Incorporating Sparse Representation Phone Identification Features in Automatic Speech Recognition using Exponential Families

Vaibhava Goel, Tara N. Sainath, Bhuvana Ramabhadran,
Peder A. Olsen, David Nahamoo, Dimitri Kanevsky

IBM T. J. Watson Research Center, Yorktown Heights, NY 10598
{vgoel, tsainath, bhuvana, pederao, nahamoo, kanevsky}@us.ibm.com

Abstract

Sparse representation phone identification features (SPIF) is a recently developed technique to obtain an estimate of phone posterior probabilities conditioned on an acoustic feature vector. In this paper, we explore incorporating SPIF phone posterior probability estimates in large vocabulary continuous speech recognition (LVCSR) task by including them as additional features of exponential densities that model the HMM state emission likelihoods. We compare our proposed approach to a number of other well known methods of combining feature streams or multiple LVCSR systems. Our experiments show that using exponential models to combine features results in a word error rate reduction of 0.5% absolute (18.7% down to 18.2%); this is comparable to best error rate reduction obtained from system combination methods, but without having to build multiple systems or tune the system combination weights.

1. Introduction

System combination is a popular method to combine different speech recognition systems which exhibit complementary information to improve overall word error rate. Combining systems can be done at many different levels in the recognition process. Typically the success of a combination scheme depends on the amount of complementarity between different methods at a specific level, where the more complementary two systems are the more gain can be achieved by combining systems.

Early fusion methods typically combine two different feature streams at the input feature level [1]. A new model is built using this combined feature stream and used for decoding. Mid-fusion methods typically explore, for each acoustic segment or frame, combining scores from different systems [2]. Finally, late-fusion methods look to combine the hypothesizes of different recognizers. Recognizer Output Voting Error Reduction (ROVER), N-best rover [3], and cross-adaptation [4] are three popular late-fusion methods.

In this paper we explore use of general exponential density based acoustic models [5] to combine multiple feature streams. This combination is achieved by simply including these streams as features of the exponential model. In particular, we combine a baseline set of feature space maximum mutual information (fMMI) features [6] with a set of phone-based posterior probabilities obtained from recently introduced sparse representation phone identification features (SPIF) [7]. SPIF relies on a sparse representation of the given test acoustic vector in an over complete basis spanned by training acoustic vectors. SPIF naturally extends to other classes such as sub-phones or context dependent sub-phones, and has been successfully applied to frame level phone classification tasks.

We compare our proposed feature combination method with a number of alternative system combination techniques, namely model-combination [2], rover [9], N-best rover [3], and cross-adaptation [4]. Our LVCSR experiments indicate that using exponential models to combine features produces results comparable to other system combination methods, but without having to build multiple systems or tune the combination weights.

The rest of this paper is organized as follows. Section 2 reviews the SPIF technique for obtaining phone posteriors. Section 3 presents our proposed exponential families for combining fMMI and SPIF features. Section 4 presents the alternative system combination methods that we compared with. Section 5 presents the experimental setup, followed by a discussion of results in Section 6. Finally, Section 7 concludes the paper.

2. Sparse Representation Phone Identification Features

We first discuss classification using sparse representation and then show how it is used to compute the SPIF features.

2.1. Classification Using Sparse Representations

Let $x_{i,j} \in \mathbb{R}^m, j = 1, \ldots, n_i$ be m dimensional feature vectors from training set of class i, and let

$$H_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}] \in \mathbb{R}^{m \times n_i} \quad (1)$$

be a matrix containing these training vectors. Furthermore, let

$$H = [H_1, H_2, \ldots, H_w] = [x_{1,1}, x_{1,2}, \ldots, x_{w,n_w}] \quad (2)$$

be a dictionary containing training features from classes $i = 1, \ldots, w$. $H \in \mathbb{R}^{m \times N}$.

Given a test feature vector y, we find $\beta \in \mathbb{R}^N$ satisfying $y = H\beta$ while requiring β to be as sparse as possible. Ideally, all nonzero entries of β should correspond to the entries in H with the same class as y. In practice, however, this is not the case, and the class of y is determined as follows. Let $\beta^{(i)}$ be a vector formed from entries of β that correspond to class i. y is then classified to belong to class for which $\| \beta^{(i)} \|_2$ is largest.

This sparse representation classification decision was explored in [7] to measure frame accuracy, and we will use this classification decision to construct a set of phone identification features, which we discuss further in the next section.

2.2. Phone Identification Features

Let us define $H_{\text{phnid}} = [p_{1,1}, p_{1,2}, \ldots, p_{w,n_w}] \in \mathbb{R}^{w \times N}$. A column of $p_{i,j} \in H_{\text{phnid}}$ corresponds to training vector $x_{i,j}$

belonging to class i. $p_{i,j}$ is a vector of size w with a value 1.0 in position i and zeros everywhere else. Figure 1 illustrates the mapping between H and H_{phnid}.

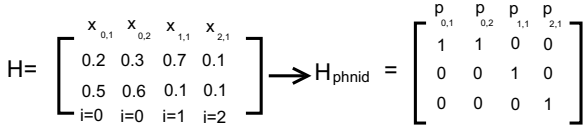

Figure 1: H_{phnid} corresponding to H

Given a test feature vector y, we first find a sparse β as a solution of $y = H\beta$. We then compute $\mathbf{p}_{\text{spif}}$ as

$$\mathbf{p}_{\text{spif}} = \frac{H_{\text{phnid}}\beta^2}{\parallel H_{\text{phnid}}\beta^2 \parallel_2}, \tag{3}$$

where β^2 contains squared elements of β. Notice we use β^2, as this is similar to the classification rule discussed in previous section. We will refer to this $\mathbf{p}_{\text{spif}}$ vector as a SPIF feature.

2.3. Constructing H

Ideally, H represents a dictionary of all training examples. However, pooling together all training data from all classes into H will make it large and will make solving for β intractable. We therefore construct an H matrix for each test frame y by identifying training frames that are "similar" to y, as follows. The training data is decoded using a trigram language model (LM) and for each frame the Gaussian that best aligns with this frame is determined. At test time, the test data is decoded using a trigram LM and the Gaussian g_1 that best aligns with frame y is determined. Next, four more Gaussians in the acoustic model that are closest to g_1, based on Euclidean distance between Gaussian means, are identified. The matrix H is then constructed from training frames that aligned with these top five Gaussians. Often a large number of frames correspond to a Gaussian and we keep a randomly sampled subset of size N. For experiments conducted in this paper, N is chosen to be 200, 100, 100, 50, and 50 for the top five Gaussians, respectively.

3. Exponential Families Combining fMMI and SPIF Features

The general exponential family is given by

$$\mathrm{P}(\mathbf{x};\theta) = \frac{e^{\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x})}}{Z(\boldsymbol{\theta})} \quad \text{where} \quad Z(\boldsymbol{\theta}) = \int_D e^{\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x})}\, \mathrm{dx} \tag{4}$$

where $\boldsymbol{\phi}$ is the feature function (also called sufficient statistic) characterizing the family, $\boldsymbol{\theta}$ are the parameters in the exponential family and $Z(\boldsymbol{\theta})$ is the partition function or normalizer. D is the domain for which x is defined.

Let $\mathbf{x}$ denote m dimensional vector of fMMI features, and let s denote an HMM state. For fMMI features, the emission density of state s is modeled by a mixture of diagonal covariance Gaussians. Using

$$\begin{aligned} \boldsymbol{\theta} &= \left[\frac{\mu_1}{\sigma_1^2}, \cdots, \frac{\mu_m}{\sigma_m^2}, \frac{-0.5}{\sigma_1^2}, \cdots, \frac{-0.5}{\sigma_m^2}\right] \\ \boldsymbol{\phi}^{(g)}(\mathbf{x}) &= \left[x_1, \cdots, x_n, x_1^2, \cdots, x_m^2\right], \\ \log Z^{(g)}(\boldsymbol{\theta}) &= 0.5m\log(2\pi) + \sum_{i=1}^{m} \log\sigma_i + 0.5\frac{\mu_i^2}{\sigma_i^2} \end{aligned} \tag{5}$$

the GMM can be written as a mixture of exponential densities

$$P(\mathbf{x}|s) = \sum_{e\in\mathcal{E}(s)} \pi_e P(\mathbf{x}|\boldsymbol{\theta}_e) = \sum_{e\in\mathcal{E}(s)} \pi_e \frac{e^{\boldsymbol{\theta}_e^T \boldsymbol{\phi}^{(g)}(\mathbf{x})}}{Z^{(g)}(\boldsymbol{\theta}_e)}. \tag{6}$$

As discussed in Section 2, the $\mathbf{p}_{\text{spif}}$ features provide, corresponding to each $\mathbf{x}$, an estimate of phone posterior probabilities. In the following we drop the subscript and use $\mathbf{p}$ to denote $\mathbf{p}_{\text{spif}}$.

Using $\log p_i$ as features, the Dirichlet family of densities is an exponential family with the following sufficient statistic and partition function

$$\begin{aligned} \boldsymbol{\phi}^{(d)}(\mathbf{p}) &= [\log(p_1), \ldots, \log(p_w)] \\ Z^{(d)}(\boldsymbol{\theta}) &= \frac{\prod_{i=1}^{w}\Gamma(1+\theta_i)}{\Gamma(w+\sum_{i=1}^{w}\theta_i)}. \end{aligned} \tag{7}$$

$\theta > -1$ are valid parameter values for the Dirichlet family.

To combine fMMI and SPIF features, we experimented with two exponential families. Both these families use the following sufficient statistic

$$\boldsymbol{\phi}(\mathbf{x}, \mathbf{p}, s) = \left[\boldsymbol{\phi}^{(g)}(\mathbf{x}), \log(p_{L(s)})\right] \tag{8}$$

where $L(s)$ denotes the phone index to which HMM state s belongs to. These families differ in how they treat the domain of the $\log(p_{L(s)})$ feature.

By including only one $\log(p_{L(s)})$ feature per Gaussian, we increase the number of parameters per Gaussian by one.

In general $\mathbf{p}$ and $\mathbf{x}$ are dependent variables, and hence the partition function for any exponential family specified by (8) is not easy to compute. To avoid having to resort to sampling based approaches [5], we make the simplifying assumption that $\mathbf{x}$ and $\mathbf{p}$ are independent feature sources.

3.1. Model1

The first exponential family, termed *model1*, utilizes knowledge that $\mathbf{p}$ is a vector of posterior probabilities. The partition function for this family is give as

$$\int_{\mathbf{x},\mathbf{p}} e^{\lambda^T \boldsymbol{\phi}^{(g)}(\mathbf{x}) + \theta\log(p_{L(s)})} d\mathbf{x}\, d\mathbf{p} = Z^{(g)}(\lambda)\frac{\Gamma(1+\theta)}{\Gamma(w+\theta)} \tag{9}$$

This model has a potential drawback - for $\theta \in (-1, 0)$, the model favors lower posterior values over higher ones, and this is exact opposite of the intuitively desirable behavior that acoustic likelihood for a given class should increase if that class posterior increases.

Another potential issue with this model is that if w, the dimension of $\mathbf{p}$, is large then this model is susceptible to overtraining, and we need to constrain the θ values to prevent overtraining. To alleviate these issues we explored the following alternative.

3.2. Model2

The second exponential family, *model2*, disregards the fact that $u = \log(p_{L(s)})$ is a component of a larger vector of log-posteriors. For this feature, we use $\phi(u) = u$, $u \in (-\infty, 0)$. This contributes $Z^{(e)}$ to the overall partition function, where

$$Z^{(e)} = \int_{-\infty}^{0} e^{\theta u} \mathrm{d}u = \frac{1}{\theta}. \tag{10}$$

$\theta > 0$ are valid parameter values. The overall partition function for model2 is $Z^{(g)}\, Z^{(e)}$.

3.3. Parameter Estimation

To estimate parameters of the exponential models, we first trained a GMM using fMMI features. This GMM was trained in fMMI feature space using boosted MMI [6] to discriminatively estimate the feature space and Gaussian parameters.

The two exponential models, model1 and model2, were initialized from these Gaussians by adding $\log(p_{L(s)})$ feature and setting the corresponding parameter to 0. Then, fixing the parameters corresponding to the Gaussian portion of the features, $\boldsymbol{\phi}^{(g)}(\mathbf{x})$, the single parameter corresponding to $\log(p_{L(s)})$ feature was trained under the maximum likelihood (ML) objective.

Using the expectation-maximization (EM) procedure, the auxiliary function to be maximized is

$$\begin{aligned} L(\theta) &= \theta \sum_t \gamma(t,e) \log(p_{L(s(e))}) - \log(Z(\theta)) \sum_t \gamma(t,e), \\ &= \theta \mathbf{s}(e) - n(e) \log Z(\theta), \end{aligned} \quad (11)$$

where $\gamma(t,e)$ is the posterior probability of observing component e at time t.

For model1, $L(\theta)$ and its gradient are

$$\begin{aligned} L(\theta) &= \theta \mathbf{s}(e) - n(e) \left[\log \Gamma(1+\theta) - \log \Gamma(w+\theta)\right] \\ \nabla_\theta L(\theta) &= \mathbf{s}(e) - n(e) \left[\Psi(1+\theta) - \Psi(w+\theta)\right], \end{aligned} \quad (12)$$

where $\Psi(\theta) = \Gamma'(\theta)/\Gamma(\theta)$ is the Digamma function. Both Gamma and Digamma functions were computed using a numerical implementation [8].

For model2, $L(\theta)$ and its gradient are

$$\begin{aligned} L(\theta) &= \theta \mathbf{s}(e) - n(e) \log(\theta) \\ \nabla_\theta L(\theta) &= \mathbf{s}(e) - n(e) \frac{1}{\theta} \end{aligned} \quad (13)$$

From the gradient it is immediately seen that optimum value of θ is $n(e)/\mathbf{s}(e)$.

3.4. Maximum Likelihood Linear Regression of Exponential Models

At test time, for GMMs, typically an unsupervised multi-class maximum likelihood linear regression (MLLR) based adaptation is carried out to yield speaker specific model parameters.

To create speaker specific versions of our exponential models we carry out a modified version of MLLR. Only the exponential model parameters corresponding to the Gaussian portion of the features are updated using linear regression transforms. However, the statistics needed to estimate the adaptation transforms are obtained using the entire exponential model. We will use eMLLR to denote this way of performing MLLR. The parameter corresponding to $\log p$ feature is not adapted.

4. Alternative Combination Techniques

4.1. Model Combination

Model combination is used to linearly combine log-likelihood scores coming from different systems [2]. Model combination assumes that for a given state at time t, the feature vectors for each stream S are statistically independent. This allows the output distribution $b_j(o^t)$ for a specific state j to be computed as follows, where w_s is the weight for stream S.

$$b_j(o^t) = \prod_{s=1}^{S} \left[\sum_{m=1}^{M_{js}} c_{jms} N(o_s^t; \mu_{jms}, \Sigma_{jms}) \right]^{w_s} \quad (14)$$

Several schemes have been explored to estimate the HMM parameters (i.e. $c_{jms}, \mu_{jms}, \Sigma_{jms}$) for each stream. In this paper, we explore estimating the parameters and decision trees separately for each feature stream. The system weights w_s are tuned through exhaustive search.

4.2. ROVER and N-best ROVER

In a ROVER system combination [9], first the 1-best system outputs from multiple recognizers are combined into a single word transition network using a dynamic programming alignment tool. Then, a voting module scores each branching point in the network and selects the best word at each point through majority voting. While ROVER has shown to be robust as the number of systems increases, when fewer systems are combined typically approaches which consider multiple alternative hypotheses perform better than ROVER.

One such method which considers multiple alternative hypotheses is N-best ROVER [3]. During N-best ROVER, the n-best outputs from multiple ASR systems are word-aligned. Each system computes its own word-posterior estimate, and the total word-posterior is a weighted combination of word-posteriors from individual systems, where the weights are determined empirically. Then a voting module selects the best word sequence as that with the highest score.

4.3. Cross Adaptation

Model adaptation methods in speech recognition transform the models of a system given the output of a decoder, the most popular of which is MLLR. However, after a few iterations of adapting models of a system based on it own system output, improvements from adaptation become minimal. Cross-adaptation [4] addresses this issue by using the output of one system to adapt the models of a second system. Gains from cross-adaptation occur when both systems make different errors, and thus the second system obtains complementary information that it was unable to obtain from its own output.

5. Experimental Setup

We evaluated the linear exponential family on a Broadcast News LVCSR task. The acoustic model training set comprises 50 hours of data from the 1996 and 1997 English Broadcast News Speech corpora (LDC97S44 and LDC98S71), and was created by selecting entire shows at random. The EARS Dev-04f set (dev04f), a collection of 3 hours of audio from 6 shows collected in November 2003, is used for testing the models.

The acoustic features are obtained by first computing 13-dimensional PLP features with speaker-based mean, variance, and vocal tract length normalization. Nine such features were concatenated and projected to a 40 dimensional space using LDA. The 40 dimensional features were further normalized using one feature space linear regression transform (fMLLR) per speaker. An fMMI transform [6] was estimated to arrive at the final feature space in which acoustic models were trained.

The acoustic model trained in the fMMI feature space consisted of 44 phones. Each phone was modeled as three-state, left-to-right HMMs with no skip arcs. Context dependency of these states was incorporated using decision trees, resulting in 2206 context dependent states. HMM states that model silence were context independent. Mixtures of Gaussian distributions were used to model each state, with the overall model having 50K components. The exponential models that combine fMMI and SPIF features used this acoustic model as the starting point.

To evaluate alternative combination approaches of Section 4, an acoustic model was built in the SPIF feature space. This model was very similar to the model in fMMI space, except it had 2168 context dependent states containing 50K Gaussians.

The language model used for decoding is a 54M 4-gram, interpolated backoff model trained on a collection of 335M words, as discussed by Kingsbury et. al. [10]. The recognition lexicon contains 84K word tokens, with an average of 1.08 pronunciation variants per word. Where possible, pronunciations were based on PRONLEX (LDC97L20).

6. Results

The baseline fMMI+bMMI acoustic model had a word error rate of 19.4% on the dev04f test set. If we adapt the acoustic models using speaker specific unsupervised MLLR, the word error rate drops to 18.7%. The baseline SPIF model had a WER of 19.3%; this does not improve with MLLR.

Table 1 shows the test set WER performance of model combination (Section 4.1) and N-best ROVER (Section 4.2) as a function of system combination weight w on the fMMI system; the weight on SPIF system is $1 - w$.

fMMI+bMMI+MLLR = 18.7%; SPIF baseline = 19.3%						
	system combination weight					
	0.9	0.8	0.7	0.6	0.5	0.4
M	18.1%	18.4%	18.7%	18.9%	19.1%	19.2%
N	18.5%	18.5%	18.4%	18.3%	18.3%	19.0%

Table 1: WER as a function of system combination weight. Row M corresponds to model combination and row N to N-best ROVER.

Exponential models were built by including SPIF features in baseline fMMI+bMMI model. Table 2 shows performance of model1 (Section 3.1). As discussed in Section 3.1, to prevent overtraining we need to threshold θ value; WER results with various threshold values are shown in the first row of Table 2. The second row shows WER numbers when we further constrain $0 \leq \theta$. From these results we note that bounding θ from above and below is needed to achieve optimal performance.

fMMI+bMMI baseline = 19.4%; SPIF baseline = 19.3%					
	threshold (t)				
	0.5	1.0	2.0	3.0	4.0
$-1 \leq \theta \leq t$	18.9%	18.8%	18.8%	19.1%	19.2%
$0 \leq \theta \leq t$	18.7%	18.6%	18.9%	19.1%	19.2%

Table 2: WER results of model1 with various thresholds

WER results of ML estimated model2, as well as those of various other system combination methods, are shown in Table 3. Both model1 and model2 achieve WER of 18.2% after eMLLR (Section 3.4). These WER numbers are comparable to the best error rate obtained using system combination techniques.

7. Conclusions

In this paper, we presented an technique to combining fMMI and SPIF features in an exponential model framework. Our results on an LVCSR task indicated that using exponential models to combine features resulted in a WER reduction comparable to best system combination method. Furthermore, these gains are obtained without the effort (and parameters) involved in building multiple systems, and also without any optimization of system combination weight.

(a)	baseline	fMMI+ bMMI	19.4%
(b)		(a) + MLLR	18.7%
(c)	SPIF baseline		19.3%
(d)	ROVER	(b) & (c), 1-best	19.0%
(e)		(b) & (c), n-best	18.3%
(f)	Model Comb.	(b) & (c)	18.1%
(g)	Cross Adapt	(a) using (c)	19.0%
(h)	Exp Models	model1 ($t = 1.0$)	18.6%
(i)		(h)+ eMLLR	18.2%
(j)		model2	18.6%
(k)		(j)+ eMLLR	18.2%

Table 3: WER results of various feature combination methods. eMLLR is as discussed in Section 3.4

8. Acknowledgements

The authors would like to thank Hagen Soltau, George Saon, Brian Kingsbury, Stanley Chen and Abhinav Sethy for their contributions towards the IBM toolkit and recognizer utilized in this paper.

9. References

[1] A. Zolnay, R. Schluter, and H. Ney, "Acoustic Feature Combination for Robust Speech Recognition," in *Proc. ICASSP*, 2005.

[2] C. Ma, H. Kuo, H. Soltau, X. Cui, U. Chaudhari, L. Mangu, and C. Lee, "A comparative study on system combination schemes for LVCSR," in *Proc. ICASSP*, 2010.

[3] A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. R. R. Gadde, M. Plauche, C. Richey, E. Shriberg, K. Sonmez, F. Weng, and J. Zheng, "The SRI March 2000 Hub-5 Conversational Speech Transcription System," in *NIST Speech Transcription Workshop*, 2000.

[4] S. Stuker, C. Fugen, S. Burger, and M. Wolfel, "Cross-System Adaptation and Combination for Continuous Speech Recognition: The Influence of Phoneme Set and Acoustic Front-End," in *Proc. Interspeech*, 2006.

[5] V. Goel and P. Olsen, "Acoustic Modeling Using Exponential Families," in *Proc. Interspeech*, 2009.

[6] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training ," in *Proc. ICASSP*, 2008.

[7] T. N. Sainath, D. Nahamoo, R. Ramabhadran, and D. Kanevsky, "Sparse representation phone identification features for speech recognition," Speech and Language Algorithms Group, IBM, Tech. Rep., 2010.

[8] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge University Press, 1992.

[9] J. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proc. ASRU*, 1997.

[10] B. Kingsbury, "Lattice-based Optimization of Sequence Classification Criteria for Neural-Network Acoustic Modeling," in *Proc. ICASSP*, 2009.