large constant to ensure that the constraints on the mixing coefficients, $b(l,k)$ in Equation 3, are satisfied. The required gradient, $\nabla_{b(l,k)}\mathcal{G}$, can be computed as,

$$\nabla_{b(l,k)}\mathcal{G} = \mathcal{G}\left[\sum_{(t,\overline{\mathbf{x}}_i)\in\mathcal{T}^l}\frac{a_t(k;\overline{\mathbf{x}}_i)}{c_t(l;\overline{\mathbf{x}}_i)} - \sum_{j\in\mathcal{Q}}\sum_{(t,\overline{\mathbf{x}}_i)\in\mathcal{T}^j}\frac{a_t(k;\overline{\mathbf{x}}_i)}{C_t(\overline{\mathbf{x}}_i)}\right] \quad (10)$$

3. EXPERIMENTS

All of the results reported in this paper are conducted on the standard TIMIT phone recognition task [9]. The acoustic models used in the experiments are trained on the TIMIT training portion; results are reported on the the 192-sentence core test set. Following standard practice [12] we collapse the recognized phone labels down to 39 labels for scoring. In pilot experiments, we found that training the parameters $b(l,k)$ on the TIMIT training data did not lead to improved phone recognition accuracy on the TIMIT core test set. We therefore learn these parameters using the 400-sentence development set defined by Halberstadt and Glass [13]. The development set is also used for tuning hyperparameters, since in pilot experiments, we did not observe overfitting when training with the ML criterion in Equation 4. Since the EBW updates are iterative, they are sensitive to initialization. In all of our experiments, we initialize the mixing coefficients to a uniform distribution $b(l,k) = 1/|\mathcal{Q}|$.

As we mentioned in Section 2, posterior modeling techniques are general, and applicable to any system from which posterior representations can be derived.[1] In our experiments, we consider three representative posterior systems: a NN trained on S_{pif} posteriors [4], a deep belief network (DBN) trained on fBMMI features and likelihoods obtained from a GMM-HMM system. Details of each of these systems appear in Section 3.1.

3.1. Details of Representative Posterior Systems

The GMM-HMM system that we report results on is created as follows. First 13-dimensional MFCC features are created from the speech utterances, which serve as the initial acoustic features. Acoustic models are then trained using the following recipe [14] with the IBM Attila speech recognition toolkit. Training begins with the creation of a set of context-independent (CI) models. These CI models are then used to bootstrap the training of a set of context-dependent triphone models using linear discriminant analysis (LDA) features. This is followed by vocal tract length normalization (VTLN) and feature space Maximum Likelihood Linear Regression (fMLLR) which maps the features into a canonical speaker space. This is followed by creating a set of discriminative models and features using the Boosted Maximum Mutual Information (BMMI) criterion. The resulting set of models is then adapted using MLLR. The S_{pif} [4] features and the DBN system that we report results on are trained on these fBMMI features.

[1] In fact, since we first convert the posteriors to scaled likelihoods, the techniques are also applicable to systems such as GMM-HMM systems, where we can compute acoustic likelihoods directly.

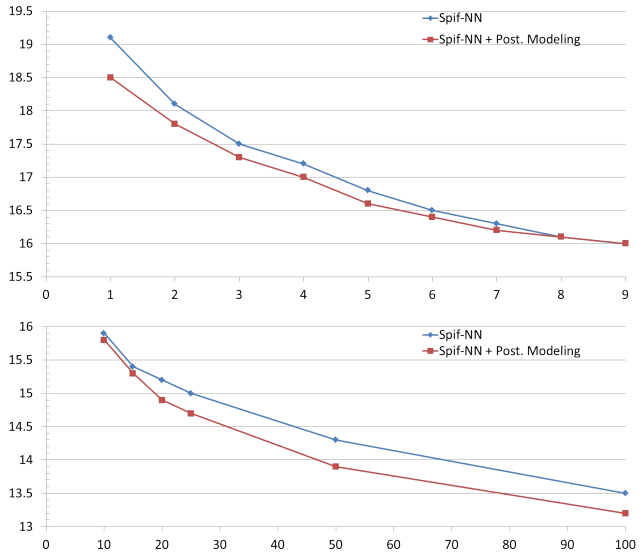

Fig. 1. N-best phone error rate results (%) for the S_{pif}–NN system across a range of N. The S_{pif} system with posterior modeling outperforms the system without posterior modeling at all levels of N.

System	PER (%) without post. modeling	PER (%) after post. modeling
S_{pif}–NN	19.1	18.5
DBN	19.0	18.8
GMM	20.0	18.9

Table 1. Phone error rates (PER) on TIMIT core test set obtained with and without posterior modeling using the ML-criterion described in Equation 4 on the three representative systems described in Section 3.1.

3.2. Results of Applying Posterior Modeling to the Representative Posterior Systems

We present results obtained on the three representative posterior systems after applying the ML-based posterior modeling technique described in Section 2.1 in Table 1. As can be seen from the results presented in the table, the use of the ML-based posterior modeling technique improves system performance in each of our representative systems by between 0.2–1.1% absolute (1.1–5.5 % relative). It is also interesting to note that the strength of the improvement is proportional to the performance of the initial system; all three systems ob-

System	Oracle Lattice PER without posterior modeling (%)	Oracle Lattice PER after posterior modeling (%)
S_{pif}–NN	2.6	1.8
DBN	2.9	2.3
GMM	3.0	2.0

Table 2. Lattice Oracle Phone error rates (PER) on TIMIT core test set obtained with and without posterior modeling using the ML-criterion described in Equation 4 on the three representative systems described in Section 3.1.

tained similar performance on the TIMIT core test set after posterior modeling. Furthermore, to the best of our knowledge, the PER of 18.5% that we obtained in these experiments, is the best reported number in the literature to date.

3.3. Lattice Oracle and N-best Results

As we mentioned in the introduction, one of the motivations for applying tied mixture-smoothing techniques for the S_{pif}–NN system was to mitigate the problem of posterior sharpness. The 1-best phone recognition error rates in Table 1 clearly suggest that posterior modeling techniques help improve system performance. In order to further examine the effect of posterior modeling on the representative posterior systems, we computed oracle lattice error rates[2] for the various representative systems. We report these results in Table 2. As can be seen in the table, posterior modeling improves even the oracle lattice WERs by between (0.6–1.0%). In order to explore this issue further, we also computed N-best PER results for the systems, which we plot for the S_{pif}–NN system for 'shallow' ($N = 1, \cdots, 9$) and 'deep' ($N = 10, \cdots, 100$), values of N in Figure 1. As can be seen in the plot, posterior modeling helps in both ranges of N. The plot for 'deep' values of N provides some evidence that posterior modeling may help to mitigate the problem of posterior sharpness by ensuring that the score of the correct class is not completely dominated by other incorrect classes.

3.4. Frame-Discriminative Objective Function

Finally, we explore the effectiveness of replacing the ML-based objective function with the discriminative frame-level objective function described in Section 2.2. Our results on the three representative posterior systems are presented in Table 3. Performance of the S_{pif}–NN and DBN systems were similar under either objective function although there was a small improvement (0.2% absolute) for the DBN system. However, the performance of the GMM posterior system was significantly worse with the frame-discriminative objective function. We observed that the GMM posterior system using the

[2]Lattices were created using the same pruning thresholds in order to make these numbers comparable across systems.

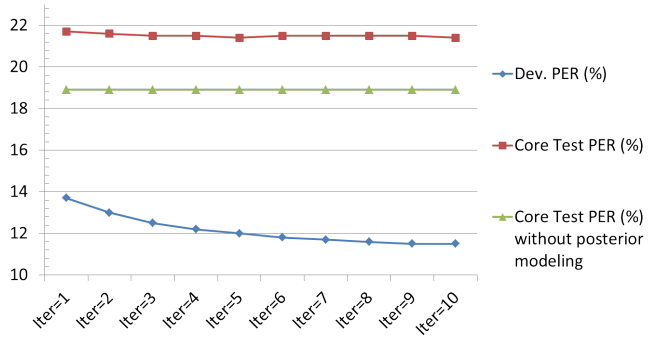

Fig. 2. PER for the GMM system with posterior modeling according to the discriminative criterion in Equation 8 plotted on the development and core test set as a function of the number of iterations of EBW updates.

System	No post. modeling	Frame-Disc obj. function	ML obj. function
S_{pif}–NN	19.1	18.8	18.5
DBN	19.0	18.6	18.8
GMM	20.0	21.5	18.9

Table 3. Phone error rates (PER) (%) on TIMIT core test set obtained with and without posterior modeling using the frame-discriminative criterion (Equation 8) and the ML-based criterion (Equation 4) on the three representative systems.

frame-discriminative objective function, had a substantially lower error rate on the development set ($\approx$ 4% absolute), which suggests that the increase in PER on the core test set is the result of overfitting as illustrated in Figure 2.

4. CONCLUSIONS AND FUTURE WORK

We presented the results of a set of detailed evaluations that were aimed at determining whether tied-mixture posterior modeling techniques are effective at improving performance for a number of posterior-based systems. In experimental evaluations, we obtained 1.1–5.5% relative improvements across three representative posterior systems using the ML-based criterion presented in Section 2.1. This included a core test error rate of 18.5% which to the best of our knowledge is the lowest reported error rate on TIMIT. We also proposed and evaluated a discriminative loss function aimed at minimizing frame-level errors. For two systems, performance of the discriminative objective was comparable to the ML objective, but did substantially worse in the GMM posterior system due to overfitting.

In future work, we would like to further explore discriminative posterior modeling techniques, and address some of the shortcomings of the model presented in Section 2.2. We would also like to evaluate these techniques for word recognition tasks.

5. REFERENCES

[1] G. Aradilla, *Acoustic models for posterior features in speech recognition*, Ph.D. thesis, École Polytechnique Federalé de Lausanne, 2008.

[2] Balakrishnan V., G. S. V. S. Sivaram, and S. Khudanpur, "Dirichlet mixture models of neural net posteriors for HMM-based speech recognition," in *Proc. ICASSP*, 2011.

[3] T. N. Sainath, D. Nahamoo, B. Ramabhadran, and D. Kanevsky, "Enhancing exemplar-based posteriors for speech recognition tasks," in *Proc. Interspeech*, September 2012.

[4] T. N. Sainath, D. Nahamoo, B. Ramabhadran, D. Kanevsky, V. Goel, and P. M. Shah, "Exemplar-based sparse representation phone identification features," in *Proc. ICASSP*, 2011.

[5] N. Morgan and H. Bourlard, "Continuous speech recognition," *IEEE Signal Processing Magazine*, vol. 12, no. 3, pp. 24–42, may 1995.

[6] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature stream extraction for conventional HMM systems," in *Proc. ICASSP*, 2000.

[7] J. Morris and E. Fosler-Lussier, "CRANDEM: Conditional random fields for word recognition," in *Proc. Interspeech*, 2009.

[8] R. Prabhavalkar, P. Jyothi, W. Hartmann, J. Morris, and E. Fosler-Lussier, "Investigations into the Crandem approach to word recognition," in *NAACL-HLT short paper session*, 2010.

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

[10] J. Bellegarda and D. Nahamoo, "Tied mixture continuous parameter modeling for speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, pp. 2033–2045, 1990.

[11] P. S. Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo, "An inequality for rational functions with applications to some statistical estimation problems," *IEEE Transactions on Information Theory*, vol. 37, pp. 107 –113, 1991.

[12] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, pp. 1641–1648, 1989.

[13] A. K. Halberstadt and J. R. Glass, "Heterogeneous acoustic measurements for phonetic classification," in *Proc. Eurospeech*, 1997.

[14] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *IEEE Workshop on Spoken Language Technology*, 2010.