FRONT-END FEATURE TRANSFORMS WITH CONTEXT FILTERING FOR SPEAKER ADAPTATION

Jing Huang[1], Karthik Visweswariah[2], Peder Olsen[1], Vaibhava Goel[1]

[1]IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
[2]IBM India Research Lab
Bangalore, India

ABSTRACT

Feature-space transforms such as feature-space maximum likelihood linear regression (FMLLR) are very effective speaker adaptation technique, especially on mismatched test data. In this study, we extend the full-rank square matrix of FMLLR to a non-square matrix that uses neighboring feature vectors in estimating the adapted central feature vector. Through optimizing an appropriate objective function we aim to filter out and transform features through the correlation of the feature context. We compare to FMLLR that just consider the current feature vector only. Our experiments are conducted on the automobile data with different speed conditions. Results show that context filtering improves 23% on word error rate over conventional FMLLR on noisy 60mph data with adapted ML model, and 7%/9% improvement over the discriminatively trained FMMI/BMMI models.

Index Terms— Feature-space transforms, feature-space maximum likelihood linear regression, context filtering

1. INTRODUCTION

Speaker adaptation techniques are commonly used to improve recognition accuracy on mismatched test data. Common front-end adaptation techniques include linear feature-space transforms ([1], [2],[3]) and non-linear transforms ([4],[5],[6]). Among these techniques feature-space maximum likelihood linear regression [1] (FMLLR, also known as constrained MLLR) is the most widely used and proven effetive technique.

The original formulation of FMLLR [1] was based on diagonal covariances for Gaussian mixture models and solved iteratively using mean and variance sufficient statistics. [7] proposed Quick FMLLR (Q-FMLLR) to save computation on the statistics accumulation while preserving adaptation performance of FMLLR. The Q-FMLLR would be very useful for resource contrained systems such as embedded applications. [8] and [9] extended FMLLR to the full covariance Gaussians.

In this paper we extend the full-rank square transformation matrix of FMLLR to a non-square matrix that use neighboring feature vectors in estimating the adapted central feature vector. The idea is inspired by extended maximum likelihood linear transform [10] (EMLLT) which extends MLLT that models the precision matrix (inverse of covariance matrix) in the form $A\Lambda_g A^T$ from A being square to A being non-square matrix. By finding the right compensation term in the auxiliary function of FMLLR we find the transform matrix that takes into account of the correlation of the feature context. In noisy test data this kind of feature transforms effectively smoothes the noisy missing features with its neighboring frames. From our experiments on noisy 60mph automobile data, the proposed maximum likelihood context filtering (MLCF) improves 23% relative over FMLLR on a ML model, and above 7% relative over FMLLR on BMMI/FMMI trained models.

The outline of the rest of the paper is as follows: baseline FMLLR is briefly reviewed and the proposed maximum likelihood context filtering (MLCF) is formulated in Section 2. The experimental setup and results are reported in Section 3, and conclusions are drawn in Section 4.

2. MAXIMUM LIKELIHOOD CONTEXT FILTERING

2.1. FMLLR

In the original formulation of FMLLR [1], features are adapted for a given speaker through an affine transformation

$$y_t = Ax_t + b = W\xi_t \tag{1}$$

where $\xi_t = \begin{bmatrix} x_t \\ 1 \end{bmatrix}$ is the input feature x_t at time t extended with one dimension of unity value, W is the extended transformation matrix $[A \quad b]$ with bias term b and transform A.

The objective function to optimize per speaker consists of the log likelihood of the transformed data given the current model, plus the Jacobian compensation term given by:

$$Q(W) = T \log\det(A) - \frac{1}{2}\sum_{j=1}^{N}\sum_{t=1}^{T}\gamma_t(j)(W\xi_t - \mu_j)^T\Sigma_j^{-1}(W\xi_t - \mu_j) \quad (2)$$

where T is the number of frames of data, j is the index of Gaussian components, and $\gamma_t(j)$ are the Gaussian occupation probabilities. Using sufficient statistics the transform matrix W can be estimated through an iterative row-by-row update procedure described in [1].

2.2. Extend A to non-square matrix

Instead of just using current input feature x_t, we concatenate it with its neighboring frames to make a context vector, for example, $\hat{x}_t = [x_{t-1}\ \ x_t\ \ x_{t+1}]$. We apply an affine transform to the input $\hat{x}_t$:

$$y_t = A\hat{x}_t + b = W\xi_t \quad (3)$$

Where the size of A is now $n \times 3n$, and n is the feature dimension at each time frame. We follow the FMLLR formulation above to extend the square matrix A to non-square case. First we derive the compensation term when A is a square matrix.

Let's simply assume $y = Ax$, ignoring the bias term b now, and A is full-rank invertible matrix. The loglikelihood value for feature x would be:

$$L_x = -\frac{1}{2}(x-\mu_x)^T\Sigma_x^{-1}(x-\mu_x) - \frac{1}{2}\log\det(\Sigma_x) \quad (4)$$

Here some constants are ignored. After transform A, loglikelihood of feature y would be:

$$L_y = -\frac{1}{2}(y-\mu_y)^T\Sigma_y^{-1}(y-\mu_y) - \frac{1}{2}\log\det(\Sigma_y) + \mathcal{C} \quad (5)$$

Set $L_x = L_y$, we have the compensation term $\mathcal{C} = \frac{1}{2}\log\frac{\det(\Sigma_y)}{\det(\Sigma_x)}$. This is because

$$(y-\mu_y)^T\Sigma_y^{-1}(y-\mu_y) = (x-\mu_x)^T A^T(A\Sigma_x A^T)^{-1}A(x-\mu_x) \quad (6)$$

When A is square and invertible, the above term simplifies to $(x-\mu_x)^T\Sigma_x^{-1}(x-\mu_x)$. Thus, the compensation term $\mathcal{C} = \frac{1}{2}\log\frac{\det(\Sigma_y)}{\det(\Sigma_x)} = \log\det(A)$ because $\Sigma_y = A\Sigma_x A^T$.

Now when A is not square, we assume the compensation term remains the same. We can drop out $\log\det(\Sigma_x)$ because it does not depend on A. By replacing the first term in (2), the objective function becomes:

$$Q(W) = \frac{1}{2}T\log\det(A\Sigma_x A^T) - \frac{1}{2}\sum_{j=1}^{N}\sum_{t=1}^{T}\gamma_t(j)(W\xi_t - \mu_j)^T\Sigma_j^{-1}(W\xi_t - \mu_j) \quad (7)$$

The first term is a replacement of the Jacobian term when A is not a square matrix, and its gradient is:

$$\frac{\partial\log\det(A\Sigma_x A^T)}{\partial A} = 2(A\Sigma_x A^T)^{-1}A\Sigma_x \quad (8)$$

Thus the gradient of objective function on the extended matrix W is:

$$\frac{\partial Q}{\partial W} = T \times [(A\Sigma_x A^T)^{-1}A\Sigma_x,\ \ \mathbf{0}] - \sum_{j=1}^{N}\sum_{t=1}^{T}\gamma_t(j)\Sigma_j^{-1}(W\xi_t - \mu_j)\xi_t^T \quad (9)$$

where $\mathbf{0}$ is zero vector of size x_t.

We simplify the computation of objective function and its gradient as follows: split the second term in (7) into

$$\frac{1}{2}\sum_{j=1}^{N}\sum_{t=1}^{T}\gamma_t(j)(\xi_t^T W^T\Sigma_j^{-1}W\xi_t - 2\xi_t^T W^T\Sigma_j^{-1}\mu_j + \mu_j^T\Sigma_j^{-1}\mu_j)$$

We can drop the third term since it is a constant that does not depend on W. Using $tr(AB) = tr(BA)$, we have

$$\frac{1}{2}\sum_{j=1}^{N}\sum_{t=1}^{T}\gamma_t(j)(\xi_t^T W^T\Sigma_j^{-1}W\xi_t - 2\xi_t^T W^T\Sigma_j^{-1}\mu_j)$$

$$= \frac{1}{2}\sum_{j=1}^{N}\sum_{t=1}^{T}\gamma_t(j)(tr(W^T\Sigma_j^{-1}W\xi_t\xi_t^T) - 2tr(W^T\Sigma_j^{-1}\mu_j\xi_t^T))$$

By using the mean and variance statistics K and G_i below (as defined in [1]), the above second term is $-2tr(W^T K)$.

$$K = \sum_{j=1}^{N}\sum_{t=1}^{T}\gamma_t(j)\Sigma_j^{-1}\mu_j\xi_t^T \quad (10)$$

$$G_i = \sum_{j=1}^{N}\sum_{t=1}^{T}\gamma_t(j)\sigma_{j,i}^{-1}\xi_t\xi_t^T \quad (11)$$

The first term can be simplified as follows:

$$\frac{1}{2}\sum_{j=1}^{N}\sum_{t=1}^{T}tr(W^T\Sigma_j^{-1}W\xi_t\xi_t^T) = \frac{1}{2}\sum_{i=1}^{n}tr(W^T E_{i,i}WG_i)$$

Using the fact that Σ_i^{-1} is diagonal, and $E_{i,i}$ is a unit matrix with only one nonzero element at position (i,i). To summarize, the objective function for context filtering and its gradient (constants are dropped by convenience) are therefore

$$Q(W) = \frac{1}{2}T\log\det(A\Sigma_x A^T) + tr(W^T K) - \frac{1}{2}(\sum_{j=1}^{n}tr(W^T E_{j,j}WG_j) \quad (12)$$

$$\frac{\partial Q}{\partial W} = T \times [(A\Sigma_x A^T)^{-1}A\Sigma_x,\ \ \mathbf{0}] + K - (\sum_{j=1}^{n}E_{j,j}WG_j) \quad (13)$$

The row-by-row iterative update algorithm in [1] cannot be applied here because that algorithm uses the fact that the determinant of a square matrix equals the dot product of any given row of the matrix with the corresponding row of cofactors. It is not obvious how to extend this algorithm to non-square matrices. We therefore use the limited memory BFGS algorithm along with line search as implemented in the open source package [11] to solve our maximization problem. The limited memory BFGS algorithm finds search directions by trying to approximate the Hessian. All that is required is to evaluate the objective function and its gradient, which are provided by (12) and (13). A maximum number of iterations is set to control the BFGS optimization module. Since we maximize the likelihood objective function on context input vectors, we named our method maximum likelihood context filtering (MLCF).

3. EXPERIMENTS

3.1. Experimental Setup

The experiments reported in this paper were performed on an automobile database [12]. The test data consists of utterances recorded in cars at three different speeds: 0mph (idling), 30mph and 60mph. The average signal-to-noise ratio (SNR) is 21/17/12 for 0mph/30mph/60mph. Figure 1 shows the SNR distribution for each speed condition. Four tasks are included in the test set: addresses, digits, commands and radio control (ADCR) with total about 26K utterances and 130K words.

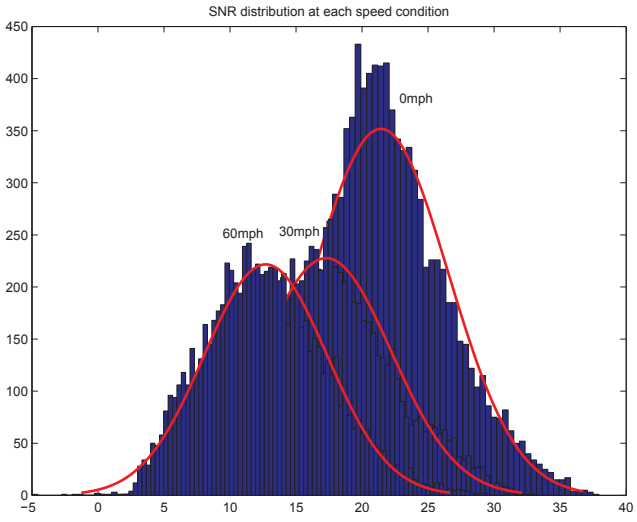

Fig. 1. SNR distribution for 0mph/30mph/60mph data.

Most of the training data was collected in stationary cars with a total of 800K training utterances. The baseline maximum likelihood (ML) acoustic model is word-internal with pentaphone context, with 830 context-dependent states and 10K Gaussians. This small model is intended for embedded speech engines in the car. The front-end features are 13-dimensional cepstra with LDA transform. The model-space discriminative model BMMI and feature-space discriminative model FMMI [13] are also tested in our experiments. Both FMLLR and MLCF are estimated per speaker from decoded output of the baseline models. Each speaker has 100 utterances. One obvious disadvantage of MLCF compared to FMLLR is that it has more parameters and thus needs more adaptation data than FMLLR. We present the effect of adaptation data amount on both FMLLR and MLCF in Table 2.

3.2. Experimental Results

Results are presented as word error rate (WER) and sentence error rate (SER) for each speed condition. Table 1 compares the performance of FMLLR with MLCF adapted on the ML model. Compared to the ML baseline, FMLLR improves WER on each speed: 26%/27%/29% relative on 0mph/30mph/60mph. However, MLCF improves upon more compared to FMLLR. MLCF-3 means the input context vector consists of 3 consecutive frames; MLCF-5 means the input context vector consists of 5 consecutive frames. The non-square transforms are initialized with zero matrices for all the frames and the identity matrix for the central frame, while MLCF-3-init uses FMLLR as the starting point for the central frame.

As we can see MLCF-3 and MLCF-5 both give a tiny gain over FMLLR on the 0mph condition. On the 30mph test data MLCF-3 degrades a little over FMLLR, while MLCF-5 gains a little over FMLLR. However, on the most noisy 60mph test data, MLCF-3 improves significantly upon FMLLR: 21%/12% relative on WER/SER, and MLCF-5 improves even more: 23%/13% relative on WER/SER over FMLLR. Starting with FMLLR for the center frame does not provide any advantage over the identity matrix. The optimization module seems to converge to the same point.

Table 2 shows the performance of FMLLR and MLCF-3 on 60mph data with different amount of adaptation data: 10-utterance, 30-utterance, 60-utterance and all utterances. As the amount of adaptation data increases, the performance of both FMLLR and MLCF get better. However, in the 10-utterance case, MLCF-3 gains even more over FMLLR (almost 30% relative) than in the all-utterance case (23%). The small amount of adaptation data seems to affect FMLLR more than MLCF: for FMLLR there is 15% degradation from all-utterance to 10-utterance, while for MLCF-3, only 7% relative degradation.

Since using 5-frame context only gains little over 3-frame context, we leave it out for the BMMI/FMMI models. Table 3 compares FMLLR and MLCF adapted on a BMMI model directly trained on top of the ML model. The trend is similar to the case of ML model: tiny improvement over FMLLR on 0mph/30mph, 9%/11% relative improvement of WER/SER over FMLLR on the noisy 60mph data. Again starting with FMLLR transform for the central frame does not provide any advantage over the identity transform. However, this is not the case for FMMI model. Table 4 shows that with FMMI model MLCF-3-init is better than MLCF-3, and gains 7%/9% relative on WER/SER over FMLLR. In addition, more iterations of optimization hurts the performance. This may due to the fact the FMMI transform is applied before MLCF, and their objective functions are not the same: the first is MMI, and the later is ML. More iterations only move MLCF further away from FMMI, which degrades the performance. Future work will use discriminative objective function [3] and check how context-filtering interacts with FMMI transform.

4. CONCLUSIONS

In this paper, we extend the full-rank square matrix of FMLLR to a non-square matrix that use neighboring feature vectors in estimating the adapted central feature vector.

WER/SER	0mph	30mph	60mph
baseline	0.77/3.34	1.28/5.15	2.65/8.94
FMLLR	0.57/2.42	0.94/3.82	1.87/6.29
MLCF-3	0.54/2.31	0.95/3.91	**1.48/5.54**
MLCF-3-init	0.54/2.32	0.96/3.89	1.50/5.58
MLCF-5	0.55/2.41	0.93/3.84	**1.44/5.49**

Table 1. Comparison of FMLLR and MLCF adapted on the ML model.

WER/SER	10-utts	30-utts	60-utts	all-utts
FMLLR	2.20/7.27	1.93/6.60	1.93/6.49	1.87/6.29
MLCF-3	1.60/5.98	1.52/5.62	1.53/5.61	1.48/5.54

Table 2. Comparison of FMLLR and MLCF adapted on different amount of 60mph data.

Through optimizing a maximum likelihood objective function we aim to filter out and transform features through the correlation of the feature context. We compare to FMLLR that just consider the current feature vector only. Our experiments are conducted on the IBM automobile data with 0mph/30mph/60mph speed conditions. Results show that context filtering improves upto 23% on word error rate over conventional FMLLR on noisy 60mph data with adapted ML model, while provide tiny gains on 0mph/30mph data. The gains over FMLLR are smaller (7%/9%) on the discriminatively trained FMMI/BMMI models. Future work includes using discriminative objective function or smoothing of discriminative and maximum likelihood objective functions.

5. REFERENCES

[1] M.J.F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer speech and language, 1998*, 1998.

[2] George Saon, Geoffrey Zweig, and Mukund Padmanabhan, "Linear feature space projections for speaker adaptation," in *Proc: ICASSP 2001*, 2001.

[3] L. Wang and P.C. Woodland, "Discriminative adaptive training using the MPE criterion," in *Proc: ASRU 2003*. IEEE, 2003.

[4] Karthik Visweswariah and Ramesh Gopinath, "Adaptation of front end parameters in a speech recognizer," in *Proc: ICSLP 2004*, 2004.

[5] P. Olsen, S. Axelrod, K. Visweswariah, and R. Gopinath, "Gaussian mixture modeling with volume preserving nonlinear feature space transforms," in *Proc: ASRU 2003*. IEEE, 2003.

WER/SER	0mph	30mph	60mph
baseline	0.63/2.76	0.96/3.82	2.02/6.95
FMLLR	0.46/1.98	0.75/3.06	1.47/5.24
MLCF-3	0.45/1.91	0.74/3.05	**1.33/4.68**
MLCF-3-init	0.43/1.86	0.74/3.04	1.33/4.77

Table 3. Comparison of FMLLR and MLCF adapted on the BMMI model.

WER/SER	0mph	30mph	60mph
baseline	0.45/1.90	0.76/3.23	1.30/5.05
FMLLR	0.33/1.40	0.60/2.52	1.00/4.06
MLCF-3	0.32/1.31	0.61/2.56	0.96/3.84
MLCF-3-init	0.32/1.34	0.59/2.47	**0.93/3.75**

Table 4. Comparison of FMLLR and MLCF adapted on the FMMI model.

[6] George Saon, Satya Dharanipragada, and Daniel Povey, "Feature space gaussianization," in *Proc: ICASSP 2004*, 2004.

[7] Balakrishnan Varadarajan, Daniel Povey, and Stephen Chu, "Quick FMLLR for speaker adaptation in speech recognition," in *Proc: ICASSP 2008*. IEEE, 2008.

[8] Daniel Povey and George Saon, "Feature and model space speaker adaptation with full covariance gaussians," in *Proc: ICSLP, 2006*, 2006.

[9] Arnab Ghoshal, Daniel Povey, and et. al., "A novel estimation of feature-space MLLR for full-covariance models," in *Peoc: ICASSP 2008*. IEEE, 2008.

[10] Peder Olsen and Ramesh Gopinath, "Modeling inverse covariance matrices by basis expansion," *IEEE Trans. on Speech and Audio Processing, 2004*, 2004.

[11] M.S. Gockenbach and W.W. Symes, "The Hilbert class library," www.trip.caam.rice.edu/software/hcldoc/html/index.html.

[12] S. Deligne, S. Dharanipragada, R. Gopinath, B. Maison, P. Olsen, and Printz H., "A robust high accuracy speech recognition system for mobile applications," *IEEE Transactions on Speech and Audio Processing, 2002*, 2002.

[13] 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*. IEEE, 2008.