FEATURE GENERATION BASED ON MAXIMUM NORMALIZED ACOUSTIC LIKELIHOOD FOR IMPROVED SPEECH RECOGNITION

Xiang Li and Richard M. Stern

Department of Electrical and Computer Engineering and School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213 USA

{xiangl, rms}@cs.cmu.edu

ABSTRACT

Feature representation is a very important factor that has a great effect on the performance of speech recognition systems. In this paper we focus on a feature generation process that is based on the linear transformation of an original log-spectral representation. While conventional linear feature generation methods generally use objective functions that are not closely related to recognition accuracy, our linear feature generation method attempts to find a transformation matrix that maximizes the normalized acoustic likelihood of the most likely state training data, a measure that is directly related to the classification error rate in speech recognition. The transformation matrix is generated using a gradient ascent optimization process, with the normalized acoustic likelihood of the most likely state sequence as the objective function. Experimental results using the DARPA RM corpus show that the proposed method consistently decreases word error rates compared to conventional linear feature generation methods.

1. INTRODUCTION

As is the case with all pattern classification systems, performance of a speech recognition system depends critically on the features it uses. Features that are based on linear transformations of log-spectral representations of speech are commonly used in speech recognition, including the feature extraction procedures of Mel frequency cepstral coefficients (MFCC) [1], principal component analysis or Karhunen-Loeve transformation (PCA/KLT) [2], and linear discriminant analysis (LDA) [3][4]). These features are relatively easy to generate, and good performance has been obtained using them.

Despite the success of these linear feature of them are based on heuristics, as neither the objectives of "maximal separation" used in LDA [3][4] nor "maximum preservation" in PCA/KLT [2] directly relate to our real objective of minimal word error rates (WERs). The objective of MFCC [1] is merely that of providing a transformation of the original log-spectral representation that is smoother, more decorrelated, and reduced in dimensionality. In contrast, the aim of this paper is to derive a linear feature based on an objective which is more intimately linked to the goal of minimizing recognition error rate. Specifically, we will use as our objective the normalized likelihood of the most likely state sequences generated from forced alignment, a measure that can be thought of as the *a posteriori* probability of the most likely state sequences assuming that the *a priori* probabilities of the state sequences are equal. With our objective defined to be this normalized acoustic likelihood, we will use a gradient ascent optimization procedure to tune a feature transformation matrix that maximizes the objective function, and achieve our goal of minimizing the WER of the system.

In the following section we will describe our new linear feature generation method starting from the simple case of a single Gaussian state output distribution, and then extend it to the case of state output distributions that are modeled as Gaussian mixtures. In Sec. 3, we report our experimental results using the DARPA Resource Management (RM) corpus, and we present our discussion and conclusions in Sec. 4.

2. LINEAR FEATURE GENERATION USING MAXIMUM NORMALIZED ACOUSTIC LIKELIHOOD

2.1. Linear feature generation using Gaussian state output distributions

As noted above, our feature generation method tries to find the transformation matrix that maximizes an objective function which is closely related with the recognition accuracy of the system. This function is the normalized acoustic likelihood P_c of the most likely state sequence in the training data:

$$P_c = \prod_i \frac{P(X_i|C_{h,i})}{\sum_{j=1}^{C} P(X_i|C_j)} \quad (1)$$

where X_i is the sample of training data in frame i, $C_{h,i}$ is the most likely state in frame i from the forced-alignment result, and C is the total number of recognition classes. The value of the normalized acoustic likelihood computed within each frame is accumulated across the training data.

It can be seen from Eq. (1) that P_c depends on the specific value of the feature vector X_i in each frame and the parameters of the model of the observation probabilities of each recognition class. Since the training data sample X_i is fixed in the original feature space before transformation, the normalized acoustic likelihood P_c computed in the original feature space will depend only on the model parameters that describe the observation probability of each recognition class. Given the fact that the training data can be partitioned into recognition classes on a frame-by-frame

basis, and the parameters of the observation probability models of each recognition class are generated using a Maximum Likelihood (ML) estimation approach based on the training data assigned (*e.g.* using approaches such as Baum-Welch training or the *K*-means training algorithms), P_c then depends only on the partitioning of the training data in the feature space before the transformation. Applying a linear transform *A* to the original feature space can cause both P_c and the new partition of the transformed training data to change. But if we assume that the partition of the transformed data is the same as the partition in the space (which can be easily enforced via forced alignment of the training data), then P_c in the new feature space is only a function of the transformation matrix *A* that can be optimized by computing its derivative with respect to the matrix *A*.

Our method tries to find the transformation matrix *A* which maximizes P_c, the normalized acoustic likelihood of the most likely state sequence in the transformed feature space as in Eq. (2):

$$A = ArgmaxP_c = Argmax\prod_i \frac{P(X_i'|C_{h,i})}{\sum_{j=1}^{C} P(X_i'|C_j)} \quad (2)$$

where

$$X_i' = AX_i \quad (3)$$

is the transformed feature vector in frame *i*.

Since the closed form solution to Eq. (2) is unknown, we use iterative procedure to maximize P_c with respect to the transformation matrix *A* using gradient ascent method. For simplicity, we replace P_c by $LogP_c$ as our objective function. The gradient of $LogP_c$ with respect to transformation matrix *A*,$\nabla_A LogP_c$, can be expressed as:

$$\nabla_A LogP_c = \sum_i \left\{ \nabla_A LogP(X_i'|C_{h,i}) - \nabla_A Log\left[\sum_{j=1}^{C} P(X_i'|C_j)\right] \right\} \quad (4)$$

where

$$\nabla_A Log\left[\sum_{j=1}^{C} P(X_i'|C_j)\right] = \frac{\sum_{j=1}^{C} \nabla_A P(X_i'|C_j)}{\sum_{j=1}^{C} P(X_i'|C_j)} = \frac{\sum_{j=1}^{C} P(X_i'|C_j)\nabla_A LogP(X_i'|C_j)}{\sum_{j=1}^{C} P(X_i'|C_j)} \quad (5)$$

Since $P(X_i'|C_j)$ is the acoustic likelihood of recognition class *j* based on the transformed data, it can be easily computed result of the previous iteration. The remaining term is $\nabla_A LogP(X_i'|C_j)$, the gradient of the log acoustic likelihood of class *j* in the transformed data X_i' with respect to the transformation matrix *A*.

If the acoustic likelihood of the transformed data X_i' given class *j* is generated from a single Gaussian probability distribution (as described in our previous paper [5]), $P(X_i'|C_j)$ can be written as:

$$P(X_i'|C_j) = \frac{1}{(2\pi)^{\frac{M}{2}}|\Sigma_j'|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(X_i' - \mu_j')^T \Sigma_j'^{-1}(X_i' - \mu_j')\right\} \quad (6)$$

where *M* is the dimensionality of the transformed data, μ_j' and Σ_j' are the transformed means and covariance of the recognition class *j,* respectively:

$$\begin{aligned} \mu_j' &= A\mu_j \\ \Sigma_j' &= A\Sigma_j A^T \end{aligned} \quad (7)$$

where μ_j and Σ_j are the mean and covariance of class *j* estimated from the training data in the original feature space.

If we only use the diagonal components of the covariance matrix in our modeling, we obtain:

$$\nabla_A LogP(X_i'|C_j) = -\sum_{k=1}^{M} \nabla_A \left[\frac{(x'_{i,k} - \mu'_{j,k})^2}{2\sigma'^2_{j,k}} + \frac{\log(\sigma'^2_{j,k})}{2}\right] \quad (8)$$

where $x'_{i,k}$, $\mu'_{j,k}$ and $\sigma'^2_{j,k}$ are the individual components of the transformed feature vectors in frame *i*, and the mean and covariance of class *j*.

We know from Eq. (3) and Eq. (7) that $x'_{i,k}$, $\mu'_{j,k}$ and $\sigma'^2_{j,k}$ are all functions of the transformation matrix *A,* so their derivatives with respect to *A* can be easily computed. This produces the closed form solution for $\nabla_A LogP(X_i'|C_j)$ in Eq. (8) that is stated in the Appendix. Substituting this result into Eq. (4) and Eq. (5) produces the closed-form solution of $\nabla_A LogP_c$. This quantity is then used as the increment in the gradient ascent the *A* matrix that optimizes the term of log-normalized acoustic likelihood as our objective function.

2.2. Linear feature generation using Gaussian-mixture state output distributions

Because state-of-the-art recognizers use mixtures of Gaussians as state output distribution instead of the single Gaussian, we extend our derivation to this case. With Gaussian mixtures as output dis-

tributions, the likelihood of transformed feature vector X_i' given the state C_j can be written as

$$P(X_i'|C_j) = \sum_{l=1}^{L} w_{j,l} P(X_i'|C_{j,l}) \quad (9)$$

where $w_{j,l}$ is the mixture coefficient for the Gaussian component l of state j. $P(X_i'|C_{j,l})$ is the acoustic likelihood or observation probability from the individual component l, which can be written as

$$P(X_i'|C_{j,l}) = \frac{\exp\left\{-\frac{1}{2}(X_i' - \mu'_{j,l})^T \Sigma'^{-1}_{j,l} (X_i' - \mu'_{j,l})\right\}}{(2\pi)^{\frac{M}{2}} |\Sigma'_{j,l}|^{\frac{1}{2}}} \quad (10)$$

From Eq. (9) and Eq. (10), it is easily seen that the term $\nabla_A LogP(X_i'|C_j)$ can be re-written as:

$$\begin{aligned}\nabla_A LogP(X_i'|C_j) &= \nabla_A Log\left[\sum_{l=1}^{L} w_{j,l} P(X_i'|C_{j,l})\right] \\ &= \frac{1}{\sum_{l=1}^{L} w_{j,l} P(X_i'|C_{j,l})} \cdot \sum_{l=1}^{L} w_{j,l} \nabla_A P(X_i'|C_{j,l}) \\ &= \frac{\sum_{l=1}^{L} w_{j,l} \cdot P(X_i'|C_{j,l}) \cdot \nabla_A LogP(X_i'|C_{j,l})}{\sum_{l=1}^{L} w_{j,l} P(X_i'|C_{j,l})}\end{aligned} \quad (11)$$

As reflected in Eq. (11), the gradient of the log likelihood $\nabla_A LogP(X_i'|C_j)$ in the case of Gaussian mixtures is simply a weighted sum of the gradient of each Gaussian component within the mixture. The only difference from the case of single-Gaussian output distributions is that we must compute the first derivative of each Gaussian component in the mixture and take the weighted sum of these derivatives.

3. EXPERIMENTAL RESULTS

We carried out a series of experiments using the DARPA Resource Management (RM) database to compare the performance of our proposed method with that of other feature generation methods. All of these experiments were conducted using the CMU SPHINX-III speech recognition system with 3-state continuous HMMs. The states of different HMMs were tied together depending on the context information of the phone associated with each HMM. The total number of tied states was 2000. All features used in these experiments were generated from linear transformations of log-spectral features. We used a bigram language model.

WER (%)	1 Gaus	2 Gauss	4 Gauss	8 Gauss
MFCC	10.29	8.68	7.66	8.31
PCA	8.96	7.86	7.30	7.85
LDA	**8.36**	**7.67**	**6.93**	**7.51**
A1Gauss	**7.57**	6.82	6.98	7.46
A2Gauss	7.62	**6.69**	6.50	7.33
A4Gauss	8.34	7.11	**6.34**	7.09
A8Gauss	8.37	7.30	6.48	**6.88**

Table 1: Word Error Rates for RM corpus obtained using various linear feature generation schemes. The columns contain results for systems that are trained and tested using 1, 2, 4, and 8 mixtures. The contain results obtained using transformation matrices *A* that were optimized using 1, 2, 4, and 8 Gaussians.

In addition to the methods described in this paper, we also evaluated the performance of MFCC, PCA and LDA features for comparison. We used a state-based class label for the class labels for each feature frame needed to generate the transformation matrix using the LDA method. In our new feature generation method, we used the steepest gradient ascent method as the optimization procedure, using the transformation matrix obtained by LDA as the initial value. The optimization process was terminated when the results converged. We compare results obtained using 1, 2, 4, and 8 Gaussian mixtures in the output probabilities to obtain mal transformation matrix *A*. Similarly, we trained and tested separate systems using 1, 2, 4, and 8 Gaussians for the state output distribution. The complete experimental results are reported in Table 1. The columns of Table 1 contain results for recognition systems that are trained and tested using 1, 2, 4, and 8 Gaussian mixtures. The last four rows of Table 1 contain results obtained with transformation matrices *A* that were optimized using 1, 2, 4, and 8 Gaussians.

We also computed the statistical significance of the difference between results obtained using our proposed methods and LDA, the best-performing previous method. In each column of WERs in Table 1, we computed the statistcial significance between the LDA result and the result from our normalized featurse that had been generated using the same number of Gaussian components as the number of Gaussians per mixture used in training/testing. The results obtained from the matched pairs method [6] are reported in Table 2.

4. DISCUSSION AND CONCLUSIONS

We first note that our proposed method outperforms conventional linear feature generation methods, and the improvements are statistical significant.

	A1Gauss	A2Gauss	A4Gauss	A8Gauss
P	0.015	0.005	0.043	0.060

Table 2: Statistical significance (P) comparing results using LDA and results using tmaximum normalized acoustic likelihood. *P* was calculated using the matched pairs test.

We also observe that the best performance in each column is obtained when the number of Gaussians used to optimize the linear transformation matrix *A* is identical to the number of Gaussians used to train and test the speech recognition system. In general, performance improves as the number of Gaussians increases except for the case of 8 Gaussian mixtures. We believe that this may be a consequence of the fact that the amount of training data in the RM corpus isn't enough to ensure that components within the 8-Gaussian mixture can be fully trained. We expect to obtain better recognition accuracy especially in the 8-Gaussian mixture case if there are more training data available.

While we made the assumption that the partitions of the training data in the original and transformed feature spaces are can relax this assumption using an iterative procedure, which partitions the training data according to the model previous iteration. We believe that this iterative further improve the performance of our new features.

Although our algorithm is similar in some ways to methds such as Semi-Tied Covariance Matrices [7] or Maximum Likelihood Linear Transformation (MLLT) [8], it is different from these methods in some ways as well. To improve the likelihood of the true classes in the training data, most of those existing form either the feature vector or model parameters individually with the dimension usually unchanged before and after the transformation. In our method, we try to find a transformation matrix that transform the data from the original feature space (such as log-spectral features) to a new feature space generally with a reduced dimensionality to improve the normalized acoustic likelihood of the true recognition classes in the transformed space. Because of the change of dimension during the transformation, our algorithm must consider the effect of transformation both on the feature vectors and model parameters simultaneously. Since the transformation matrix *A* is a non-square matrix, its transformation effect will not be cancelled out when we compute the normalized acoustic likelihood in the new feature space as in Eq. (6). We then find the transformation matrix which maximize the normalized acoustic likelihood with the transformed feature vectors and the model parameters.

We also note that we can combine our new feature generation method with existing model parameter transformation methods (*e.g.* [7][8]). Once we generate the new feature space using our method, we can apply those model parameter transformation methods in the newly generated feature space.

ACKNOWLEDGEMENTS

This research was sponsored by the Space and Naval Warfare Systems Center, San Diego, under Grant No. N66001-99-1-8905. The content of the information in this publication does not necessarily reflect the position or the policy of the US no official endorsement should be inferred.

REFERENCES

[1] Davis, S. B., and Mermelstein, P. *"*Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*" IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4): 357-366, Aug. 1980.

[2] Jolliffe, I. T. *Principal Component Analysis,* Springer-Verlag, New York, 1986

[3] Hunt, M. J, Richardson, S. M., Bateman, D. C, Piau, A. "An Investigation of PLP and IMELDA Acoustic Representations and of their Potential for Combination*," Proc. ICASSP-91*, Toronto, pp. 881-884, May 1991.

[4] Duda, R. O, Hart, P. E., and Stork, D. G. *Pattern Classification,* 2nd Edition, John Wiley & Sons, Inc., 2001.

[5] Li, X. and Stern, R. M. "Feature generation based on maximum classification probability for improved speech recognition", *Proc. Eurospeech2003,* Geneva, pp. 845-848, September, 2003.

[6] Gillick, L. and Cox, S. J. *"*Some statistical issues in the comparison of speech recognition algorithms*", Proc. ICASSP-89*, Glasgow, pp. 532-535, June 1989

[7] Gales, M. J. F. "Semi-tied covariance matrices for hidden markov models", *IEEE Transactions on Acoustics, Speech and Signal Processing,* 7(3): 272-281, 1999

[8] Gales, M. J. F. "Maximum likelihood linear transformations for HMM-based speech recognition", *Computer Speech and Language,* 12(2): 75-98, 1998

APPENDIX

$\nabla_A LogP(X_i'|C_j)$ is an *M* by *N* matrix, where *N* and *M* are the dimensions before and after the transformation. $\nabla a_{k,p}$, which is the component in the k^{th} row and p^{th} column within the matrix, can be written as Eq (A1), where $x'_{i,k}$, $\mu'_{j,k}$ and $\sigma'^2_{j,k}$ are the individual components of the transformed features and their mean and variance. $x_{i,p}$, $\mu_{j,p}$ and $\sigma^2_{j,p,q}$ are the corresponding components before transformation. (The covariance matrix of state *j* before transformation is a full matrix.) $a_{k,q}$ is the $(k,q)^{th}$ component of transformation matrix *A*.

$$\nabla a_{k,p} = -\frac{(x'_{i,k}-\mu'_{j,k})(x_{i,p}-\mu_{j,p})\sigma'^2_{j,k} - (x'_{i,k}-\mu'_{j,k})^2\left(\sum_{q=1}^{N} a_{k,q}\sigma^2_{j,p,q}\right)}{(\sigma'^2_{j,k})^2} - \frac{\left(\sum_{q=1}^{N} a_{k,q}\sigma^2_{j,p,q}\right)}{\sigma'^2_{j,k}} \qquad \text{(A1)}$$