Feature-Space Structural MAPLR with Regression Tree-based Multiple Transformation Matrices for DNN

Hiroki Kanagawa*, Yuuki Tachioka*, Shinji Watanabe†, Jun Ishii*
* Information Technology R&D Center, Mitsubishi Electric Corporation, Kamakura, Japan
E-mail: {Kanagawa.Hiroki@ds,Tachioka.Yuki@eb,Ishii.Jun@ab}.MitsubishiElectric.co.jp Tel: +81-467-41-2072
† Mitsubishi Electric Research Laboratories, Cambridge, US
E-mail: watanabe@merl.com

***Abstract*—Feature-space maximum-likelihood linear regression (fMLLR) transforms acoustic features to adapted ones by a multiplication operation with a single transformation matrix. This property realizes an efficient adaptation performed within a pre-precessing, which is independent of a decoding process, and this type of adaptation can be applied to deep neural network (DNN). On the other hand, constrained MLLR (CMLLR) uses multiple transformation matrices based on a regression tree, which provides further improvement from fMLLR. However, there are two problems in the model-space adaptations: first, these types of adaptation cannot be applied to DNN because adaptation and decoding must share the same generative model, i.e. Gaussian mixture model (GMM). Second, transformation matrices tend to be overly fit when the amount of adaptation data is small. This paper proposes to use multiple transformation matrices within a feature-space adaptation framework. The proposed method first estimates multiple transformation matrices in the GMM framework according to the first-pass decoding results and the alignments, and then takes a weighted sum of these matrices to obtain a single feature transformation matrix frame-by-frame. In addition, to address the second problem, we propose feature-space structural maximum a posteriori linear regression (fSMAPLR), which introduces hierarchal prior distributions to regularize the MAP estimation. Experimental results show that the proposed fSMAPLR outperformed fMLLR.**

I. Introduction

Adaptation is an effective technique for automatic speech recognition (ASR) especially when a mismatch of acoustic features exists between training and decoding [1], [2]. Adaptation techniques are classified into model-space adaptations and feature-space adaptations. Model-space adaptations, such as maximum-likelihood linear regression (MLLR) [3], [4], [5], have been developed within the classical Gaussian mixture model (GMM) framework. In MLLR, multiple transformation matrices are estimated based on a regression tree technique, and Gaussian means in hidden Markov models (HMMs) are adapted using these transformations more precisely than a single transformation. However, the multiple transformation matrices tend to be over-estimated when the amount of adaptation data is small. To avoid this problem, researchers have proposed structural Bayesian approaches [6], [7], [8]. Structural maximum a posteriori linear regression (SMAPLR) is an extension of MLLR, and it introduces hierarchal prior distributions in the regression tree representation to regularize the maximum a posteriori (MAP) estimation of the transformation matrix. Similar to the extension of MLLR to constrained MLLR (CMLLR [9]), SMAPLR is also extended as constrained SMAPLR (CSMAPLR [10]). CSMAPLR achieves more robust performance than both CMLLR and SMAPLR. However, it is difficult to apply these model-space adaptation techniques to the other acoustic models than the GMM because model-space adaptations with multiple transformation matrices is tightly integrated with the GMM.

On the other hands, feature-space adaptations can be applied to any acoustic models because the adaptation is performed in an adaptation module, which it is independent of a decoding module. For example, feature-space MLLR (fMLLR) transforms acoustic features into adapted ones by a multiplication operation with a single transformation matrix in an adaptation module. Other studies related to fMLLR, fMAPLR [11], and fMAPLIN [12] have been proposed to improve the robustness of the transformation matrix estimation when the amount of adaptation data is small.

This type of adaptation can be easily applied to deep neural network (DNN) and other deep network architectures, where adapted features are inputted to DNN based acoustic models [13]. In addition, as an example of incorporating fMLLR to a DNN architecture, the linear input network (LIN) [14], [15], [16] was also proposed, which imitates this linear feature transformation by adding a layer without a non-linear activation function to the bottom of the DNN. Other study proposed to insert a linear transformation layer to other layers [17]. These approaches can easily adapt neural networks in the training time but in the testing time it is difficult to adapt a linear layer robustly because there are many parameters and the incorrect alignments degrade the accuracy of the parameter estimation significantly.

However, the performance may be worse than that of the model-space adaptation because this adaptation only uses a single transformation matrix and it cannot represent the complicated acoustic mismatch precisely. To exploit the advantages of model- and feature-space adaptations, our method uses multiple transformation matrices in the feature-space adaptation. After the first-pass decoding, we obtain the multiple transformation matrices based on the GMM-based adaptation framework. Then, we estimate a single transformation matrix by a weighted sum of these transformation matrices at the second-pass decoding. This process increases little computational cost from fMLLR. These weights are estimated frame-by-frame based on the HMM state alignments obtained from the first-pass decoding. In addition, to avoid over-fitting in the estimation of transformation matrices, structural maximum a posteriori (SMAP) criterion is adopted for our feature-space technique. This method is an extension of CSMAPLR, the feature-space SMAPLR (fSMAPLR). Experimental results show that the proposed fSMAPLR outperforms fMLLR for both GMM and DNN acoustic models in the feature-space adaptation.

This paper first describes the conventional adaptation techniques including model- and feature-space adaptations in Section II and proposes the feature-space adaptation method with multiple transformation matrices in Section III. Finally, experiments in Section IV show the effectiveness of our proposed method.

II. Conventional adaptation techniques

This section describes conventional adaptation techniques. The first two ones (II-A and II-B) are model-space adaptation. The CMLLR (II-A) is the most widely used model-space adaptation. This method can construct multiple transformation matrices based on regression tree. Because the CMLLR tends to be overly tuned for adaptation data, SMAP criterion is introduced to CMLLR (II-B). The last one (II-C) is a feature-space adaptation. The CMLLR with a single transformation matrix is equivalent to transforming feature vectors in feature space, which is called as the fMLLR.

A. CMLLR

In CMLLR, the D-dimensional mean vector $\boldsymbol{\mu}_{jm} \in \mathbb{R}^D$ and diagonal covariance matrix $\boldsymbol{\Sigma}_{jm} \in \mathbb{R}^{D\times D}$ of the Gaussian distribution are transformed into the adapted mean vector $\hat{\boldsymbol{\mu}}_{jm}$ and covariance matrix $\hat{\boldsymbol{\Sigma}}_{jm}$ by Eqs. (1) and (2) where j and m be the HMM state and GMM component index, respectively.

$$\hat{\boldsymbol{\mu}}_{jm} = \boldsymbol{\Theta}_{r(m,j)}\boldsymbol{\mu}_{jm} + \boldsymbol{\varepsilon}_{r(m,j)}, \quad (1)$$
$$\hat{\boldsymbol{\Sigma}}_{jm} = \boldsymbol{\Theta}_{r(m,j)}\boldsymbol{\Sigma}_{jm}\boldsymbol{\Theta}^{\top}_{r(m,j)}. \quad (2)$$

Here, r represents the regression class index and is uniquely specified from both m and j, and this structure is obtained by a regression tree-based method [5]; $\boldsymbol{\Theta}_{r(m,j)} \in \mathbb{R}^{D\times D}$ and $\boldsymbol{\varepsilon}_{r(m,j)} \in \mathbb{R}^D$ represent the transformation matrix and the bias vector, respectively. If the diagonal covariance matrix $\boldsymbol{\Sigma}_{jm}$ is transformed with Eq. (2), $\hat{\boldsymbol{\Sigma}}_{jm}$ becomes full covariance. Then the cost of likelihood computation and the size of acoustic model will extremely increase. However, the full-covariance Gaussian likelihood of the t-th frame D-dimensional feature vector $\boldsymbol{o}_t \in \mathbb{R}^D$ at the state j and component m can be rewritten with the diagonal-covariance Gaussian likelihood, as follows:

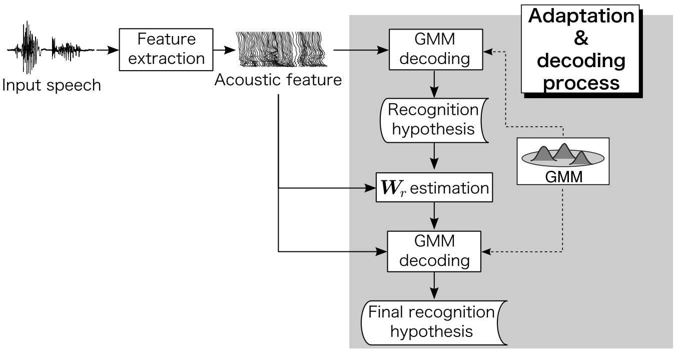

Fig. 1. Overview of model-space adaptation methods.

$$\mathcal{L}_{jm}(\boldsymbol{o}_t) = \mathcal{N}(\boldsymbol{o}_t|\hat{\boldsymbol{\mu}}_{jm}, \hat{\boldsymbol{\Sigma}}_{jm}) \quad (3)$$
$$= \left|\boldsymbol{A}_{r(m,j)}\right| \mathcal{N}(\hat{\boldsymbol{o}}_{r(m,j),t}|\boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}), \quad (4)$$

where $\mathcal{N}$ denotes a Gaussian distribution. The transformation matrix $\boldsymbol{A}_{r(m,j)}$, the bias vector $\boldsymbol{b}_{r(m,j)}$, and the transformed feature $\hat{\boldsymbol{o}}_{r(m,j),t}$ are defined as follows:

$$\boldsymbol{A}_{r(m,j)} \triangleq \boldsymbol{\Theta}^{-1}_{r(m,j)}, \quad (5)$$
$$\boldsymbol{b}_{r(m,j)} \triangleq -\boldsymbol{\Theta}^{-1}_{r(m,j)}\boldsymbol{\varepsilon}_{r(m,j)}, \quad (6)$$
$$\hat{\boldsymbol{o}}_{r(m,j),t} \triangleq \boldsymbol{A}_{r(m,j)}\boldsymbol{o}_t + \boldsymbol{b}_{r(m,j)} = \boldsymbol{W}_{r(m,j)}\begin{bmatrix}\boldsymbol{o}_t\\1\end{bmatrix}. \quad (7)$$

Thus, the use of Eq. (4) instead of Eq. (3), can avoid the full-covariance issues. However, this trick is highly depending on the Gaussian based likelihood calculation (e.g., the affine transformation $\boldsymbol{W}_{r(m,j)}$ of features depends on the state j and Gaussian indexes m), and cannot be used for the DNN-based score calculation. Fig. 1 shows the overview of model-space adaptation methods. In this types of adaptation, because adaptation and decoding processes are coupled, their processes must share the same acoustic model. Another issue of CMLLR is over-fitting in the case of the small amount of adaptation data.

B. CMLLR with structural priors (CSMAPLR)

The over-fitting issues of CMLLR can be mitigated by using a Bayesian approach. CSMAPLR [10] estimates a set of transformation matrices $\bar{\mathcal{W}} \triangleq \{\bar{\boldsymbol{W}}_r\}_{r=1}^R$ with a MAP criterion as follows

$$\bar{\mathcal{W}} = \underset{\mathcal{W}}{\operatorname{argmax}}\, P\left(\mathcal{W}\right) P\left(\boldsymbol{O}|\lambda, \mathcal{W}\right), \quad (8)$$

where $\boldsymbol{O} = \{\boldsymbol{o}_t|t = 1, \dots, T\}$ and λ represent the feature sequence and the set of GMM model parameters, respectively. We use a hierarchical prior distribution for $P\left(\mathcal{W}\right)$. For example, CSMAPLR uses the following prior distribution for $P\left(\boldsymbol{W}_r\right)$:

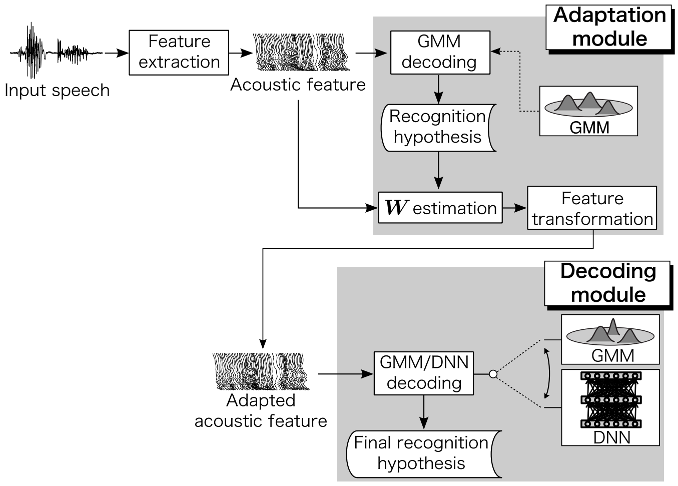

Fig. 2. Overview of feature-space adaptation methods.

$$P(\boldsymbol{W}_r) \propto |\boldsymbol{\Omega}|^{-D/2} |\boldsymbol{\Psi}|^{-(D+1)/2} \times \exp\left\{-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{W}_r - \boldsymbol{W}_{\mathsf{pa(r)}}\right)^{\top}\boldsymbol{\Omega}^{-1}\left(\boldsymbol{W}_r - \boldsymbol{W}_{\mathsf{pa(r)}}\right)\boldsymbol{\Psi}^{-1}\right\}, \tag{9}$$

where $\mathsf{pa(r)}$ represents the regression class index of the parent node of r, and $\boldsymbol{\Omega} \in \mathbb{R}^{D\times D}$ and $\boldsymbol{\Psi} \in \mathbb{R}^{(D+1)\times(D+1)}$ are hyper parameters for the prior distribution. In this study, we used the same settings as in [7], [10], i.e. $\boldsymbol{\Omega} = \tau \boldsymbol{I}_D$ and $\boldsymbol{\Psi} = \boldsymbol{I}_{D+1}$ where τ is a scaling parameter for SMAP (SMAP scale), which is a positive constant to control the effect of the prior distribution in the MAP estimation. When $\tau = 0$, it corresponds to the CMLLR with multiple transformation matrices.

C. fMLLR

In Section II-A, if we use a single transformation matrix instead of multiple ones, we can omit the regression index r, and further rewrite Eq. (4) as follows:

$$\mathcal{L}_{jm}(\boldsymbol{o}_t) = |\boldsymbol{A}|\,\mathcal{N}(\hat{\boldsymbol{o}}_t|\boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}), \tag{10}$$

where $\hat{\boldsymbol{o}}$ is a transformed feature defined as:

$$\hat{\boldsymbol{o}}_t \triangleq \boldsymbol{A}\boldsymbol{o}_t + \boldsymbol{b} = \boldsymbol{W}\begin{bmatrix}\boldsymbol{o}_t \\ 1\end{bmatrix}. \tag{11}$$

Thus, we can obtain adapted features from an adaptation module, which is separated from a decoding module as shown in Fig. 2. This type of feature-space adaptation technique such as fMLLR has been widely used. However, the performance may be worse than that of the model-space adaptation because this adaptation only uses a single transformation matrix.

III. Feature transformation with multiple transformation matrices in feature space

A. General form of weighting multiple transformation matrices

Fig. 3 shows an outline of the proposed method. This figure describes the application of five CMLLR transformation matrices to acoustic features when the input utterance is "aki". We associate acoustic features with transformation matrices frame-by-frame via state alignments to follow the temporal changes of acoustic features. To realize this association, we employ a state alignment[1], which is obtained by a GMM based acoustic model. In Fig. 3, the alignment is represented as the state index sequence for the HMM, i.e., $S = \{s_t | t = 1, \ldots T\}$, where T denotes the number of frames. With obtained s_t, we can specify a set of Gaussian components in a GMM as $\mathcal{M}_{s_t}$ and multiple regression classes $\{r(m, s_t)\}_{m\in\mathcal{M}_{s_t}}$. Therefore, we can associate the acoustic feature $\boldsymbol{o}_t$ with multiple transformation matrices $\left\{\boldsymbol{W}_{r(m,s_t)}\right\}_{m\in\mathcal{M}_{s_t}}$.

As discussed in Section II-A, model-space adaptations compute the output probability for each Gaussian component using a corresponding transformation matrix. However, it is necessary to avoid Gaussian-specific computation for applying it to the DNN, which can be performed by converting the

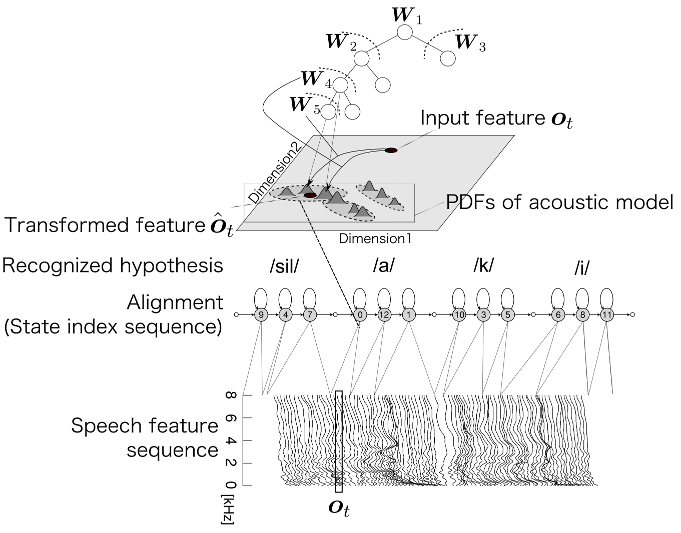

Fig. 3. Outline of the proposed method.

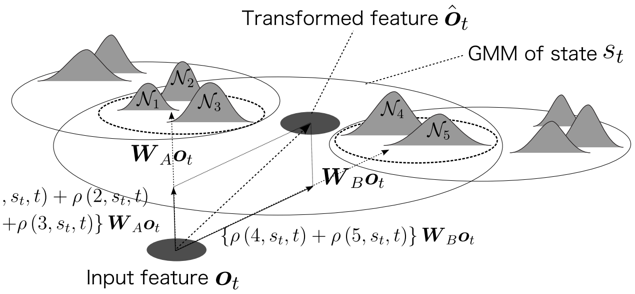
Transformed feature $\hat{\boldsymbol{o}}_t$
GMM of state s_t
$\mathcal{N}_1$
$\mathcal{N}_2$
$\mathcal{N}_3$
$\mathcal{N}_4$
$\mathcal{N}_5$
$\boldsymbol{W}_A\boldsymbol{o}_t$
$\boldsymbol{W}_B\boldsymbol{o}_t$
$\{\rho(1,s_t,t)+\rho(2,s_t,t)+\rho(3,s_t,t)\}\boldsymbol{W}_A\boldsymbol{o}_t$
$\{\rho(4,s_t,t)+\rho(5,s_t,t)\}\boldsymbol{W}_B\boldsymbol{o}_t$
Input feature $\boldsymbol{o}_t$

Fig. 4. Concrete example of the proposed feature transformation where the component of the state s_t includes five Gaussian distributions ($\mathcal{N}_1, \ldots, \mathcal{N}_5$, $\{1,2,3,4,5\} \in \mathcal{M}_{s_t}$) and $\boldsymbol{W}_A$ and $\boldsymbol{W}_B$ are transformation matrices. Distributions 1,2, and 3 share $\boldsymbol{W}_A$ and distributions 4 and 5 share $\boldsymbol{W}_B$. Their weight parameters are $\rho(1, s_t, \boldsymbol{o}_t), \ldots,$ and $\rho(5, s_t, \boldsymbol{o}_t)$.

[1] It is also possible to use lattices or N-best recognized hypotheses instead of alignments, because both of them can associate acoustic features with multiple HMM states existing in a certain frame.

Algorithm 1 The proposed feature transformation algorithm.

```
Input: Acoustic feature sequence $\boldsymbol{O} = \{\boldsymbol{o}_t | t = 1, \ldots, T\}$ and
  GMM acoustic model parameters $\lambda$
  Obtain state sequence $S = \{s_t | t = 1, \ldots T\}$ at the first-pass
  decoding ($S = \mathrm{decode}(\boldsymbol{O})$)
  Estimate transformation matrices $\bar{\mathcal{W}}$ by Eq. (8)
  for $t = 1, \cdots, T$ do
    for $m \in \mathcal{M}_{s_t}$ do
      $\hat{\boldsymbol{o}}_t = \sum_{m \in \mathcal{M}_{s_t}} \rho(m, s_t, \boldsymbol{o}_t) \left(\boldsymbol{A}_{r(m,s_t)} \boldsymbol{o}_t + \boldsymbol{b}_{r(m,s_t)}\right)$
        $= \sum_{m \in \mathcal{M}_{s_t}} \rho(m, s_t, \boldsymbol{o}_t) \boldsymbol{W}_{r(m,s_t)} \begin{bmatrix} \boldsymbol{o}_t \\ 1 \end{bmatrix}$
    end for
  end for
  Second-pass decoding with $\hat{\boldsymbol{O}} = \{\hat{\boldsymbol{o}}_t | t = 1, \ldots, T\}$
  (GMM/DNN)
```

transformation in the model-space adaptation $\boldsymbol{W}_r$ to that in the feature-space adaptation $\boldsymbol{W}$. To achieve this, our proposed method estimates a single transformation matrix by a weighted sum of these matrices $\boldsymbol{W}_r$. Unlike Eq. (11), the transformed feature vector at the t-th frame is represented as

$$\begin{aligned} \hat{\boldsymbol{o}}_t &= \sum\nolimits_{m \in \mathcal{M}_{s_t}} \rho(m, s_t, \boldsymbol{o}_t) \left(\boldsymbol{A}_{r(m,s_t)} \boldsymbol{o}_t + \boldsymbol{b}_{r(m,s_t)}\right) \\ &= \sum\nolimits_{m \in \mathcal{M}_{s_t}} \rho(m, s_t, \boldsymbol{o}_t) \boldsymbol{W}_{r(m,s_t)} \begin{bmatrix} \boldsymbol{o}_t \\ 1 \end{bmatrix}, \end{aligned} \tag{12}$$

where $\rho(m, s_t, \boldsymbol{o}_t)$ represents a frame-dependent weight parameter associated with both the state s_t and the m-th mixture of the GMM, which will be discussed in Section III-B. Fig. 4 shows a concrete example of the proposed feature transformation. A GMM component of the state s_t is composed of five Gaussian distributions ($\mathcal{N}_1, \ldots, \mathcal{N}_5$). Distributions 1, 2, and 3 share $\boldsymbol{W}_A$ and distributions 4 and 5 share $\boldsymbol{W}_B$. Once their weight parameters ρ are fixed, transformed features can be obtained by using a weighted sum of five transformed features corresponding to each Gaussian distribution. This adaptation can be performed in the feature-space, thus we can use the precise feature transformation for DNN, as well as GMM.

Algorithm 1 describes the proposed fSMAPLR procedure. We first obtain recognized hypotheses for the whole adaptation data and the context-dependent state alignments S from the first-pass decoding. Then, we estimate transformation matrices $\bar{\mathcal{W}}$ based on the CSMAPLR by Eq. (8). To adjust the influence of the prior distributions, we also introduce the SMAP scale τ as same as the CSMAPLR. When a single transformation matrix is used with $\tau = 0$, this method is equivalent to fMLLR. After $\bar{\mathcal{W}}$ is estimated, $\bar{\mathcal{W}}$ transforms the original feature $\boldsymbol{o}_t$ into the adapted feature $\hat{\boldsymbol{o}}_t$ with Eq. (12). Finally, the second-pass decoding obtains the ASR results with $\hat{\boldsymbol{o}}_t$ by using either GMM or DNN acoustic models.

B. Two types of weight parameters

In Section III-A, we described a method to obtain the transformed feature $\hat{\boldsymbol{o}}_t$ by using a weighted feature transformation matrix. In this section, we propose two types of weight parameters.

The first one uses the posterior of the m-th GMM component $\gamma(m, s_t, \boldsymbol{o}_t)$ for the weight $\rho(m, s_t, \boldsymbol{o}_t)$. Because the state s_t is known from the first-pass decoding, the weight parameter $\rho(m, s_t, \boldsymbol{o}_t)$ can be computed from $\gamma(m, s_t, \boldsymbol{o}_t)$:

$$\begin{aligned} \rho(m, s_t, \boldsymbol{o}_t) &= \gamma(m, s_t, \boldsymbol{o}_t) \\ &= \frac{w(m, s_t) \mathcal{N}\left(\boldsymbol{o}_t | \boldsymbol{\mu}_{m,s_t}, \boldsymbol{\Sigma}_{m,s_t}\right)}{\sum_{m' \in \mathcal{M}_{s_t}} w(m', s_t) \mathcal{N}\left(\boldsymbol{o}_t | \boldsymbol{\mu}_{m',s_t}, \boldsymbol{\Sigma}_{m',s_t}\right)}, \end{aligned} \tag{13}$$

where $\boldsymbol{\mu}_{m,s_t}$ and $\boldsymbol{\Sigma}_{m,s_t}$ are the non-adapted mean vector and diagonal covariance matrix, respectively[2]. However, it is well known that a few mixture components are very dominant over all components, and the posterior distribution tends to be very sparse. As a result, a single transformation matrix is selected with Eq. (13), and multiple transformation extension in Eq. (12) cannot be fully utilized.

The second one uses the mixture weight of the GMM as a weight for each transformation matrix. The motivation of this approach is to avoid the sparseness of $\gamma(m, s_t, \boldsymbol{o}_t)$. This can be approximated from Eq. (13) by ignoring $\mathcal{N}\left(\boldsymbol{o}_t | \boldsymbol{\mu}_{m,s_t}, \boldsymbol{\Sigma}_{m,s_t}\right)$, as follows[3]:

$$\begin{aligned} \rho(m, s_t, \boldsymbol{o}_t) &= \gamma(m, s_t, \boldsymbol{o}_t) \\ &\cong \frac{w(m, s_t)}{\sum_{m' \in \mathcal{M}_{s_t}} w(m', s_t)} = w(m, s_t), \end{aligned} \tag{14}$$

where $w(m, s_t)$ is different frame-by-frame because s_t depends on frame t. Note that m is also dependent on s_t ($m \in \mathcal{M}_{s_t}$). This enables our method to estimate transformation matrices more accurately than those using Eq. (13) because the estimation using Eq. (14) is more stable under noisy conditions.

IV. Experiments

A. Setups

We validated the effectiveness of our proposed approach for noisy "isolated" speech[4] using Track 2 from the second CHiME challenge [21]. This is a medium-vocabulary task in reverberant and noisy environments, whose utterances are taken from the *Wall Street Journal* database. The "isolated" speech was created by adding real-world noises recorded in the same room to reverberated speech at a −6, −3, 0, 3, 6, and 9 dB signal-to-noise ratio (SNR). The training dataset (si_tr_s) contains 7,138 utterances (15 [hour]) by 83 speakers (si84). Acoustic models (GMM and DNN) were

[2] Eq. (12) with (13) becomes very similar to discriminative feature transformation [18], [19], [20]. However these techniques are performed within a GMM discriminative training framework, and it is different from our approach, which focuses on feature-space adaptation for both GMM and DNN.

[3] The transformed features calculated from Eq. (14) may lead to the discontinuity of the transformed acoustic features because the transformation matrix is the same during the same s_t and at the changing point of the state, transformation matrices can be changed more drastically than those calculated from Eq. (13).

[4] There are two types of noisy speech: "isolated" and "embedded".

constructed with st_tr_s. The performance was evaluated on both the evaluation dataset (si_et_05), which contains 330 utterances (0.67[hour/SNR]×6[SNR]) by 12 speakers (Nov'92), and the development set (si_dt_05), which contains 409 utterances (0.77[hour/SNR]×6[SNR]) by 10 speakers. All utterances from each speaker (approximately 4–5min) were used for adaptation and evaluation. The added noises were non-stationary, such as the utterances of other speakers, home noises, or music. In addition, we suppressed noise by using prior-based binary masking [22] as a front-end processing. The tri-gram language model size was 5 k (basic). Some of the tuning parameters (e.g., the language-model weights and the number of transformation matrices) were optimized on the word-error rates (WERs) for si_dt_05.

We prepared two types of acoustic feature settings. The first setting was MFCC with feature transformations. After concatenating 0-12th order static MFCCs in nine contiguous frames, a total of 117 dimensional features were compressed into 40 dimensions by linear discriminant analysis (LDA) [23][5]. Then, a global semi-tied covariance (STC) [24] was applied to the LDA-transformed features in order to decorrelate between dimensions. After feature transformations with LDA and STC, the speaker-adaptive training [25] was used. The second setting was filter-bank (fbank) features with decorrelation. It starts from 0-22th order fbank features with Δ and $\Delta\Delta$. Fbank features cannot be represented well by a diagonal covariance model, thus GMM cannot model fbank features [26]. From this limitation, fMLLR with fbank features did not improve the ASR performance [27] and it is necessary to decorrelate fbank features before adaptation. In adaptation phase, a global STC $\boldsymbol{H}$ was applied to fbank features in order to decorrelate them, whereas in decoding phase, we applied an inverse STC $\boldsymbol{H}^{-1}$ to decorrelated and adapted (fMLLR/fSMAPLR) features in the same manner as [26].

The tri-phone GMM has 2,500 states and 15,000 Gaussian distributions. The DNN acoustic model has three hidden layers and 500,000 parameters. The initial learning rate of cross-entropy training was 0.02 and was decreased to 0.004 at the end of training. The mini-batch size was 128. Acoustic model training and decoding used the Kaldi toolkit [28] and the training procedure of acoustic models are the same as [22].

B. Appropriate weight parameters for the transformation matrices

Before comparing the proposed method with the conventional methods, we examined two types of weight parameters for transformation matrices, as discussed in Section III-B.

Table I shows the average WER using five and ten transformation matrices with both the posterior from Eq. (13) and the mixture weight from Eq. (14). The SMAP scale τ was set to 0, 100 and 1,000. These results show that mixture weights were better than posteriors in all cases because posteriors were sparse among their mixtures, as discussed in Section III-B. The optimized number of transformation matrices and τ will be discussed in more detail in Section IV-C and IV-D. The proposed fSMAPLR used the mixture weight (Eq. (14)) below.

TABLE I
WER (%) FOR THE DEVELOPMENT SET OF THE TRACK 2 OF THE SECOND CHIME CHALLENGE WHEN EITHER A POSTERIOR (EQ. (13)) OR A MIXTURE WEIGHT (EQ. (14)) WAS USED FOR THE WEIGHT IN EQ. (12). GMM ACOUSTIC MODEL WITH MFCC FEATURES WAS USED.

the number of		τ (SMAP scale)		
transformation matrices	weight	0	100	1000
5	posterior (Eq. (13))	39.7	39.6	39.3
	mixture weight (Eq. (14))	**39.5**	**39.5**	**39.2**
10	posterior (Eq. (13))	40.8	40.5	39.8
	mixture weight (Eq. (14))	**40.4**	**40.2**	**39.7**

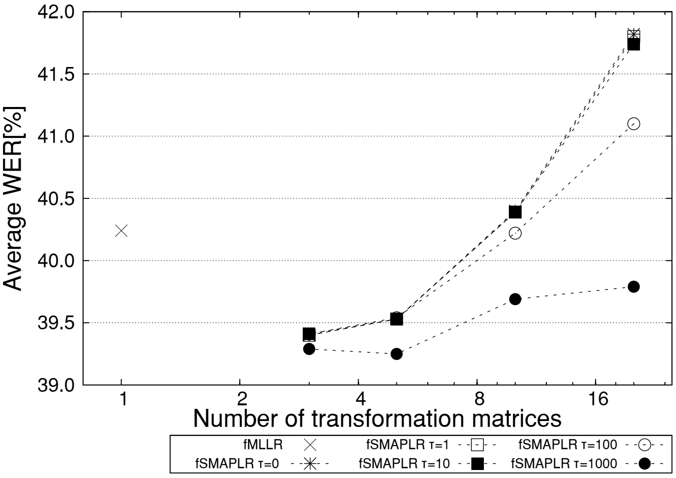

Fig. 5. Average WER (%) for isolated speech (si_dt_05) with the GMM acoustic model with MFCC features. Parametric study of the SMAP scale τ and the number of transformation matrices.

C. GMM acoustic model

Fig. 5 shows the average WER over all SNRs. The proposed fSMAPLR transforms the acoustic features frame-by-frame based on Eq. (12), using the multiple transformation matrices obtained by Eq. (8). The SMAP scale τ was set to 0, 1, 10, 100, and 1,000. The number of transformation matrices was 3, 5, 10, and 20. The proposed method with three or five transformation matrices outperformed fMLLR, for all τ's. When the proposed method used 10 and more transformation matrices, its performance degraded because the transformation matrices can be overly fit for the child nodes which contain only a small amount of data. However, increasing τ prevented over-fitting even if the number of transformation matrices increased. It confirmed the effectiveness of the proposed fSMAPLR. Based on these results, the optimal number of transformation matrices and τ for GMM were 5 and 1,000, respectively.

Table II shows the performance comparison on the evaluation set (si_et_05). This shows the WER for each SNR and their average denoted by "avg.". For reference, CSMAPLR was evaluated to show the performance of a model-space adaptation technique using multiple transformation matrices. We compared the fSMAPLR with the unadapted (w/o adapta-

[5] Delta features were not used for LDA.

TABLE II
WER (%) FOR ISOLATED SPEECH (SI_ET_05) WITH THE GMM ACOUSTIC MODEL USING MFCC FEATURES IN TERMS OF SNR. "W/O ADAPTATION" DENOTES THE BASELINE WITHOUT ADAPTATION.

Method	SNR [dB]						
	-6	-3	0	3	6	9	avg.
w/o adaptation	62.7	54.7	48.0	40.6	35.4	31.8	45.5
fMLLR	54.3	45.7	36.9	28.5	23.6	20.1	34.8
fSMAPLR	**52.9***	**44.7***	**35.2***	**27.3***	**22.5***	**18.7***	**33.6***
CSMAPLR	*52.7*	*43.7*	35.5	27.4	*22.5*	19.1	*33.5*

* significant at the 0.05 level.

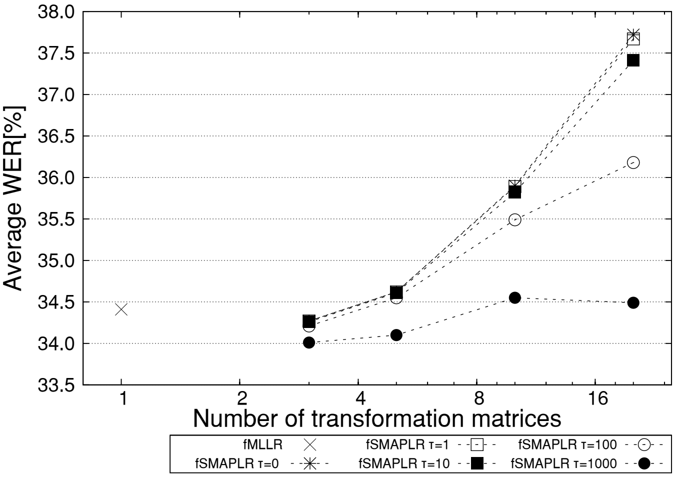

Fig. 6. Average WER (%) for isolated speech (si_dt_05) with the DNN acoustic model with MFCC features.

tion), fMLLR, and CSMAPLR. These results show that the adaptation was effective and that the proposed fSMAPLR outperformed fMLLR in all SNR cases and improved the average WER by 1.2%. Statistical hypothesis testing confirmed the effectiveness of the fSMAPLR at the 0.05 level. The performance of the fSMAPLR was comparable to that of the CSMAPLR. This confirmed the effectiveness of using multiple transformation matrices in both the model- and feature-space.

D. DNN acoustic model with MFCC feature

In this section, we evaluate the performance with DNN acoustic model. Fig. 6 shows the average WER for the development set (si_dt_05). The notations are the same as Fig. 5. These results show that the fSMAPLR with three transformation matrices outperformed fMLLR. As the number of transformation matrices increased too much, the performance of fSMAPLR degraded in the same manner as the GMM cases. Based on these results, the optimal number of transformation matrices and τ were 3 and 1,000, respectively.

Table III shows the performance of the proposed method on the evaluation set (si_et_05) and compared the fSMAPLR with the unadapted (w/o adaptation) and fMLLR. Note that the CSMAPLR cannot be realized with a DNN as described in the introduction. When compared with Table II, DNN outperformed GMM in all cases. The results show that the adaptation of DNN was still effective. Compared with fMLLR, the proposed fSMAPLR improved the ASR performance for all SNR conditions and the average WER by 0.7%. The significance was also confirmed by statistical hypotheses testing. Experiments thus confirm that the proposed fSMAPLR outperformed fMLLR for both GMM and DNN acoustic models.

TABLE III
WER (%) FOR ISOLATED SPEECH (SI_ET_05) WITH THE DNN ACOUSTIC MODEL USING MFCC FEATURES IN TERMS OF SNR.

Method	SNR [dB]						
	-6	-3	0	3	6	9	avg.
w/o adaptation	56.3	47.0	39.3	32.7	29.3	26.1	38.5
fMLLR	47.0	37.4	29.5	22.0	18.4	15.4	28.3
fSMAPLR	**46.6**	**36.4**	**29.2**	**21.6**	**17.2***	**15.0***	**27.6***

* significant at the 0.05 level.

TABLE IV
WER(%) FOR ISOLATED SPEECH (SI_ET_05) WITH THE DNN ACOUSTIC MODEL USING FBANK FEATURES IN TERMS OF SNR.

Method	SNR [dB]						
	-6	-3	0	3	6	9	avg.
w/o adaptation	47.9	38.7	32.4	24.7	21.4	19.5	30.8
fMLLR	**45.2**	35.7	29.1	21.5	18.2	16.6	27.7
fSMAPLR	45.3	**35.1**	**28.5**	**21.4**	**18.1**	**16.2***	**27.4***

* significant at the 0.05 level.

E. DNN acoustic model with the filter-bank feature

Table IV shows the WER for the fbank features. The performance of DNN w/o adaptation was significantly improved from the MFCC features. For adapted features, although the gains were small, fMLLR and our proposed fSMAPLR improved the performance further by 0.3%. This comparison also shows that our proposed fSMAPLR outperformed fMLLR. These experiments confirmed the robustness of the proposed method for both MFCC and fbank features.

V. CONCLUSIONS

This paper proposed a feature-space adaptation using multiple transformation matrices based on a regression tree, and introduced the SMAP criterion to avoid over-fitting in the estimation of transformation matrices. This aims to improve the performance of DNN because feature-space adaptations are suitable for DNN. The experimental results with GMM showed that the proposed method outperformed fMLLR, and was comparable to the model-space CSMAPLR, despite the fact that the proposed method was only applied to feature transformation, which is separated from decoding process. The computational time of the proposed method was almost comparable to that of the conventional fMLLR. Furthermore, adapted feature vectors by the proposed method can be inputted to DNN, which cannot be realized by the CSMAPLR, and we confirmed that the proposed fSMAPLR outperformed fMLLR for DNN in addition to GMM. Future work will seek to derive optimal weight parameters, introduce VBLR [8], [29] for estimating transformation matrices, and apply speaker adaptive training to DNNs by using obtained transformation matrices.

REFERENCES

[1] C.-H. Lee and Q. Huo, "On adaptive decision rules and decision parameter adaptation for automatic speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 88, no. 8, pp. 1241–1269, 2000.

[2] K. Shinoda, "Speaker adaptation techniques for automatic speech recognition," *Proc. APSIPA*, 2011.

[3] C. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," in *Computer Speech and Language*, 1995, pp. 100–104.

[4] V. Digalakis, D. Ritischev, and L. Neumeyer, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Trans. Speech and Audio Processing*, vol. 3, no. 5, pp. 357–366, 1995.

[5] M. Gales, "The generation and use of regression class trees for MLLR adaptation," *Technical Report CUED/F-INFENG/TR*, vol. 263, 1996.

[6] K. Shinoda and C.-H.Lee, "Structural MAP speaker adaptation using hierarchical priors," *Proc. of IEEE Workshop on Speech Recognition and Understanding*, pp. 381–388, 1997.

[7] O.Siohan, T.A.Myrvoll, and C.-H.Lee, "Structural maximum a posteriori linear regression for fast HMM adaptation," *Computer Speech and Language*, vol. 16, pp. 5–24, 2002.

[8] S. Watanabe, A. Nakamura, and B. H. Juang, "Bayesian linear regression for hidden Markov model based on optimizing variational bounds," *Proc. MLSP*, pp. 1–6, 2011.

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

[10] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," *IEEE Trans. Speech and Audio Processing*, vol. 17, no. 1, pp. 66–83, 2009.

[11] X.Lei, J.Hamaker, and X.He, "Robust feature space adaptation for telephony speech recognition," *Proc. ICSLP*, pp. 773–776, 2006.

[12] Z.Huang, J.Li, S.M.Siniscalchi, I.Chen, C.Weng, and C.-H. Lee, "Feature space maximum a posteriori linear regression for adaptation of deep neural networks," *Proc. INTERSPEECH*, pp. 2992–2996, 2014.

[13] T. Yoshioka, A. Ragni, and M. Gales, "Investigation of unsupervised adaptation of DNN acoustic models with filter bank input," *Proc. ICASSP*, pp. 13–16, 2014.

[14] J. Neto, L. Almeida, M. Hochberg, C. Martins, and L. Nunes, "Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system," *Proc. EUROSPEECH*, pp. 2171–2174, 1995.

[15] V. Abrash, H. Franco, A. Sankar, and M. Cohen, "Connectionist speaker normalization and adaptation," *Proc. Eurospeech*, 1995.

[16] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," *IEEE Workshop on Spoken Language Technology*, pp. 366–369, 2012.

[17] T. Ochiai, S. Matsuda, H. Watanabe, X. Lu, C. Hori, and S. Katagiri, "Speaker adaptive training for deep neural networks embedding linear transformation networks," *Proc. ICASSP*, pp. 4605–4609, 2015.

[18] D. Povey, "Improvements to fmpe for discriminative training of features." in *Proc. INTERSPEECH*, 2005, pp. 2977–2980.

[19] J. Droppo and A. Acero, "Maximum mutual information splice transform for seen and unseen conditions." in *Proc. INTERSPEECH*, 2005, pp. 989–992.

[20] B. Zhang, S. Matsoukas, and R. M. Schwartz, "Recent progress on the discriminative region-dependent transform for speech feature extraction." in *Proc. INTERSPEECH*, 2006, pp. 1573–1576.

[21] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'CHiME' speech separation and recognition challenge: datasets, tasks and baselines," *Proc. ICASSP*, pp. 126–130, 2013.

[22] Y. Tachioka, S. Watanabe, J. Le Roux, and J. R. Hershey, "Discriminative methods for noise robust speech recognition: A CHiME challenge benchmark," *The 2nd International Workshop on Machine Listening in Multisource Environments*, 2013.

[23] R. Haeb-Umbach and H. Ney, "Linear discriminant analysis for improved large vocabulary continuous speech recognition," *Proc. ICASSP*, pp. 13–16, 1992.

[24] M. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Trans. Speech and Audio Processing*, vol. 3, no. 7, pp. 272–281, 1999.

[25] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for Speaker-Adaptive Training," *Proc. ICSLP*, pp. 1137–1140, 1996.

[26] T. Sainath, B. Kingsbury, A. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran, "Improvements to deep convolutional neural networks for LVCSR," *Proc. ASRU*, pp. 315–320, 2013.

[27] T. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," *Proc. ICASSP*, pp. 8614–8618, 2013.

[28] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlcek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," *Proc. ASRU*, pp. 1–4, 2011.

[29] S.-J.Hahm, A. Ogawa, M. Delcroix, M. Fujimoto, T. Hori, and A. Nakamura, "Feature space variational Bayesian linear regression and its combination with model space VBLR," *Proc. ICASSP*, pp. 7898–7902, 2013.