MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Stereo-based Feature Enhancement Using Dictionary Learning

Watanabe, S.; Hershey, J.R.

TR2013-019 May 2013

Abstract

This paper proposes stereo-based speech feature enhancement using dictionary learning. Instead of posterior values obtained by a Gaussian mixture as in other methods, we use sparse weight vectors and their variants as an alternative noisy speech feature representation. This paper also provides an efficient algorithm that can be applied to large-scale speech processing. We show the effectiveness of the proposed approach by using a middle vocabulary noisy speech recognition task based on WSJ, which was provided by the 2nd CHiME Speech Separation and Recognition Challenge.

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

STEREO-BASED FEATURE ENHANCEMENT USING DICTIONARY LEARNING

Shinji Watanabe, John R. Hershey

Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

ABSTRACT

This paper proposes stereo-based speech feature enhancement using dictionary learning. Instead of posterior values obtained by a Gaussian mixture as in other methods, we use sparse weight vectors and their variants as an alternative noisy speech feature representation. This paper also provides an efficient algorithm that can be applied to large-scale speech processing. We show the effectiveness of the proposed approach by using a middle vocabulary noisy speech recognition task based on WSJ, which was provided by the 2nd CHiME Speech Separation and Recognition Challenge.

***Index Terms*—** Speech recognition, speech feature enhancement, dictionary learning, sparse representation, 2nd CHiME challenge track 2

1. INTRODUCTION

The investigation of speech enhancement techniques for automatic speech recognition (ASR) is one of the most important topics in modeling realistic environments. Unfortunately, speech enhancement techniques do not always improve ASR performance and sometimes degrade it, even when noises are correctly subtracted in terms of SNR measure or some subjective criteria. The main reason for this degradation is the difference in speech signal representations for power spectrum and MFCC domains. For example, spectral subtraction can aggressively denoise speech signals. However, since spectral subtraction makes speech signals unnatural (e.g., discontinuity due to a flooring process), these outliers are enhanced during the MFCC feature extraction step, which degrades ASR performance. Thus, the speech enhancement techniques for ASR must consider compatibility with the back-end ASR process.

One promising approach is to address this denoising problem in the MFCC domain. Note that, unlike the power spectrum domain, the MFCC domain does not retain the additivity property of signals and noises, and therefore, this approach may not effectively reduce noise components. However, since this approach directly enhances MFCC features, it yields steady improvement in terms of ASR performance. SPLICE is a typical technique for feature-domain speech enhancement that can obtain transformation functions from noisy to enhanced speech features [1, 2]. An interesting concept in SPLICE is that a (linear) feature transformation depends on a region that includes a set of neighboring noisy feature vectors. By modeling noisy features on the basis of a Gaussian mixture, the region information is represented by a posterior distribution of mixture components (soft clustering). The transformation parameters are usually estimated using parallel clean and noisy features that have the same label information uttered by the same speaker. Thus, the approach considers approximately piece-wise linear transformations, and can precisely enhance noisy feature vectors.

In this paper, we emphasize an alternative view of this piecewise linear feature enhancement technique. In this interpretation, the noisy signals are represented in an augmented feature space by expanding the original noisy feature space with the additional Gaussian posterior-based feature space. That is, the noisy signals are first mapped into points in the high-dimensional space, and the transformation can be realized as a projection from the augmented feature space to the enhanced feature space. This view is inspired by [2], which uses various types of posterior values based on discriminative methods other than GMM posteriors.

As an alternative augmented feature, we focus on sparse signal representation based on compressive sensing [3]. Compressive sensing with dictionary learning decomposes an original signal into a dictionary matrix and a sparse weight vector. This is a very efficient representation of high-dimensional signals, and sparse signal representation with dictionary learning has been applied to speech/audio enhancement techniques [4–7] as well as feature extraction and acoustic modeling [8–10]. For example, because of the non-negative constraint of the power spectrum, non-negative matrix factorization has been actively studied [4–7] in this framework. Although these approaches successfully model speech signals in the power spectrum domain, they do not always improve speech recognition performance because of the domain mismatch. Therefore, this paper applies the dictionary learning techniques to the MFCC domain speech enhancement, which minimizes speech distortion in the MFCC domain, and improves ASR performance more directly.

In this application, we also extend the proposed approach to deal with multistep feature transformation and long context information. On the basis of these extensions, we provide an efficient compressive sensing algorithm with dictionary learning, which is applicable to large-scale speech corpus. We applied these techniques to a reverberant and noisy speech recognition task based on a 5K vocabulary WSJ setup, provided by 2nd CHiME Speech Separation and Recognition Challenge [11].

2. STEREO-DATA BASED FEATURE ENHANCEMENT

This section explains the conventional stereo-data-based feature enhancement using Gaussian mixtures [1, 2]. In our task setting, we have stereo data composed of a clean feature sequence $\mathbf{X} = \{\mathbf{x}_t \in \mathbb{R}^D | t = 1, \cdots, T\}$ and the corresponding noisy feature sequence $\mathbf{Y} = \{\mathbf{y}_t \in \mathbb{R}^D | t = 1, \cdots, T\}$, where T is the number of frames, D is the number of dimensionality, and $\mathbf{X}$ and $\mathbf{Y}$ are $D \times T$ matrices. The key concept of stereo-based feature enhancement is to obtain a transformation from noisy features to clean features depending on the posterior probability of a Gaussian mixture component k at a frame t.

$$\mathbf{x}_t = \mathbf{y}_t + \sum_{k=1}^{K} \gamma_{k,t} \mathbf{b}_k, \tag{1}$$

where $\gamma_{k,t}$ is the posterior probability of a Gaussian mixture component (i.e., $p(k|\mathbf{y}_t)$), and K is the number of mixture components. $\mathbf{b}_k$ is a bias vector that represents a shift transformation from $\mathbf{y}_t$ to

$\mathbf{x}_t$[1].

By considering the above process for all frames, Eq. (1) is represented in the following matrix form:

$$\mathbf{X} = \mathbf{Y} + \mathbf{B}\mathbf{\Gamma} = \begin{bmatrix}\mathbf{I}_D & \mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{Y} \\ \mathbf{\Gamma}\end{bmatrix}, \quad (2)$$

where $\mathbf{I}_D$ is the $D \times D$ identity matrix. $\mathbf{\Gamma}$ is a $K \times T$ matrix composed of the posterior probabilities $\{\{\gamma_{k,t}\}_{k=1}^{K}\}_{t=1}^{T}$. $\mathbf{B}$ is a $D \times K$ matrix composed of K bias vectors, i.e., $\mathbf{B} = [\mathbf{b}_{k=1}, \cdots, \mathbf{b}_{k=K}]$. Eq. (2) indicates the interpretation that the noisy signals $\mathbf{Y}$ are represented in an augmented feature space $[\mathbf{Y}^\intercal, \mathbf{\Gamma}^\intercal]^\intercal$ by expanding the original noisy feature space with the additional Gaussian posterior-based feature space. That is, the noisy signals are first mapped into points in the high-dimensional space, and the transformation matrix $[\mathbf{I}_D, \mathbf{B}]$ can be obtained as a projection from the augmented feature space to the enhanced feature space.

The bias matrix is obtained by the following MMSE criterion.

$$\underset{\mathbf{B}}{\operatorname{argmin}} ||\mathbf{X} - \mathbf{Y} - \mathbf{B}\mathbf{\Gamma}||_2^2. \quad (3)$$

Thus, the bias matrix is estimated as follows:

$$\hat{\mathbf{B}} = (\mathbf{X} - \mathbf{Y})\mathbf{\Gamma}^\intercal(\mathbf{\Gamma}\mathbf{\Gamma}^\intercal)^{-1}. \quad (4)$$

In the evaluation step, unseen noisy features are transformed to the enhanced features by substituting $\hat{\mathbf{b}}_k$ into $\mathbf{b}_k$ in Eq. (1), which can be directly used in a back-end ASR process. This is a basic formulation of SPLICE, which has many variants. The most popular extension of SPLICE is to consider the frame-level context *in the posterior value domain* by concatenating the contiguous Gaussian posteriors in $\mathbf{\Gamma}$. This is an efficient way of considering long context information since the frame-level context expansion *in the MFCC domain* leads to a serious dimensionality problem in Gaussian modeling.

3. DICTIONARY LEARNING BASED FEATURE ENHANCEMENT

This paper replaces Gaussian mixture modeling with dictionary learning in the stereo-based feature enhancement approach, which is based on the interpretation in Eq. (2). The approach mainly consists of 1) dictionary learning with compressive sensing and 2) transformation estimation. In the dictionary learning step, we focus on the following decomposition of data:

$$\underset{\mathbf{D},\mathbf{w}_t}{\operatorname{argmin}} ||\mathbf{y}_t - \mathbf{D}\mathbf{w}_t||_2^2 + \Lambda(\mathbf{w}_t) \quad \forall t \quad (5)$$

where $\mathbf{D}$ is a $D \times K$ dictionary matrix and $\mathbf{w}_t$ is a weight vector at a frame t. $\Lambda(\mathbf{w}_t)$ is a regularization term for $\mathbf{w}_t$, and L1 norm is usually used to find a sparse solution. The step corresponds to the GMM estimation step with the Gaussian posterior estimation in the conventional SPLICE framework.

In the transformation estimation step, we simply replace the posterior distributions in Eq. (3) with the feature vector $\boldsymbol{\psi}_t$, which is obtained by compressive sensing via $\mathbf{w}_t$, as follows:

$$\underset{\mathbf{B}}{\operatorname{argmin}} ||\mathbf{X} - \mathbf{Y} - \mathbf{B}\mathbf{\Psi}||_2^2, \quad (6)$$

where $\mathbf{\Psi} = [\boldsymbol{\psi}_{t=1}, \cdots, \boldsymbol{\psi}_{t=T}]$. In the following sections, we will discuss each step in detail.

3.1. Compressive sensing

This paper uses two approaches to obtain sparse weight vectors. The first approach is orthogonal matching pursuit (*OMP*), which is a greedy search algorithm commonly used for the recovery of compressive sensed sparse signals [12].

$$\underset{\mathbf{w}_t}{\operatorname{argmin}} ||\mathbf{w}_t||_0 \text{ s.t. } ||\mathbf{y}_t - \mathbf{D}\mathbf{w}_t||_2^2 \leq \varepsilon \quad (7)$$

The approach finds the smallest number of non-zero elements among $\mathbf{w}_t$ that satisfies the upper bound (ε) of the signal residual. The second approach is called *Lasso*, which uses an L1 regularization term to obtain sparse weight vectors.

$$\underset{\mathbf{w}_t}{\operatorname{argmin}} ||\mathbf{y}_t - \mathbf{D}\mathbf{w}_t||_2^2 + \lambda||\mathbf{w}_t||_1 \quad (8)$$

This paper mainly investigates the effectiveness of dictionary learning by using these two approaches. However, there are many variants used to obtain sparse vectors (see [13, 14]), and the proposed approach can in principle apply any sparse coding algorithm.

Once we obtain the weight vectors, we can compute the posterior value at each dictionary atom k as follows:

$$p(k|\mathbf{y}_t) = \frac{p(\mathbf{y}_t|k)}{\sum_{k=1}^{K} p(\mathbf{y}_t|k)} \propto \exp\left(-\frac{||\mathbf{y}_t - w_{k,t}\mathbf{d}_k||_2^2}{2\sigma^2}\right) \quad (9)$$

Because of the sparseness of $w_{k,t}$, the computational cost of this posterior estimation is very low. As a feature $\psi_{k,t}$ for the latter transformation step, we have the following two options:

- Weight: $\psi_{k,t} \triangleq w_{k,t}$
- Posterior: $\psi_{k,t} \triangleq p(k|\mathbf{y}_t)$

Similar to SPLICE, we may use the posterior values in the transformation step, which adjusts the dynamic range of the transformation features from 0 to 1. However, since the scale of the weights is important information in dictionary learning, this paper evaluates both feature settings in the experiment.

3.2. Dictionary learning

Once we obtain the weight vectors, we can also estimate a dictionary matrix. This paper uses a typical dictionary learning algorithm named method of optimal direction (MOD), which estimates $\mathbf{D}$, as follows:

$$\tilde{\mathbf{D}} = f_{nc}\left(\mathbf{Y}\mathbf{W}^\intercal(\mathbf{W}\mathbf{W}^\intercal)^{-1}\right) \quad (10)$$

where $f_{nc}(\cdot)$ is a function used to normalize the column vectors $\tilde{\mathbf{d}}_k$ to be unit vectors (e.g., $\tilde{\mathbf{d}}_k \rightarrow \tilde{\mathbf{d}}_k/|\tilde{\mathbf{d}}_k|$). There are other approaches for estimatinge the dictionary matrix (e.g., k-SVD [13], online dictionary learning [15]). A dictionary matrix and sparse vectors are iteratively updated.

3.3. Transformation estimation

Once we obtain $\mathbf{w}_t$ given $\tilde{\mathbf{D}}$, we can consider the similar transformation to Eq. (1) by replacing $\gamma_{k,t}$ with $\psi_{k,t}$, as follows:

$$\mathbf{x}_t = \mathbf{y}_t + \sum_{k=1}^{K} \psi_{k,t}\mathbf{b}_k, \quad (11)$$

[1]We can also consider the linear transformation matrix of the feature vectors (e.g., $\sum_{k=1}^{K} \gamma_{k,t}(\mathbf{A}_k\mathbf{y}_t + \mathbf{b}_k)$). However, it is very practical to consider only bias vectors since the linear transformation does not significantly improve ASR performance and requires a complicated estimation process. Therefore, the paper does not involve linear transformation in the formulation. Note that the following discussion can consider the linear transformation matrix in this paper.

or we can represent this equation with a weight matrix $\mathbf{\Psi}$ as follows:

$$\mathbf{X} = \mathbf{Y} + \mathbf{B}\mathbf{\Psi}. \tag{12}$$

Thus, by using the same MMSE criterion with Eq. (3), we can obtain the following transformation matrix:

$$\tilde{\mathbf{B}} = (\mathbf{X} - \mathbf{Y})\mathbf{\Psi}^\intercal(\mathbf{\Psi}\mathbf{\Psi}^\intercal)^{-1}. \tag{13}$$

Thus, the approach first transforms noisy feature vectors $\mathbf{Y}$ to sparse vectors $\mathbf{\Psi}$ on the basis of a dictionary $\tilde{\mathbf{D}}$, and then the sparse vectors are transformed to the bias vectors $\tilde{\mathbf{B}}\mathbf{\Psi}$ between clean and noisy feature vectors in order to denoise the noise elements in $\mathbf{Y}$.

3.4. Multistep feature transformation

Since the approach enhances original features to transformed features in the same speech feature domain, the process can be iteratively undertaken. This multistep feature transformation is inspired by the feature learning concept in deep learning [16]. We consider the following extension of feature transformation from Eq. (12):

$$\mathbf{X}^{(n+1)} = \mathbf{X}^{(n)} + \mathbf{B}^{(n)}\mathbf{\Psi}^{(n)}, \tag{14}$$

where n is the number of transformation step and $\mathbf{X}^{(1)} \triangleq \mathbf{Y}$. The sparse vectors $\mathbf{\Psi}^{(n)}$ and the transformation matrix $\mathbf{B}^{(n)}$ are estimated step-by-step as follows:

$$\begin{aligned} &\underset{\mathbf{D}^{(n)}, \mathbf{w}_t^{(n)}}{\text{argmin}} \; ||\mathbf{x}_t^{(n)} - \mathbf{D}^{(n)}\mathbf{w}_t^{(n)}||_2^2 + \Lambda(\mathbf{w}_t^{(n)}) \quad \forall t. \\ &\mathbf{B}^{(n)} = (\mathbf{X} - \mathbf{X}^{(n)})(\mathbf{\Psi}^{(n)})^\intercal(\mathbf{\Psi}^{(n)}(\mathbf{\Psi}^{(n)})^\intercal)^{-1}. \end{aligned} \tag{15}$$

We experimentally observe that the iterative process monotonically decreases the L2 norm between the clean and enhanced speech features in the training step. The consideration of the theoretical convergence property of this multistep transformation is our future work.

3.5. Long context features

Similar to SPLICE, the approach can consider long context information. There are two ways of considering long context features in the stereo-based feature transformation approach. One is to consider the context information in the posterior domain at the transformation step used in SPLICE, i.e., $\boldsymbol{\gamma}_{t,c} = [\boldsymbol{\gamma}_{t-c}^\intercal, \cdots, \boldsymbol{\gamma}_t^\intercal, \cdots, \boldsymbol{\gamma}_{t+c}^\intercal]^\intercal$, where c is the number of contiguous frames to be considered in this feature expansion. The other is to consider the long context MFCC features in the dictionary learning step, i.e., $\mathbf{x}_{t,c}^{(n)} = [(\mathbf{x}_{t-c}^{(n)})^\intercal, \cdots, (\mathbf{x}_t^{(n)})^\intercal, \cdots, (\mathbf{x}_{t+c}^{(n)})^\intercal]^\intercal$.

In general, since the Gaussian mixture cannot correctly deal with high-dimensional features because of the dimensionality problem, SPLICE uses the posterior domain feature expansion. However, [2] points out the effectiveness of considering the long context MFCC features in the posterior estimation by employing dimensionality reduction techniques. One of the advantages of dictionary learning is that the approach does not significantly suffer from the dimensionality problem unlike the Gaussian mixture case. In addition, by considering multistep transformation, as discussed in the previous section, the latter transformation step can consider a longer context. Thus, this paper proposes to use the long context MFCC features in the dictionary learning step.

Algorithm 1 Dictionary learning in the (n) step

Initialize $\mathbf{D}^{(n)}$
repeat
  $\mathbf{S}_{ww} = \mathbf{0}$, $\mathbf{S}_{xw} = \mathbf{0}$
  for $u = 1$ to U **do**
    argmin$_{\mathbf{W}_u^{(n)}}$ $||\mathbf{X}_{u,c}^{(n)} - \mathbf{D}^{(n)}\mathbf{W}_u^{(n)}||_2^2$
    Accumulate $\mathbf{S}_{ww} \mathrel{+}= \mathbf{W}_u^{(n)}(\mathbf{W}_u^{(n)})^\intercal$
    Accumulate $\mathbf{S}_{xw} \mathrel{+}= \mathbf{X}_{u,c}^{(n)}(\mathbf{W}_u^{(n)})^\intercal$
  end for
  Update $\mathbf{D}^{(n)} = f_{nc}\left(\mathbf{S}_{xw}(\mathbf{S}_{ww})^{-1}\right)$
until some condition is met

Algorithm 2 Transformation estimation in the (n) step

$\mathbf{S}_{\psi\psi} = \mathbf{0}$, $\mathbf{S}_{x\psi} = \mathbf{0}$
for $u = 1$ to U **do**
  argmin$_{\mathbf{W}_u^{(n)}}$ $||\mathbf{X}_u^{(n)} - \tilde{\mathbf{D}}^{(n)}\mathbf{W}_u^{(n)}||_2^2$
  Get $\mathbf{\Psi}_u^{(n)}$ from $\mathbf{W}_u^{(n)}$
  Accumulate $\mathbf{S}_{\psi\psi} \mathrel{+}= \mathbf{\Psi}_u^{(n)}(\mathbf{\Psi}_u^{(n)})^\intercal$
  Accumulate $\mathbf{S}_{x\psi} \mathrel{+}= (\mathbf{X}_u - \mathbf{X}_u^{(n)})(\mathbf{\Psi}_u^{(n)})^\intercal$
end for
Update $\mathbf{B}^{(n)} = \mathbf{S}_{x\psi}(\mathbf{S}_{\psi\psi})^{-1}$
Update $\mathbf{X}^{(n+1)} = \mathbf{X}^{(n)} + \mathbf{B}^{(n)}\mathbf{\Psi}^{(n)}$

3.6. Implementation

An important factor in applying a new technique to speech processing is that we must consider the computational efficiency of dealing with large-scale speech database. For example, the famous WSJ0 training set used in the 2nd CHiME challenge track 2 holds a total of 5.4 million speech feature frames, and cannot store the entire $\mathbf{W}$ or $\mathbf{\Psi}$ if the dictionary size (K) is large. Therefore, this paper introduces utterance-by-utterance processing of dictionary learning and transformation estimation, which can only store utterance unit features, weights, and posteriors.

We consider an utterance index u, where the number of frames of u is T_u. The whole set of sparse weight vectors in a corpus is represented as $\mathbf{W} = \{\mathbf{W}_u \in \mathbb{R}^{D \times T_u} : u = 1, \cdots, U\}$, and the other frame-dependent values are represented similarly. We mainly have to compute the statistics $\mathbf{W}\mathbf{W}^\intercal$, $\mathbf{\Psi}\mathbf{\Psi}^\intercal$, $\mathbf{Y}\mathbf{W}^\intercal$, and $(\mathbf{X} - \mathbf{Y})\mathbf{\Psi}^\intercal$. We use the following relationship of the sub-matrix property:

$$\mathbf{W}\mathbf{W}^\intercal = \sum_{u=1}^{U} \mathbf{W}_u\mathbf{W}_u^\intercal. \tag{16}$$

This equation indicates that we can compute $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{W}$, and $\mathbf{\Psi}$ without storing these matrices in memory by accumulating these statistics for each utterance, similar to the E-step in the EM algorithm[2]. We can also parallelize this algorithm for each utterance or set of utterances. Finally, we provide the algorithms for the dictionary learning and transformation estimation steps, as shown in Algorithm 1 and 2.

[2]Some dictionary learning techniques (e.g., k-SVD) need to explicitly process full frame size matrices, and cannot be represented in (16). In this case, an online learning based extension is required.

Table 1. Experimental setup for the 2nd CHiME challenge track 2

Sampling rate	16 kHz
Feature type	MFCC + log Energy $+\Delta + \Delta\Delta$ (39 dim.)
Frame length	25 ms
Frame shift	10 ms
Window type	Hamming
# of categories	41 phonemes
Context-dependent HMM topology	1,860 HMM states (3-state left to right) 8 GMM components
Language model	2-gram (provided with WSJ0 corpus)
Vocabulary size	5k (closed vocabulary)

Table 2. WERs (%) for OMP and Lasso obtained using different types of features.

	Weight ($w_{k,t}$)	Posterior ($p\left(k|\mathbf{y}_t\right)$)
OMP	65.3	65.4
Lasso	66.5	65.6

4. EXPERIMENTS

We show the effectiveness of the proposed feature enhancement by using the 2nd CHiME challenge track 2 [17] based on HTK [18]. The task considers the problem of recognizing utterances being spoken in a noisy living room from recordings. The task uses the same setup as the 2011 CHiME Challenge [11] in terms of reverberation and noise conditions, but the target utterances here are taken from the speaker-independent medium (5k) vocabulary subset of the Wall Street Journal (WSJ0) corpus[3].

4.1. Experimental setup

Table 1 summarizes the experimental setup, which is used in the 2nd CHiME challenge. We used standard 39-dimensional MFCC vectors processed by cepstral mean normalization and a triphone HMM that has 1860 states and 8 components per state. The HMM was trained by using 7,138 reverberant speech utterances, and a language model was a 5K non-verbalized closed bigram provided by WSJ0. We used the development set (409 $\times$ 6 SNR conditions = 2,454 utterances) provided by CHiME for our evaluation, and the average WER for 6 SNR conditions (-6dB, -3dB, 0dB, 3dB, 6dB, 9dB) was 72.6%. 7,138 reverberant ($\mathbf{X}$) and noisy ($\mathbf{Y}$) speech utterances were used as parallel data in the stereo data feature enhancement techniques. The dictionary size (K) was fixed at 1,024 throughout the experiments.

4.2. Experimental results

Figure 1 examines the proposed approach for changing the number of context lengths (c), as discussed in Section 3.5, and the number of steps n, as discussed in Section 3.4. The experiments used OMP in compressive sensing, and used sparse weight vectors directly as transformation features (i.e., $\psi_{k,t} = w_{k,t}$). The result showed that there was a clearly improved WER from the baseline (72.6%) by more than 5%, and the multistep iterations and long context features further improved WER by 2% at most. These results show the effectiveness of dictionary learning for speech enhancement, particularly based on the multistep and long context extensions.

Table 2 compares the results with OMP and Lasso in compressive sensing, and sparse weight and posteriors ($\sigma = 1$ in Eq. (9)) used in the transformation estimation step. The number of context frames and steps were set as 4 and 2, respectively, on the basis of the result in Figure 1. In the case where weight vectors were used directly, the WERs depended on the compressive sensing methods (by 1%). In fact, we observed that the dynamic ranges of weight vectors were different from those of OMP and Lasso, and the degradation of Lasso would be due to this dynamic range difference. However, the difference was mitigated when we used the posterior values in the transformation step, which adjusted the dynamic ranges to the same scale. Therefore, the posterior value based feature transformation can somewhat absorb the difference caused by compressive sensing methods.

[3]Because the official CHiME challenge does not allow the use of stereo data processing, this result does not satisfy the challenge regulation.

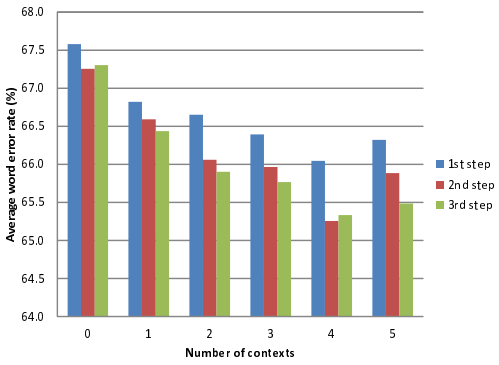

Fig. 1. Average word error rate for each context length and layer.

Table 3. WERs (%) for SPLICE and dictionary learning

	-6dB	-3dB	0dB	3dB	6dB	9dB
Baseline	86.25	82.79	76.08	71.35	63.04	55.87
SPLICE	80.72	75.55	67.37	62.40	54.39	49.48
Dictionary	80.66	75.46	67.74	62.60	54.64	49.13
SPLICE + Dictionary	80.43	74.63	67.16	62.49	54.08	48.92

Finally, we compared the result with SPLICE using a similar setting (the number of mixture components was 1024 and that of context frames was 4 in SPLICE). The WERs were almost comparable; thus, dictionary learning would be an alternative method to realize stereo-based feature enhancement. In addition, by combining SPLICE and dictionary learning, the performance was slightly but steadily improved for almost all SNR conditions. These results confirm the effectiveness of dictionary learning in stereo-based speech enhancement techniques.

5. SUMMARY

The paper proposed a stereo-based feature enhancement technique using dictionary learning as an alternative method for the Gaussian-based technique. The speech recognition experiments show improvements in terms of WER; thus, we confirm the effectiveness of dictionary learning in this framework. Our main future work is to overcome the limitations using the MMSE criterion. For example, the framework requires noisy and clean parallel data, which is not a realistic situation in some cases. In addition, it is generally agreed that the reduction of the L2 norm in the MFCC domain does not always reduce the word error rates, although the MFCC domain is more effective than the spectrum domain. Therefore, discriminative criterion should be considered in this framework to further improve the proposed approach, as represented by discriminative feature transformation techniques [19–22].

6. REFERENCES

[1] J. Droppo, L. Deng, and A. Acero, "Evaluation of the splice algorithm on the aurora2 database," in *Proc. Eurospeech*, 2001, vol. 1, pp. 217–220.

[2] M. Suzuki, T. Yoshioka, S. Watanabe, N. Minematsu, and K. Hirose, "Mfcc enhancement using joint corrupted and noise feature space for highly non-stationary noise environments," in *Proc. ICASSP'12*, 2012, pp. 4109–4112.

[3] D.L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[4] P. Smaragdis and J.C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. WASPA'03*, 2003, pp. 177–180.

[5] H. Kameoka, N. Ono, K. Kashino, and S. Sagayama, "Complex NMF: A new sparse representation for acoustic signals," in *Proc. ICASSP'09*, 2009, pp. 3437–3440.

[6] C. Févotte, N. Bertin, and J.L. Durrieu, "Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, 2009.

[7] J.F. Gemmeke, H. Van Hamme, B. Cranen, and L. Boves, "Compressive sensing for missing data imputation in noise robust speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 272–287, 2010.

[8] T.N. Sainath, A. Carmi, D. Kanevsky, and B. Ramabhadran, "Bayesian compressive sensing for phonetic classification," in *Proc. ICASSP'10*, 2010, pp. 4370–4373.

[9] G. Sivaram, S.K. Nemala, M. Elhilali, T.D. Tran, and H. Hermansky, "Sparse coding for speech recognition," in *Proc. ICASSP'10*, 2010, pp. 4346–4349.

[10] G. Saon and J.T. Chien, "Bayesian sensing hidden Markov models for speech recognition," in *Proc. ICASSP'11*, 2011, pp. 5056–5059.

[11] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green, "The pascal chime speech separation and recognition challenge," *Computer Speech & Language*, 2012.

[12] Y.C. Pati, R. Rezaiifar, and PS Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. ASILOMAR'93*, 1993, pp. 40–44.

[13] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[14] M. Fornasier and H. Rauhut, "Compressive sensing," *Handbook of Mathematical Methods in Imaging*, vol. 1, pp. 187–229, 2011.

[15] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[16] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 28, no. 6, 2012.

[17] 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," in *Proc. ICASSP'13*, 2013, accepted.

[18] Steve J Young, Gunnar Evermann, MJF Gales, D Kershaw, G Moore, JJ Odell, DG Ollason, D Povey, V Valtchev, and PC Woodland, "The HTK book (for HTK version 3.4)," *Cambridge University Engineering Department*, 2006.

[19] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition-," in *Proc. ICASSP 2005*, 2005, vol. 1, pp. 961–964.

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

[21] B. Zhang, S. Matsoukas, and R. Schwartz, "Discriminatively trained region dependent feature transforms for speech recognition," in *Proc. ICASSP'06*, 2006, vol. 1, pp. 313–316.

[22] M. Delcroix, A. Ogawa, S. Watanabe, T. Nakatani, and A. Nakamura, "Discriminative feature transforms using differenced maximum mutual information," in *Proc. ICASSP'12*, 2012, pp. 4753–4756.