UNSEEN NOISE ROBUST SPEECH RECOGNITION USING ADAPTIVE PIECEWISE LINEAR TRANSFORMATION

Keigo Chijiiwa, Masayuki Suzuki, Nobuaki Minematsu, Keikichi Hirose

The University of Tokyo, 7–3–1, Hongo, Bunkyo-ku, Tokyo, 113–0033, Japan

{keigo,suzuki,mine,hirose}@gavo.t.u-tokyo.ac.jp

ABSTRACT

SPLICE is one of the speech enhancement methods based on feature conversion, which shows a high performance with a relatively small amount of calculation. After modeling noisy speech features as GMM, conversion functions are obtained for individual GMM components. The original SPLICE estimates clean feature vectors as a weighted summation of the converted versions of input vectors. Since the conversion functions are determined and fixed only by using training data, the effectiveness of the original SPLICE will be lower in the case of unseen noisy environments. In this paper, we propose a novel method to adapt the conversion functions to work well in unseen environments. First, to realize adaptive conversion functions, we characterize those functions using their super vectors. Then, we conduct PCA on the super vectors to reduce the number of parameters to be adapted. By representing the super vectors through their PCA–based base functions and weights, we implement an efficient adaptation method of conversion functions, which we call Eigen–SPLICE here after. Evaluation experiments show that Eigen–SPLICE has reduced word error rate by 21.0% relative to the conventional SPLICE, and by 24.1% relative to EMS SPLICE in the test set B of the AURORA–2 task.

Index Terms— SPLICE, Piecewise linear techniques, Noise robust, Speech recognition, Principal component analysis

1. INTRODUCTION

As people begin to use speech recognition applications not only in quiet environments but also in noisy environments, noise robustness becomes a required feature for any speech recognition application. In real noisy environments speech signals are distorted by various kinds of noises, which cause mismatch easily between input speech features and the acoustic models which were trained in a clean condition. To solve this problem, various methods have been proposed.

For example, [1] and [2] proposed a method of using VTS (Vector Taylor Series) and [3] took another approach of feature enhancement using noisy speech and its original clean speech, called Piecewise LInear Compensation for Environments, or SPLICE. Both methods were shown to be effective to realize noise robustness in automatic speech recognition. They use statistical information of the environmental noise, with which the VTS method adapts clean speech acoustic models into the ones for noisy speech. On the other hand, SPLICE trains a noisy speech model using GMM (Gaussian Mixture Model) in advance and uses it for feature enhancement, where noisy speech features are converted to clean ones. These two methods showed high performances in a continuous digits recognition task in noisy environments using AURORA–2 [4].

These methods, however, have some problems in real noisy environments. First, the VTS method is computationally expensive to perform frame-by-frame model adaptation especially when it is applied to MFCC features, while it is not costly when applied to FBANK features. One the other hand, the conversion functions of the original SPLICE are determined and fixed only by using training data, so the effectiveness of the original SPLICE will be lower in the case of unseen noisy environments. To solve this problem, EMS (Environmental Model Selection) method was already proposed. EMS SPLICE softly estimates the kind of environment beforehand by calculating its posterior probability and enhance input noisy features using the calculated results. Similar to the original SPLICE, since EMS also uses fixed sets of training environments, however, its performance is expected to be still insufficient in unseen environments.

In this paper, we propose a novel method to improve SPLICE to work well in unseen environments by adapting the conversion functions for any input noisy environments. First, to realize adaptive conversion functions, we characterize those functions using their super vectors. Then, we conduct PCA on the super vectors to reduce the number of parameters to be adapted. By representing the super vectors through their PCA–based base functions and weights, we implement an efficient adaptation method of conversion functions. To confirm whether it works or not, we conduct experiments with AURORA–2 database.

The rest of the paper is organized as follows. Section 2 describes the conventional SPLICE method. Section 3 presents our proposed method of Eigen–SPLICE. Section 4 presents the experimental results in the AURORA–2 database. Finally, Section 5 concludes the paper and describes future directions.

2. CONVENTIONAL SPLICE

In this section, we review the conventional SPLICE, which has the training phase and the enhancing phase. In the training phase, SPLICE assumes that noisy speech features can be modeled as a GMM and linear conversion functions can be estimated for each sub-space of the GMM using time-aligned sequences of clean speech features and noisy speech features, i.e., stereo data. In the enhancing phase, SPLICE converts given noisy speech features into clean ones. The piecewise linear conversions are intended to approximate the true nonlinear conversions of noisy speech features into their clean versions. This SPLICE technique has been improved by various methods, such as [5] and [6].

Given clean speech feature, $\boldsymbol{x}$, and noisy speech feature, $\boldsymbol{y}$, we describe the detail procedure of SPLICE.

2.1. Model of speech and its distortion

The first assumption is that the noisy speech feature follows GMM:

$$\begin{aligned} p(\boldsymbol{y}) &= \sum_s p(\boldsymbol{y}, s), \\ &= \sum_s p(\boldsymbol{y}|s)p(s), \text{where} \\ p(\boldsymbol{y}|s) &= N(\boldsymbol{y}; \boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s). \end{aligned} \tag{1}$$

s, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ represent the index of GMM's component, mean vector and variance and covariance matrix respectively.

The second assumption made by the SPLICE is that the conditional probability density function (PDF) for clean speech feature $\boldsymbol{x}$ given noisy speech feature $\boldsymbol{y}$ and the GMM component index s, is Gaussian whose mean vector is obtained through linear transformation of $\boldsymbol{y}$. Thus, the conditional PDF is assumed to have the following form of:

$$p(\boldsymbol{x}|\boldsymbol{y}, s) = N(\boldsymbol{x}; \boldsymbol{A}_s\boldsymbol{y} + \boldsymbol{r}_s, \boldsymbol{\Gamma}_s). \tag{2}$$

$\boldsymbol{\Gamma}_s$ represents variance and covariance matrix of the distribution.

2.2. Feature enhancement

Because of these two basic assumptions, it becomes easier to estimate clean speech features from their distorted counterparts by the MMSE method. The MMSE estimate is obtained by the following conditional expectation of clean speech feature $\boldsymbol{x}$ given the observed noisy speech feature $\boldsymbol{y}$:

$$\hat{\boldsymbol{x}} = E_x[\boldsymbol{x}|\boldsymbol{y}] = \sum_s p(s|\boldsymbol{y})E_x[\boldsymbol{x}|\boldsymbol{y}, s]. \tag{3}$$

Using Eq. 2, it is clear that:

$$E_x[\boldsymbol{x}|\boldsymbol{y}, s] = \boldsymbol{A}_s\boldsymbol{y} + \boldsymbol{r}_s. \tag{4}$$

By using Eq. 4 in Eq. 3, we get

$$\hat{\boldsymbol{x}} = \sum_s p(s|\boldsymbol{y}) \left(\boldsymbol{A}_s\boldsymbol{y} + \boldsymbol{r}_s\right). \tag{5}$$

The MMSE estimate of $\boldsymbol{x}$ is a weighted summation of the converted features based on each GMM component's conversion function.

2.3. SPLICE training

Since noisy speech distribution $p(\boldsymbol{y})$ is assumed to follow a mixture of Gaussians, the standard EM algorithm can be used to estimate $\boldsymbol{\mu}_s$, and $\boldsymbol{\Sigma}_s$ on noisy speech. If stereo data are available, the parameters $\boldsymbol{A}_s$ and $\boldsymbol{r}_s$ of the conditional PDF $p(\boldsymbol{x}|\boldsymbol{y}, s)$ can be estimated using the maximum likelihood criterion:

$$\begin{aligned} \boldsymbol{r}_s &= \frac{\sum_t p(s|\boldsymbol{y}_t)(\boldsymbol{x}_t - \boldsymbol{y}_t)}{\sum_t p(s|\boldsymbol{y}_t)}, \\ \boldsymbol{A}_s &= \frac{\sum_t p(s|\boldsymbol{y}_t)\left(\boldsymbol{x}_t - \boldsymbol{r}_s\right)\boldsymbol{y}_t^{\mathsf{T}}}{\sum_t p(s|\boldsymbol{y}_t)\boldsymbol{y}_t\boldsymbol{y}_t^{\mathsf{T}}}, \text{where} \\ p(s|\boldsymbol{y}_t) &= \frac{p(\boldsymbol{y}_t|s)p(s)}{\sum_s p(\boldsymbol{y}_t|n)p(s)}. \end{aligned} \tag{6}$$

t represents the index of time and $^{\mathsf{T}}$ represents transposition. In this training procedure, SPLICE requires a set of stereo data. Since conversion functions of SPLICE are determined and fixed only by using training data, the effectiveness of the original SPLICE will be lower in the case of unseen noisy environments. To solve this problem, EMS (Environmental Model Selection) was already proposed.

2.4. Environmental model selection

The original SPLICE uses GMM (a set of Gaussians) to model all the kinds of noisy environments prepared in the training data. Contrary to this, EMS (Environmental Model Selection) SPLICE uses GMM for each noisy environment. Mathematically speaking, EMS SPLICE is realized as follows.

First, we calculate $p(e|\boldsymbol{y}_t)$, and then calculate EMS SPLICE's output as weighted sum of each specific environment's output using its own GMM and conversion function:

$$\hat{\boldsymbol{x}}_{t\text{EMS}} = \sum_e p(e|\boldsymbol{y}_t) \left\{ \sum_{s_e} p(s_e|\boldsymbol{y_t}) \left(\boldsymbol{A}_{s_e}\boldsymbol{y_t} + \boldsymbol{r}_{s_e}\right) \right\}, \tag{7}$$

where e represents the index of environment, and s_e represents the index of GMM component for environment e. Similar to the original SPLICE, since EMS uses fixed sets of training environments, its performance is expected to be still insufficient in unseen environments.

In the next section, we describe our proposed method for improving the conventional method to work well in unseen environments by adapting the conversion functions.

3. EIGEN–SPLICE

We try to improve SPLICE to work well in unseen environments by adapting its conversion functions for any input noisy environments. In this paper we adapt only $\boldsymbol{r}_s$ in Eq. 5 to simplify the implementation. This PCA–based method is commonly used by such as Eigen–MLLR [7] and Eigen Voice Conversion (EVC) [8]. Unlike Eigen–MLLR or EVC, our proposed method does not adapt the parameters of PDF, but adapts the parameters of conversion functions. Thus, our proposed method needs not only input noisy speech but also its clean counterparts for the adaptation. However, it is actually impossible to get both of stereo data in unseen noisy environments, so we have to prepare quasi-stereo data from input noisy speech and clean speech in the training data.

3.1. Calculation of principal components

First, we train the common GMM and the common conversion matrices $\boldsymbol{A}_s$ for all types and SNR levels of noisy environments in the training data. This procedure is the same as in the original SPLICE. Next, we tain $\boldsymbol{r}_s^i$ for each type and SNR level of noisy environments:

$$\hat{\boldsymbol{r}_s}^i = \underset{\boldsymbol{r}_s^i}{\text{argmin}} \sum_t \sum_s p(s|\boldsymbol{y}_t^i) \left\{ \boldsymbol{x}_t^i - \left(\boldsymbol{A}_s\boldsymbol{y}_t^i + \boldsymbol{r}_s^i\right) \right\}^2, \tag{8}$$

where i represents the index of environments. Then, we concatenate $\boldsymbol{r}_s^i$ of all components into a super vector $\boldsymbol{SV}^i$:

$$\boldsymbol{SV}^i = \{\hat{\boldsymbol{r}_1^i}, \cdots, \hat{\boldsymbol{r}_s^i}, \cdots, \hat{\boldsymbol{r}_S^i}\}, \tag{9}$$

where S represents the number of Gaussian mixtures. This super vector is calculated for each environment. After that, we conduct

PCA and obtain a bias vector $\boldsymbol{BV}$ and the principal components $\boldsymbol{PC}^m$ $(m = 1, \cdots M)$ from these super vectors:

$$\begin{aligned} \boldsymbol{BV} &= \{\boldsymbol{b}_1, \cdots, \boldsymbol{b}_s, \cdots, \boldsymbol{b}_S\}, \quad (10) \\ \boldsymbol{PC}^1 &= \{\boldsymbol{c}_1^1, \cdots, \boldsymbol{c}_s^1, \cdots, \boldsymbol{c}_S^1\}, \\ &\vdots \\ \boldsymbol{PC}^m &= \{\boldsymbol{c}_1^m, \cdots, \boldsymbol{c}_s^m, \cdots, \boldsymbol{c}_S^m\}, \\ &\vdots \\ \boldsymbol{PC}^M &= \{\boldsymbol{c}_1^M, \cdots, \boldsymbol{c}_s^M, \cdots, \boldsymbol{c}_S^M\}, \quad (11) \end{aligned}$$

where m and M represent the index and the number of the principal components, respectively. Finally, using the obtained bias vector and the principal components, we rewrite Eq. 5 into the new formula:

$$\begin{aligned} \hat{\boldsymbol{x}}_t &= \sum_s p(s|\boldsymbol{y}_t)\left(\boldsymbol{A}_s\boldsymbol{y}_t + \boldsymbol{B}_s\boldsymbol{w} + \boldsymbol{b}_s\right), \text{where} \\ \boldsymbol{B}_s &= \{\boldsymbol{c}_s^{1\mathsf{T}}, \cdots, \boldsymbol{c}_s^{M\mathsf{T}}\}, \quad (12) \end{aligned}$$

where $\boldsymbol{w}$ represents the weight parameters of the principal components.

3.2. Adaptation

In Eq. 12, $\boldsymbol{A}_s$, $\boldsymbol{B}_s$ and $\boldsymbol{b}_s$ were estimated in the training phrase and, in the adaptation phase, only $\boldsymbol{w}$ is adaptively estimated and used for conversion. The weight parameters are estimated by the MMSE method using a small amount of stereo data of a new environment:

$$\hat{\boldsymbol{w}} = \underset{w}{\text{argmin}} \sum_t \{\boldsymbol{x}_t - \sum_s p(s|\boldsymbol{y}_t)\left(\boldsymbol{A}_s\boldsymbol{y}_t + \boldsymbol{B}_s\boldsymbol{w} + \boldsymbol{b}_s\right)\}^2 \quad (13)$$

$\hat{\boldsymbol{w}}$ can be obtained analytically:

$$\begin{aligned} \hat{\boldsymbol{w}} &= \left(\sum_t \boldsymbol{M}_t^{\mathsf{T}}\boldsymbol{M}_t\right)^{-1}\left(\sum_t \boldsymbol{M}_t^{\mathsf{T}}\boldsymbol{E}_t\right), \text{where} \\ \boldsymbol{M}_t &= \sum_s p(s|\boldsymbol{y}_t)\boldsymbol{B}_s, \\ \boldsymbol{E}_t &= \boldsymbol{x}_t - \sum_s p(s|\boldsymbol{y}_t)\left(\boldsymbol{A}_s\boldsymbol{y}_t + \boldsymbol{b}_s,\right). \quad (14) \end{aligned}$$

3.3. Quasi–stereo data

As we described, the proposed method needs the stereo data of new environments. However, it is actually impossible to get stereo data in unseen noisy environments, so we have to generate quasi-stereo data from input noisy speech and clean speech in training data. First, we assume that the beginning part and the ending part of noisy speech contains only noise. Then, we add the noisy segments to clean speech in the training data. The obtained quasi–stereo data for the new environment is used for adaptation. It should be noted that, $\hat{\boldsymbol{w}}$ is estimated by using noise features, not noisy speech features, of a new noisy utterance and some clean utterances in the training data. The figure 1 shows the over view of the convectional SPLICE and our proposed Eigen–SPLICE.

4. EXPERIMENT

To confirm whether our proposed method works or not, several experiments are conducted using AURORA–2 database.

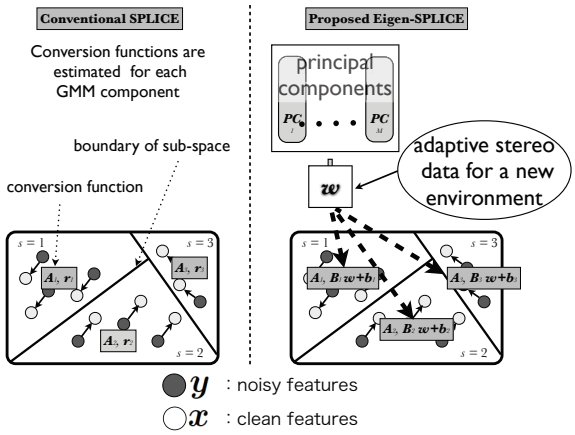

Fig. 1. Overview of Eigen-SPLICE

4.1. Experimental condition

First, we train the common conversion function of $\boldsymbol{A}_s$ using 16 environments, 4 types of Subw, Babble, Car and Exhibit and 4 different SNR levels, in the training set of AURORA–2. Next, we calculate $\boldsymbol{r}_s^i$ $(i = 1, \cdots, 16)$ for each environment using Eq. 8, and made 16 super vectors $\boldsymbol{SV}^i$ $(i = 1, \cdots 16)$. Then, we conduct PCA to obtain a bias vector $\boldsymbol{BV}$ and principal components $\boldsymbol{PC}^m$ $(m = 1, \cdots M)$.

We use test sets of A and B to test our proposed method. Test set A contains utterances in the same environments as those in the training data and test set B contains utterances in unseen environments (Rest., Street, Airport, Sta.). Experiments using the test sets A and B show us the noise–closed performance and the noise–open performance, respectively.

In these experiments, we assume that the beginning 250 [ms] and the ending 250 [ms] of each input utterance correspond to noisy segments without speech. We add these noise segments to 8 utterances which are randomly selected from the clean training data. And we use the first 6 principal components for representing the super vectors. This experimental set–up is because a preliminary experiment had showed these parameters were the best for adaptation. We use 39 dimensional features of MFCC $+\Delta + \Delta\Delta$. We evaluate the proposed method in the form of the averaged recognition accuracy in SNR 0, 5, 10, 15, 20 [dB]. The conventional SPLICE and Eigen–SPLICE have a 512–mixture GMM for modeling the noisy speech feature vectors, EMS has a 128–mixture GMM for each environment. We trained word HMMs only for clean environments, each of which has 18 states for each word and each state has 20–mixture GMM.

4.2. Results and discussion

Table 1. shows the results of speech recognition in test-set A and set B. Set A contains utterances in the same environments as those used in the training data. In this situation, EMS SPLICE works the best, but its superiority is very small. However, the performance of SPLICE and EMS SPLICE in set B, unseen environments, are degraded, while the performance of Eigen–SPLICE remains very close to the performance in the closed situation.

Table 2. shows the detail of the results in set B. We can see that

Table 1. Word recognition accuracies in the four cases of no enhancement, conventional SPLICE, EMS SPLICE and Eigen-SPLICE

	no enhance	SPLICE	EMS SPLICE	E–SPLICE
Set A	51.45	85.95	87.53	87.32
Set B	44.86	83.71	83.04	87.13

Eigen–SPLICE works well even in high noise levels, in which the other methods don't work well. We consider that this performance gain is directly attributed to adaptive estimation of the conversion functions.

5. CONCLUSION

We proposed a novel method to improve SPLICE to work well in unseen environments by adapting the conversion functions using quasi–stereo data. We reduced the number of parameters by using PCA, so much less stereo data are needed to adapt the conversion functions, and quasi–stereo data are prepared by using noise segments in input utterances and clean utterances in the training data. We conducted evaluation experiments to investigate accuracy of speech recognition in noisy environments. As the result, it was demonstrated that Eigen–SPLICE works the best in unseen environments, in which the conventional SPLICE and EMS don't work well.

The proposed method can be applied to other types of features. For example, the features obtained through noise mean normalization [3] and those obtained with multi layer perceptron can be used in the proposed method [9]. Furthermore, our proposed method can be applied to uncertainty decoding [10]. We believe that they will improve the performance of Eigen–SPLICE even more.

6. REFERENCES

[1] J.C. Segura, A. De La Torre, M.C. Benitez, and A.M. Peinado, "Model-based compensation of the additive noise for continuous speech recognition. Experiments using the AURORA II database and tasks," *Proc. Eurospeech*, vol. 1, pp. 221–224, 2001.

[2] Y. Zhao and B.H. Juang, "On noise estimation for robust speech recognition using vector Taylor series," *Proc. ICASSP*, pp. 4290–4293, 2010.

[3] J. Droppo, L. Deng, and A. Acero, "Evaluation of the SPLICE algorithm on the Aurora2 database," *Proc. Eurospeech*, vol. 1, pp. 217–220, 2001.

[4] D. Pearce and H.G. Hirsch, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," *Proc. ICSLP*, vol. 4, pp. 29–32, 2000.

[5] Y. Shinohara, T. Masuko, and M. Akamine, "Feature enhancement by speaker-normalized SPLICE for robust speech recognition," *Proc. ICASSP*, pp. 4881–4884, 2008.

[6] J. Droppo, A. Acero, and L. Deng, "Uncertainty decoding with splice for noise robust speech recognition," *Proc. ICASSP*, vol. 1, pp. 57–60, 2002.

Table 2. The detail of word recognition accuracies in set B in the four cases of (a) no enhancement, (b) SPLICE, (c) EMS SPLICE and (d) Eigen–SPLICE

(a)	Rest.	Street	Airport	Station	Avg.
20dB	90.16	94.63	86.88	88.18	89.96
15dB	71.31	84.59	66.39	69.01	72.83
10dB	44.77	59.90	40.42	42.77	46.97
5dB	10.51	31.08	11.08	15.82	17.12
0dB	-15.2	11.24	-6.26	0.00	-2.55
Avg.	40.31	56.29	39.70	43.16	44.86
(b)	Rest.	Street	Airport	Station	Avg.
20dB	99.20	98.55	99.02	99.04	98.95
15dB	98.68	97.70	98.63	97.90	98.23
10dB	95.76	93.53	95.50	94.26	94.76
5dB	85.45	76.00	83.72	77.72	80.72
0dB	56.16	40.51	50.70	36.07	45.86
Avg.	87.05	81.26	85.51	81.00	83.71
(c)	Rest.	Street	Airport	Station	Avg.
20dB	99.29	98.40	99.08	99.04	98.95
15dB	98.43	97.67	98.36	97.72	98.05
10dB	95.52	92.62	94.69	94.01	94.21
5dB	83.70	75.73	81.36	77.41	79.55
0dB	52.01	40.75	47.24	37.70	44.42
Avg.	85.79	81.03	84.15	81.18	83.04
(d)	Rest.	Street	Airport	Station	Avg.
20dB	99.26	98.58	99.02	99.11	98.99
15dB	98.80	97.82	99.08	98.33	98.51
10dB	96.56	94.23	97.14	95.34	95.82
5dB	87.93	82.86	88.16	84.60	85.89
0dB	61.93	51.60	62.24	49.98	56.44
Avg.	88.90	85.02	89.13	85.47	87.13

[7] K. Chen, W. Liau, H. Wang, and L. Lee, "Fast speaker adaptation using eigenspace-based maximum likelihood linear regression," *Proc. ICSLP*, vol. 3, pp. 742–745, 2000.

[8] T. Toda, Y. Ohtani, and K. Shikano, "Eigenvoice conversion based on gaussian mixture model," *Proc. ICSLP*, pp. 2446–2449, 2006.

[9] Z. Tüske, C. Plahl, and R. Schlüter, "A study on speaker normalized MLP features in LVCSR," *Proc. INTERSPEECH*, pp. 1089–1092, 2011.

[10] H. Liao and M.J.F. Gales, "Issues with uncertainty decoding for noise robust automatic speech recognition," *Speech Communication*, vol. 50, no. 4, pp. 265 – 277, 2008.