TEMPORAL FILTER DESIGN BY MINIMUM KL DIVERGENCE CRITERION FOR ROBUST SPEECH RECOGNITION

Xiong Xiao[1], *Eng Siong Chng*[1,2], *Haizhou Li*[1,2,3]

[1]Temasek Lab@NTU, Nanyang Technological University, Singapore
[2]School of Computer Engineering, Nanyang Technological University, Singapore
[3]Department of Human Language Technology, Institute for Infocomm Research, Singapore
xiaoxiong@ntu.edu.sg, aseschng@ntu.edu.sg, hli@i2r.a-star.edu.sg

ABSTRACT

In this paper, we propose a new temporal filter design method based on minimum KL divergence criterion for robust recognition of noisy and reverberant speech. The main idea is to optimize the filter parameters by minimizing the KL divergence of two distributions, of which one is the feature distribution in the test environment, and another is the feature distribution represented by the acoustic model. The minimization of the KL divergence reduces the mismatch between the acoustic model and the test data. Experimental results on Aurora-5 task shows that the new filter design outperforms other filter design methods significantly in noisy and reverberant test conditions. In addition, the proposed filtering of feature trajectories is shown to be complementary to linear transformation of feature vectors, which is popular in feature processing.

Index Terms— Temporal filtering, feature transform, robust ASR, reverberation, KL divergence.

1. INTRODUCTION

Robust speech recognition in adverse environments is one of the challenges in automatic speech recognition (ASR) research. ASR systems trained on clean speech data usually perform poorly on noisy and/or reverberant speech. There are two major approaches to improve the robustness of ASR. One is called feature space approach (e.g. [1, 2, 3]) which reduces noise and reverberation effects in the speech features before the recognition, and the other is called model adaptation approach (e.g. [4, 5, 6]) which focuses on adapting the acoustic model's parameters to fit the test environment better. Although model space techniques are usually more powerful, feature space techniques are more efficient and easier to be integrated to existing ASR systems, as they usually do not require modification to the acoustic model and decoder.

There are two major ways to process speech features for robust speech recognition, one is the linear transformation of individual feature vectors, and the other is the temporal filtering of feature trajectories. A widely used linear transform is the constrained maximum likelihood linear regression (CMLLR) method [4]. Although CMLLR is a model space technique, it is often implemented as feature space transforms for better efficiency [4, 7]. It is noted that CMLLR is effective in reducing additive noises effects and speaker variations. Examples of temporal filters include RASTA [8], MVA processing [9], data-driven filters [10], and temporal structure normalization (TSN) [11]. Compared to linear transformation of individual feature vectors, one advantage of temporal filters is their ability to modify the temporal characteristics of feature trajectories. Therefore, temporal filters may be more suitable to deal with distortions that affect the long term temporal structure of features, e.g. reverberation. Temporal filters are reported to be effective in dealing with both additive noise [8, 10, 11] and reverberation [12, 13, 14].

A limitation of the conventional temporal filters is that their design criteria do not make use of the clean feature distribution captured by acoustic model. Recently, clean feature distribution is used in the optimization of temporal filters. In [13, 14], a Gaussian mixture model (GMM) trained from clean features is used to guide the filter estimation by using a maximum likelihood (ML) criterion. Unfortunately this method is not well grounded mathematically, i.e. the scaling of feature space is not considered in the ML criterion. In [14], this problem is alleviated by normalizing the variances of filtered features. In [13], a temporal normalization term similar to TSN is used as regularization. However, both approaches are not optimal.

In this paper, we consider the design of temporal filters by using rich clean feature distribution represented by the acoustic model. To address the feature scaling problem in ML estimation, we propose a new objective function which aims at minimizing KL divergence. The filter designed from the new criterion is called maximum normalized likelihood linear filtering (MNLLF). We will also study whether temporal filter MNLLF is complementary to linear transform CMLLR as they are different ways of linear feature processing.

The paper is organized as follows. In section 2, the KL divergence based filter design is proposed. In section 3, the proposed filter is compared with other filters. In addition, the interactions between linear filtering and linear transformation is studied. Finally, we conclude in section 4.

2. FILTER DESIGN BY MINIMUM KL DIVERGENCE

2.1. Background

Assume that we have an acoustic model with parameters Λ_m which are trained from clean speech data. The feature distribution represented by the model is $p_m(\mathbf{x}|\Lambda_m)$, where the subscript m denotes 'model distribution' and $\mathbf{x}$ is a feature vector. In many applications, we want to use the model to recognize utterances from noisy acoustic conditions. The recognition performance is poor as the distribution $p_m(\mathbf{x}|\Lambda_m)$ learnt from clean data does not represent the feature distribution of the test environment well. Such problem is called training-test mismatch problem. To reduce the mismatch, we can either move the features closer to the acoustic model (by using feature space techniques), or move the acoustic model closer to the test features (by using model space techniques), or both. In this paper, we focus on adapting the features towards the acoustic model.

2.2. Problem of Maximum Likelihood Criterion for Feature Adaptation

A simple criterion of adapting the features to fit the acoustic model is to maximize the likelihood of the adapted features on the model. Let $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_T]^T$ where $\mathbf{x}_t$ is the D dimensional test feature vector for frame t and T is the number of test feature vectors. The adapted features will be $\mathbf{Y} = f(\mathbf{X})$, where $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_T]^T$ and $f(\cdot)$ represents the transformation function. The log likelihood of the adapted features on the model is $\log p_m(\mathbf{Y}|\Lambda_m) = \sum_{t=1}^{T} \log(\mathbf{y}_t|\Lambda_m)$ if we assume that the acoustic model does not consider the correlation between frames. The adaptation based on maximizing $\log p_m(\mathbf{Y}|\Lambda_m)$ w.r.t. $f(\cdot)$ is problematic. Our experimental study shows that the adapted features usually have much smaller variances than the original features, leading to very bad recognition performance. This is because the scaling of feature space by the transformation $f(\cdot)$ is not taken into consideration in the likelihood function $\log p_m(\mathbf{Y}|\Lambda_m)$. In previous feature space processing methods that also maximize $\log p_m(\mathbf{Y}|\Lambda_m)$, this problem is alleviated by either renormalizing the adapted features' variances [14] or using a regularization term [15][13]. In this paper, we will propose a new filter design criterion that takes into consideration the feature scaling.

2.3. Minimum KL Divergence Criterion for Feature Adaptation

To address the problem of the pure ML criterion for feature adaptation, we propose a new design criterion based on the concept of minimizing the KL divergence between the acoustic model distribution and the test data's distribution.

Let's assume that $p_{\mathbf{x}}(\mathbf{x})$ is the unobserved feature distribution for the test environment. The mismatch problem can be explained as the difference between the model distribution $p_m(\mathbf{x}|\Lambda_m)$ and the test data distribution $p_{\mathbf{x}}(\mathbf{x})$. To minimize the mismatch, we can minimize the Kullback-Leibler divergence (KL divergence) between the two distributions. One definition of the KL divergence is [16]:

$$D(p_{\mathbf{x}}||p_m) = \int p_{\mathbf{x}}(\mathbf{x}) \log \frac{p_{\mathbf{x}}(\mathbf{x})}{p_m(\mathbf{x}|\Lambda_m)} d\mathbf{x} \tag{1}$$

where the integration is evaluated over the test feature space. When the KL divergence is minimized (through either feature processing or model adaptation), the model and data distributions will be closer to each other, and the model-data mismatch will be reduced.

In practice, the data distribution is not observed completely. What we observe is samples from the data distribution, i.e. T feature vectors of test utterances $\mathbf{X}$. Hence, we approximates the KL divergence in (1) as

$$\hat{D}(p_{\mathbf{x}}||p_m) = \sum_{t=1}^{T} \log \frac{p_{\mathbf{x}}(\mathbf{x}_t)}{p_m(\mathbf{x}_t|\Lambda_m)} \tag{2}$$

Note that there are two major differences between the KL divergence in (1) and its approximation in (2). One is that the integration over the test feature space is changed to summation over test feature vectors. Using the summation to approximate the integration is reasonable if we assume that the test feature vectors in $\mathbf{X}$ are faithfully sampled from the data distribution $p_{\mathbf{x}}(\mathbf{x})$. Another difference is that the first $p_{\mathbf{x}}(\mathbf{x})$ term on the right of (1) disappears in (2). Again, this is because if the test samples are drawn from $p_{\mathbf{x}}(\mathbf{x})$, the summation implicitly takes into account the first $p_{\mathbf{x}}(\mathbf{x})$ term in (1). That is, there are more test samples drawn from the region in the test feature space where $p_{\mathbf{x}}(\mathbf{x})$ is high and vice versa. The equation (2) can also be seen as a Monte Carlo [17] simulation of the KL divergence where the random samples are the test feature vectors.

There are two ways to minimize the approximated KL divergence in (2). One way is to modify the model parameters Λ_m to increase $p_m(\mathbf{x}_t|\Lambda_m)$. As $p_{\mathbf{x}}(\mathbf{x}_t)$ is independent of Λ_m, maximizing $p_m(\mathbf{x}_t|\Lambda_m)$ by tuning the acoustic model parameters will guarantee the minimization of the approximated KL divergence. This is actually the case of many acoustic model adaptation techniques, such as CMLLR [4]. Another way to minimize the approximated KL divergence is to adapt the test features to minimize the following function:

$$\hat{D}(p_{\mathbf{y}}||p_m) = \sum_{t=1}^{T} \log p_{\mathbf{y}}(\mathbf{y}_t) - \sum_{t=1}^{T} \log p_m(\mathbf{y}_t|\Lambda_m) \tag{3}$$

where $p_{\mathbf{y}}(\mathbf{y}_t)$ is the probability distribution of the processed features and is different from $p_{\mathbf{x}}(\mathbf{x}_t)$ as the features are processed. The minimization of equation (3) will increase the likelihood of the processed features evaluated on the acoustic model, i.e. $p_m(\mathbf{y}_t|\Lambda_m)$. At the same time, it will prevent the likelihood on data distribution $p_{\mathbf{y}}(\mathbf{y}_t)$ from increasing too much. It will be clear in the following sections that by using (3) for filter design, the scaling of feature space is taken into consideration.

2.4. Feature Adaptation by Filtering Feature Trajectories

In this paper, we study the temporal filtering of feature trajectories. The filtered feature vector is obtained as:

$$y_t^{(d)} = \sum_{\tau=-L}^{L} w_\tau^{(d)} x_{t+\tau}^{(d)} = \mathbf{w}^{(d)} \tilde{\mathbf{x}}_t^{(d)} \tag{4}$$

where $\mathbf{w}^{(d)} = [w_{-L}^{(d)}, ..., w_0^{(d)}, ..., w_L^{(d)}]$ is a $1 \times (2L+1)$ weight vector for the d^{th} feature dimension. $\tilde{\mathbf{x}}_t^{(d)} = [x_{t-L}^{(d)}, ..., x_t^{(d)}, ...x_{t+L}^{(d)}]^T$ is the input of the linear filter at frame t and dimension d. The filtered feature vector at frame t is defined as $\mathbf{y}_t = [y_t^{(1)}, ..., y_t^{(D)}]^T$.

2.5. Filter Weights Estimation

The weight vectors will be obtained by minimizing the cost function in (3). There are two terms in (3), one is the data likelihood $p_{\mathbf{y}}(\mathbf{y}_t)$ and the other is the acoustic model likelihood $p_m(\mathbf{y}_t|\Lambda_m)$. The calculation of the acoustic model likelihood is straightforward, while the calculation of the data likelihood is not so easy as we don't have $p_{\mathbf{y}}(\mathbf{y}_t)$. In this paper, we will approximate $p_{\mathbf{y}}(\mathbf{y}_t)$ by a single Gaussian and estimate its mean and variances from the test data.

Assume the processed feature distribution $p_{\mathbf{y}}(\mathbf{y}_t)$ can be approximated by a single Gaussian with diagonal covariance matrix as we are working on weakly correlated cepstral features. Then we have

$$\log p_{\mathbf{y}}(\mathbf{Y}) = K - \frac{1}{2} \sum_{t=1}^{T} \sum_{d=1}^{D} \left(\log(\sigma_y^{(d)})^2 - \frac{(y_t^{(d)} - \mu_y^{(d)})^2}{(\sigma_y^{(d)})^2} \right) \tag{5}$$

where K is a constant term not related to the filter weights, $\mu_y^{(d)}$ and $(\sigma_y^{(d)})^2$ are the mean and variance of the Gaussian for dimension d, respectively. If we estimate the mean and variance of $\mathbf{y}_t$ from the test data, i.e. $(\sigma_y^{(d)})^2 = \sum_{t=1}^{T} (y_t^{(d)} - \mu_y^{(d)})^2 / T$, it can be easily seen that the term $\sum_{t=1}^{T} \frac{(y_t^{(d)} - \mu_y^{(d)})^2}{(\sigma_y^{(d)})^2} = T$ is a constant. Therefore, the data log likelihood function is only a function of the variances of the filtered features:

$$\log p_{\mathbf{y}}(\mathbf{Y}) = K' - \frac{1}{2} \sum_{t=1}^{T} \sum_{d=1}^{D} \log(\sigma_y^{(d)})^2 \tag{6}$$

Substitute (6) into (3), the objective function can be rewritten as

$$\hat{D}(p_d||p_m) = K' - \frac{T}{2}\sum_{d=1}^{D}\log(\sigma_y^{(d)})^2 - \sum_{t=1}^{T}\log p_m(\mathbf{y}_t|\Lambda_m) \quad (7)$$

From (7), it is clear that to minimize the KL divergence between the data and model distributions, one should increase not only the likelihood of the processed features on the acoustic model, i.e. $p_m(\mathbf{y}_t|\Lambda_m)$, but also the variance of the processed features. Note that this finding is not limited to temporal filtering, but also applicable to any form of feature space transformation.

Motivated by (7), the filter weights can be estimated as

$$\begin{aligned}\hat{\mathbf{W}} &= \arg\max_{\mathbf{W}}\Big\{\frac{\lambda T}{2}\sum_{d=1}^{D}\log(\sigma_y^{(d)})^2 - \frac{\beta}{2}\sum_{d=1}^{D}|\mathbf{w}^{(d)}-\mathbf{w}_0^{(d)}|^2 \\ &\quad + \sum_{t=1}^{T}\log p_m(\mathbf{y}_t|\Lambda_m)\Big\} \quad (8)\end{aligned}$$

where $\mathbf{W} = [(\mathbf{w}^{(1)})^T, ..., (\mathbf{w}^{(D)})^T]^T$ is the matrix of filter weights for all feature dimensions. The L2 norm is for regularizing the weight estimation when the test utterance is very short. $\mathbf{w}_0^{(d)} = [0, ..., 0, 1, 0, ..., 0]$ is the initial weights, i.e. the weight corresponding to the current frame is set to 1 and all rest weights are set to 0. We also introduce λ to control the importance of the variance term that comes from the data distribution. This is because we are using an approximated data distribution rather than the true data distribution. Hence, the best weight for the variance term may be different from 1. We call the proposed filter maximum *normalized* likelihood linear filtering (MNLLF), as the variance term in (8) can be viewed as a normalization of the likelihood term.

2.6. Solution of MNLLF Linear Filter Weights

First, let's find the variances of the processed features. From (4), the variance $(\sigma_y^{(d)})^2$ can be estimated from the test samples as

$$(\sigma_y^{(d)})^2 = \frac{1}{T}\sum_{t=1}^{T}(y_t^{(d)} - \mu_y^{(d)})^2 = \mathbf{w}^{(d)}\mathbf{C}_{\tilde{\mathbf{x}}}^{(d)}(\mathbf{w}^{(d)})^T \quad (9)$$

where $\mathbf{C}_{\tilde{\mathbf{x}}}^{(d)} = \frac{1}{T}\sum_t(\tilde{\mathbf{x}}_t^{(d)} - \tilde{\boldsymbol{\mu}}_x^{(d)})(\tilde{\mathbf{x}}_t^{(d)} - \tilde{\boldsymbol{\mu}}_x^{(d)})^T$ and $\tilde{\boldsymbol{\mu}}_x^{(d)} = \frac{1}{T}\sum_t \tilde{\mathbf{x}}_t^{(d)}$ are the estimated covariance matrix and mean of original features across frames for feature dimension d. It is clear that we will need temporal information of the original features, represented by the cross-frame covariance $\mathbf{C}_{\tilde{\mathbf{x}}}^{(d)}$, for estimating the variances of filtered features.

To compute the acoustic model likelihood, we can either use the hidden Markov model (HMM) of the acoustic model, or simply a GMM. In this paper, we will use a GMM with M mixtures for simplicity. The acoustic model likelihood score is:

$$\log p_m(\mathbf{y}_t|\Lambda_m) = \log\sum_{m=1}^{M} c_m\mathcal{N}(\mathbf{y}_t;\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m) \quad (10)$$

where c_m, $\boldsymbol{\mu}_m$, and $\boldsymbol{\Sigma}_m$ are the prior weight, mean vector, and diagonal covariance matrix of the m^{th} Gaussian in the GMM.

As there is no closed form solution to the optimization problem in (8), we will use gradient ascent algorithm to find the solution of the weights iteratively. From (8-10), the gradient of $\mathbf{w}^{(d)}$ is

$$\begin{aligned}\nabla_{\mathbf{w}}^{(d)} &= \frac{\lambda T}{2}\frac{\partial\mathbf{w}^{(d)}\mathbf{C}_{\tilde{\mathbf{x}}}^{(d)}(\mathbf{w}^{(d)})^T/\partial\mathbf{w}^{(d)}}{\mathbf{w}^{(d)}\mathbf{C}_{\tilde{\mathbf{x}}}^{(d)}(\mathbf{w}^{(d)})^T} - \frac{\beta}{2}\frac{\partial|\mathbf{w}^{(d)}-\mathbf{w}_0^{(d)}|^2}{\partial\mathbf{w}^{(d)}} \\ &\quad + \sum_{t=1}^{T}\frac{\sum_{m=1}^{M}c_m\mathcal{N}(\mathbf{y}_t;\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m)}{\sum_{n=1}^{M}c_n\mathcal{N}(\mathbf{y}_t;\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_n)}\frac{\partial\mathcal{N}(\mathbf{y}_t;\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m)}{\partial\mathbf{w}^{(d)}} \\ &= \frac{\lambda T\mathbf{w}^{(d)}\mathbf{C}_{\tilde{\mathbf{x}}}^{(d)}}{(\sigma_y^{(d)})^2} - \beta(\mathbf{w}^{(d)}-\mathbf{w}_0^{(d)}) - \mathbf{w}^{(d)}\mathbf{G}^{(d)} + \mathbf{p}^{(d)} \quad (11)\end{aligned}$$

where

$$\mathbf{G}^{(d)} = \sum_{t=1}^{T}\sum_{m=1}^{M}\frac{\gamma_m(t)}{(\sigma_m^{(d)})^2}\tilde{\mathbf{x}}_t^{(d)}(\tilde{\mathbf{x}}_t^{(d)})^T \quad (12)$$

$$\mathbf{p}^{(d)} = \sum_{t=1}^{T}\sum_{m=1}^{M}\frac{\gamma_m(t)\mu_m^{(d)}}{(\sigma_m^{(d)})^2}(\tilde{\mathbf{x}}_t^{(d)})^T \quad (13)$$

and $\gamma_m(t) = \frac{c_m\mathcal{N}(\mathbf{y}_t;\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m)}{\sum_{m=1}^{M}c_m\mathcal{N}(\mathbf{y}_t;\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m)}$ is the posterior probability of mixture m given the current processed feature vector $\mathbf{y}_t$. The filter weights are updated iteratively until convergence:

$$\mathbf{w}_{i+1}^{(d)} = \mathbf{w}_i^{(d)} + \alpha\nabla_{\mathbf{w}_i}^{(d)} \quad (14)$$

where i is the iteration index, α is the learning rate, and $\nabla_{\mathbf{w}_i}^{(d)}$ is the gradient of the weight vector evaluated by using current weight estimate $\mathbf{w}_i^{(d)}$ in (11). Since the computing of $\mathbf{G}^{(d)}$ and $\mathbf{p}^{(d)}$ is computationally expensive, we only compute them once using the original features. At each iteration, only $(\sigma_y^{(d)})^2$, $\nabla_{\mathbf{w}}^{(d)}$, and $\mathbf{w}^{(d)}$ are updated according to (9), (11), and (14).

3. EXPERIMENTS

3.1. Experimental Settings

The proposed filter is evaluated on the Aurora-5 connected English digit string recognition benchmark task [18]. We focused on test cases of "living room" and "office", which are corrupted by both additive noises and reverberation. Besides these artificially generated test data, we also test on real meeting recordings, which were simultaneously recorded by 4 hands-free microphones and corrupted by reverberation and a small amount of background noise. The acoustic model training follows the standard clean-condition training configuration of Aurora-5 task. For details of these test cases and model training, please refer to [18]. The raw speech features are the 39D MFCC features, including c0-c12, and their first and second derivatives. The features of each utterance are normalized first by mean and variance normalization (MVN) [19] and then by TSN [11].

The TSN processed features are used as the input of the proposed MNLLF, and also JSTN [13] and CMLLR [4]. In both MNLLF and JSTN filters, the GMM for filter design is obtained by pooling the 716 Gaussians of the acoustic model. In CMLLR, the HMMs of the acoustic model are used together with the initial hypothesis produced by the step directly prior to CMLLR. The filter length of JSTN and MNLLF are both 33 taps, hence there are 33×39 free parameters. For CMLLR, there are 39×2 and 39×40 free parameters for the diagonal and full transforms, respectively. For MNLLF, λ and β are empirically set to 0.6 and 30, respectively. The other settings of JSTN is the same as that in [13]. Whenever MNLLF, JSTN, or CMLLR are applied, the acoustic model trained from TSN processed features are used to recognize the test sentences.

Table 1. Recognition accuracy achieved by utterance based feature processing on artificial noisy data. TSN is the preprocessing for JSTN, MNLLF, and CMLLR (diagonal transform). The column denoted as "Combine" refers to the cascade of MNLLF and CMLLR.

SNR	MVN	TSN	JSTN	MNLLF	CMLLR	Combine
Clean	**99.38**	**99.38**	99.34	99.22	99.37	99.27
Office	93.86	94.26	96.18	96.66	95.32	**96.83**
Office 15dB	81.00	83.99	88.23	89.82	85.64	**90.05**
Office 10dB	68.79	75.61	79.78	82.72	76.76	**82.78**
Office 5dB	49.89	61.65	64.60	**69.61**	61.88	69.30
Office 0dB	28.17	41.69	42.06	**48.53**	40.27	47.46
Living	81.30	83.07	88.06	89.93	85.72	**90.58**
Living 15dB	62.62	68.62	75.59	79.58	70.62	**80.02**
Living 10dB	50.87	60.37	65.64	71.14	61.37	**71.26**
Living 5dB	35.76	48.77	50.78	**57.72**	48.53	57.31
Living 0dB	21.11	33.20	32.54	**39.16**	31.83	37.97
Avg	57.34	65.84	68.35	**72.88**	66.38	72.73

Table 2. Recognition accuracy achieved by utterance based feature processing on the 4 microphone meeting data.

Mic.	MVN	TSN	JSTN	MNLLF	CMLLR	Combine
6	87.12	89.18	90.26	91.60	90.02	**91.89**
7	82.28	85.97	86.65	88.64	86.91	**89.11**
E	80.65	84.14	85.15	86.92	84.79	**86.95**
F	85.67	87.21	88.93	89.34	88.09	**89.68**
Avg	83.93	86.63	87.75	89.13	87.45	**89.41**

3.2. Utterance based processing

We first examine utterance based processing of speech features, where linear filters and linear transforms are estimated based on the information of a single utterance. The performance on artificial test data and real meeting data are shown in Table 1 and Table 2, respectively. From the two tables, it is observed that the proposed MNLLF filter consistently outperforms previous filter design methods, i.e. TSN [11] and JSTN [13], except for clean test case. This demonstrates the effectiveness of the proposed minimum KL divergence objective function for temporal filter design. In addition, the MNLLF outperforms CMLLR significantly. This is due to that: 1) for CMLLR, we are forced to use simple *diagonal* transform as the utterances are generally very short (0.5-3s); 2) CMLLR processes each frame individually and is not effective in dealing with reverberations. When we apply CMLLR after MNLLF (last column of the tables), only a small gain is obtained for high SNR levels (>5dB). This is contradictory to our expectation that MNLLF should be complementary to CMLLR as they use different information. We suspect that the small gain from combination is due to that the diagonal CMLLR transforms are too weak in per utterance processing.

3.3. Interactions between MNLLF and CMLLR in speaker based processing

To investigate the full synergy between CMLLR and MNLLF, we also study the speaker based processing of speech features, i.e. the linear filters and transforms are optimized based on the information of all utterances of a speaker. In this case, full transforms are used in CMLLR as there are enough test data from each speaker.

Table 3. Recognition accuracy achieved by speaker based feature processing on artificial noisy data.

SNR	TSN	MNLLF	CMLLR	Combine
Clean	99.38	99.35	**99.57**	**99.57**
Office	94.26	97.05	98.03	**98.54**
Office 15dB	83.99	90.49	92.88	**95.11**
Office 10dB	75.61	83.00	85.20	**88.85**
Office 5dB	61.65	68.79	69.94	**75.05**
Office 0dB	41.69	46.81	44.73	**50.22**
Living room	83.07	91.40	92.07	**95.37**
Living room 15dB	68.62	81.01	79.13	**87.38**
Living room 10dB	60.37	72.07	69.10	**77.95**
Living room 5dB	48.77	57.25	52.99	**61.35**
Living room 0dB	33.20	37.95	33.21	**38.40**
Avg	65.84	72.92	72.03	**77.00**

Table 4. Recognition accuracy achieved by speaker based feature processing on meeting data.

Mic.	TSN	MNLLF	CMLLR	Combine
6	89.18	91.40	91.85	**92.98**
7	85.97	88.41	88.71	**90.50**
E	84.14	86.33	87.30	**88.89**
F	87.21	89.52	89.70	**90.62**
Avg	86.63	88.92	89.39	**90.75**

The performance of MNLLF, CMLLR, and their combination are shown in Table 3 and Table 4. From the tables, it can be observed that full transform CMLLR produces much better results than the diagonal CMLLR transforms in Table 1 and Table 2. The full transform CMLLR achieves similar performance as MNLLF. This could be due to that CMLLR could reduce reverberation effects to a certain degree via the dynamic features which capture temporal information up to about 0.1s. On the other hand, the speaker based MNLLF does not show much improvement over utterance based MNLLF. Possible reason may be that in the speaker based MNLLF, the filters cannot be optimized for each test utterance.

When MNLLF and CMLLR are applied in sequence, significantly better performance is obtained than when they are applied alone. From the results, it is clear that MNLLF and CMLLR are complementary to each other. This is because the two methods use different information, i.e. CMLLR uses short term temporal information, while MNLLF uses long term temporal information.

4. CONCLUSIONS

In this paper, we proposed a novel temporal filter design method for robust ASR, called MNLLF. The filter is designed to reduce the training-test mismatch by minimizing the KL divergence of the two distributions, i.e. the distribution of the filtered features and the distribution of the acoustic model. Experimental results on Aurora-5 task show that MNLLF produces consistently better results than previous temporal filters, such as TSN and JSTN. It is also shown that the temporal filter MNLLF is complementary to the widely used CMLLR, which is a linear transform of feature vectors.

5. REFERENCES

[1] Á. de la Torre, A. M. Peinado, J. C. Segura, J. L. Pérez-Córdoba, M. C. Benítez, and A. J. Rubio, "Histogram equalization of speech representation for robust speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 3, pp. 355–366, 2005.

[2] L. Deng, J. Droppo, and A. Acero, "Enhancement of log mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise," *IEEE Trans. Speech and Audio Processing*, vol. 12, no. 2, pp. 133–143, Mar. 2004.

[3] A. Krueger and R. Haeb-Umbach, "Model-based feature enhancement for reverberant speech recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1692–1707, 2010.

[4] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[5] J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero, "A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions," *Computer Speech and Language*, vol. 23, no. 3, pp. 389–405, Jul. 2009.

[6] H. Hirsch and F. Harald, "A new approach for the adaptation of hmms to reverberation and background noise," *Speech Communication*, vol. 50, pp. 244–263, March 2008.

[7] G. Stemmer, F. Brugnara, and D. Giuliani, "Adaptive training using simple target models," in *Proc. ICASSP '05*, Philadelphia, USA, Mar. 2005, vol. I, pp. 997–1000.

[8] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, 1994.

[9] C.-P. Chen and J. A. Bilmes, "MVA processing of speech features," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 257–270, 2007.

[10] J.-W. Hung and L.-S. Lee, "Optimization of temporal filters for constructing robust features in speech recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 808–832, 2006.

[11] X. Xiao, E. S. Chng, and H. Li, "Normalization of the speech modulation spectra for robust speech recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1662–1674, Nov. 2008.

[12] X. Lu, M. Unoki, and S. Nakamura, "Sub-band temporal modulation envelopes and their normalization for automatic speech recognition in reverberant environments," *Computer Speech and Language*, vol. 25, no. 3, pp. 571 – 584, 2011.

[13] X. Xiao, E. S. Chng, and H. Li, "Joint spectral and temporal normalization of features for robust recognition of noisy and reverberated speech," in *Proc. ICASSP '12*, Kyoto, Japan, Apr. 2012, pp. 4325–4328.

[14] K. Kumar and R.M. Stern, "Maximum-likelihood-based cepstral inverse filtering for blind speech dereverberation," in *Proc. ICASSP '10*, Dallas, Texas, USA, Apr. 2010, pp. 4282 –4285.

[15] X. Xiao, J. Li, H. Li, and E. S. Chng, "Maximum likelihood adaptation of histogram equalization with constraint for robust speech recognition," in *Proc. ICASSP '11*, Prague, Czech, May 2011, pp. 5480–5483.

[16] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 1 edition, 2006.

[17] N. Metropolis and S. Ulam, "The Monte Carlo method," *Journal of the American statistical association*, vol. 44, no. 247, pp. 335–341, 1949.

[18] H. G. Hirsch, "Aurora-5 experimental framework for the performance evaluation of speech recognition in case of a hands-free speech input in noisy environments," Tech. Rep., 2007.

[19] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, pp. 133–147, 1998.