FEATURE ENHANCEMENT BY SPEAKER-NORMALIZED SPLICE FOR ROBUST SPEECH RECOGNITION

Yusuke Shinohara, Takashi Masuko, and Masami Akamine

Toshiba Corporate Research and Development Center
1, Komukai-Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582, Japan
yusuke.shinohara@toshiba.co.jp

ABSTRACT

The SPLICE method of feature enhancement is known for its powerful performance. It learns a mapping from noisy to clean feature vectors given a set of stereo training data. However, feature vector variation caused by speaker changes conceals noise-induced variation, which is what we want to find in the SPLICE training. In this paper, an improvement of SPLICE by means of speaker-normalization is proposed. The training data is first normalized with respect to speaker variation, and a mapping is learned afterward. CMLLR with a GMM as its target is utilized for the speaker-normalization, where the GMM representing a standard speaker is learned via a novel variant of the speaker adaptive training. The proposed method was evaluated on Aurora2, and achieved a relative word error rate reduction of 38% over the conventional SPLICE.

Index Terms— Feature enhancement, SPLICE, speaker normalization, speaker adaptive training, robust speech recognition.

1. INTRODUCTION

The performance of speech recognition systems degrades in noisy conditions, which is a primary issue in utilizing the systems in real world [1]. A large number of techniques have been developed over the past decades to increase the noise robustness of the systems. Many of those techniques adopt the approach of feature enhancement, which is a technique to restore clean feature vectors from noisy ones. Examples of this approach include the spectral subtraction, the minimum mean-square error short-time spectral amplitude estimator [2], the vector Taylor series technique [3], and the SPLICE (Stereo-based Piecewise Linear Compensation for Environments) [4].

The SPLICE is a method of feature enhancement, and is known for its powerful performance in noisy environments. The method works as follows. In the training phase, given a set of stereo training data collected in some noise environment, the method learns a piecewise linear mapping from noisy to clean feature spaces. A set of mappings are learned, one for each noise environment. In the test phase, given a sequence of noisy vectors, the most likely environment is selected first, and the mapping of the selected one is applied to the noisy vectors to clean them up. In summary, SPLICE learns and normalizes the variation induced by noises to improve the performance of speech recognition systems in noisy conditions.

However, speaker variation of the training data affects the process of SPLICE training, and degrades the speech recognition performance as a consequence. The training data of SPLICE is usually collected from a large variety of speakers, and a non-negligible amount of feature vector variation is induced by speaker changes. This variation conceals the noise-induced variation, which is what we want to find in the SPLICE training.

To remedy this problem, we propose an improvement of SPLICE by means of speaker normalization. First, all the training data is processed to normalize the speaker characteristics of feature vectors. Then, the SPLICE training is conducted to find a set of mappings, one for each environment. By removing the speaker-dependent variation first, noise induced variation can be found more clearly. We refer to this approach the *speaker-normalized SPLICE*. To normalize speaker characteristics, CMLLR (constrained maximum likelihood linear regression) [5, 6] is utilized in this study, where a Gaussian mixture model representing a standard speaker is used as a target of CMLLR. A novel variant of the speaker adaptive training [7, 6] is also proposed to construct the standard speaker model more robustly, particularly effective in the case when the training data is highly noisy.

2. METHOD

2.1. Conventional SPLICE

The SPLICE learns a map from noisy to clean feature vectors given a set of stereo training data, which is typically created by artificially adding noises to a set of clean utterances. A Gaussian mixture model (GMM) is first constructed from the noisy training data. Then, for each mixture component k of

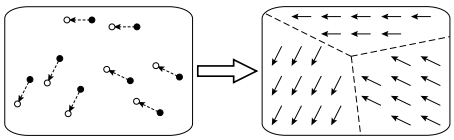

Fig. 1. The SPLICE training process. White and black points represent clean and noisy training data, respectively.

the GMM, a correction vector r_k is trained as

$$r_k = \frac{\sum_i p(k|y_i)(x_i - y_i)}{\sum_i p(k|y_i)}, \tag{1}$$

where $\{x_i\}$ and $\{y_i\}$ are the clean and noisy vectors, respectively. A piecewise linear mapping is formed with those correction vectors to get an estimate of a clean vector $\hat{x}$ from a noisy one y as

$$\hat{x} = y + \sum_k p(k|y) r_k. \tag{2}$$

Figure 1 depicts the SPLICE training process. The feature space is split into regions (three in this figure), and a correction vector is learned for each of them to form a piecewise linear mapping. Note that a hard split of regions is used to simplify the figure. For each noisy environment, one SPLICE mapping is constructed as described above. In the test phase, the most likely environment e^* is first selected as $e^* = \arg\max_e p(Y|e)$, where $Y = \{y_1 \dots y_T\}$ is a noisy vector sequence of length T, e is an environment index, and the likelihood is calculated with the GMM of e. The SPLICE mapping of the selected environment is used to clean-up the vectors.

2.2. Speaker-normalized SPLICE

The SPLICE training is a process to find noise-induced variation of feature vectors (i.e., correction vectors). Since the performance level of a training-based system like SPLICE is heavily dependent on training data, training data is usually collected from a wide variety of speakers so as to make the system robust against speaker changes. However, a non-negligible amount of feature vector variation is induced by the speaker changes. We model a speaker-dependent variation by an affine transformation of feature space as

$$x_r = A_r x_s + b_r, \tag{3}$$

where x_s and x_r are feature vectors of a standard speaker s and a particular speaker r respectively, and matrix A_r and vector b_r specify the affine transformation. Figure 2 depicts a speaker-dependent transformation of feature space. As can be seen from the figure, due to the speaker-dependent transformation, noise-induced variation (i.e., correction vectors) cannot be seen clearly. The speaker-dependent variation conceals the noise-induced variation, and interferes in the SPLICE training process.

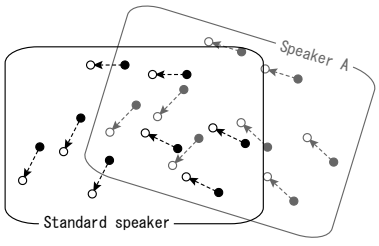

Fig. 2. Variation induced by speaker changes, depicted by rotation and shift of the feature space.

To remedy this problem, speaker variation is normalized beforehand of the SPLICE training. By doing so, correction vectors can be learned more clearly. This is the basic idea of the proposed speaker-normalized SPLICE approach. In the training phase of the proposed method, training data, both clean and noisy, are processed to normalize speaker characteristics, and the SPLICE mappings are learned afterward (Figure 3). Note that region splitting is omitted in the figure for simplicity. In the test phase, the input vectors are first processed to normalize speaker characteristics, and are then processed with SPLICE to normalize noise characteristics.

Conventionally, the cepstral mean normalization (CMN) or the histogram equalization is used prior to SPLICE to normalize channel characteristics. These techniques have the effect of speaker normalization, and can be seen as examples of the speaker-normalized SPLICE approach. In this paper, we use more dedicated and powerful method for speaker normalization. That is CMLLR.

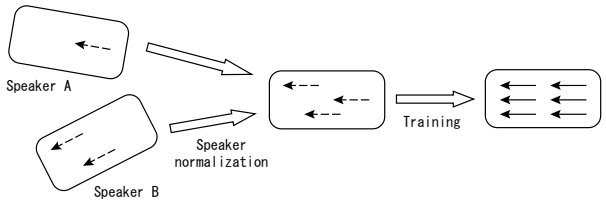

Fig. 3. The training process of speaker-normalized SPLICE.

2.3. Speaker-normalized SPLICE with CMLLR

To normalize speaker characteristics, we use the CMLLR technique beforehand of SPLICE. A GMM representing a standard speaker model is used as a target of CMLLR. Given a sequence of noisy feature vectors, an affine transformation maximizing the likelihood against the standard speaker model is estimated with CMLLR, and used to normalize the feature vectors.

The training of speaker-normalized SPLICE with CMLLR goes as follows (Figure 4a). First, affine transformations for speaker normalization are found, one for each speaker, via

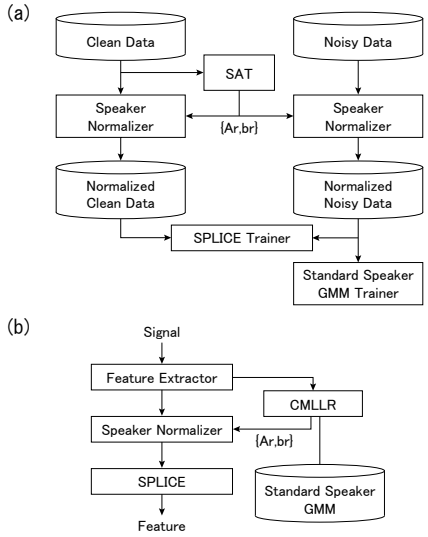

Fig. 4. The speaker-normalized SPLICE system: (a) in training, and (b) in test.

the speaker adaptive training (SAT) with CMLLR and GMM. A clean dataset is used to conduct SAT, because speaker-dependent variation can be estimated more reliably without the interference of noise-induced variation of feature vectors. Note that SAT outputs transforms and a standard speaker GMM, but the GMM is discarded here, since the GMM trained with clean data does not match with noisy test data. The obtained transforms are then applied to both clean and noisy training data, and the resultant speaker-normalized stereo data is used to train the SPLICE mappings. A GMM representing the standard speaker model is build with the normalized noisy data, and used as a target of CMLLR in the test phase.

In the test phase, the speaker-normalized SPLICE works as follows (Figure 4b). The input signal is converted to a sequence of feature vectors. Using those vectors as adaptation data, a speaker-normalization transform is estimated via CMLLR with the standard speaker GMM as its target. The transform is then applied to the same vectors to normalize speaker characteristics. The most likely environment is selected using those speaker-normalized vectors, and the SPLICE map of the selected one is used to clean them up.

The idea of feature vector normalization by CMLLR with GMM as its target is also reported in [8]. Our method is different from theirs in that we use CMLLR in combination with SPLICE. Also, the way of building the standard speaker GMM is different. They used the traditional SAT with noisy training data to construct the GMM. However, due to the presence of noise-induced feature variation, speaker-dependent transforms cannot be estimated robustly. On the other hand, we estimate speaker-dependent transforms with clean data and apply them to noisy data to avoid the interference by noise-induced feature variation. This procedure can be said to be a novel variant of SAT, and is particularly effective when the training data is highly noisy.

3. EXPERIMENT

3.1. Experimental conditions

The proposed and related methods were evaluated in the Aurora2 experimental framework [9]. The Aurora2 is a framework to evaluate the performance of speech recognition systems under noisy conditions. The task is connected digits recognition in English. The test consists of three parts; set A to evaluate the performance in known noise conditions, set B in unknown noise conditions, and set C in unknown channel conditions. See [9] for the full description of the Aurora2 framework.

The acoustic model was trained using the noisy (or multi condition) training data. The Aurora2 reference training script was used without modification except that the number of mixtures per state was increased from three to 20, and the spectral type was changed from magnitude to power.

A CMLLR transform was defined by a block diagonal matrix of (13 13 13) and a bias vector. A 512-mixture GMM was used as a target of CMLLR. In the test, nine to 10 utterances collected in a specific speaker-noise-SNR condition were used to estimate a CMLLR transform, which was then applied to the same utterances to normalize speaker characteristics. Although some 1500 frames were used to estimate a transform in this experiment, 300 frames or so should be enough to reliably estimate a transform, according to our preliminary experiment. For each of the 20 environments (combination of four noises and five SNRs), a 256-mixture GMM and a SPLICE mapping (i.e., 256 correction vectors) were trained using the speaker-normalized stereo training data. The HTK 3.4 was used for extracting feature vectors, training acoustic models, estimating CMLLR transforms, and decoding in the backend.

Five other methods (Baseline, CMN, CMLLR, SPLICE and CMN+SPLICE) were evaluated along with the proposed method (CMLLR+SPLICE). Note that an acoustic model was trained for each of the feature enhancers using the multi condition training data processed by that enhancer.

3.2. Two ways of speaker adaptive training

A GMM representing a standard speaker is used as a target to estimate a CMLLR speaker-normalization transform. There are two ways to train the GMM, and their comparative performance was evaluated. In the first way (denoted Conventional), the speaker adaptive training using a GMM as a target of CMLLR is carried out with the noisy data to yield a standard speaker model. In the second way (denoted Proposed),

Table 1. A comparison of the two ways of SAT.

	A	B	C	Ave
Conventional	93.58	93.10	93.44	93.36
Proposed	93.92	93.35	93.66	93.64

the GMM is trained as described in section 2.3; the speaker-normalization transforms are first estimated via SAT using the clean data, the obtained transforms are applied to the noisy data, and the normalized noisy data are used to build a standard speaker GMM.

The standard speaker model was trained in each way, and the obtained model was plugged-in to the CMLLR speaker normalizer. Table 1 shows the comparative performance of the two ways. The proposed way of training achieved a slightly better result than the conventional way in all of the three sets. This result supports our conjecture that the speaker normalization transform does not change with respect to the noise level, and can be estimated more reliably using the clean data than using the noisy data.

3.3. Experimental result

Table 2 shows the summary of the Aurora2 results. Note that the CMLLR result in Table 1 is shown again for comparison. The baseline performance without applying any kind of feature enhancement was 90.19%. The performance of set B was worse than that of set A, which was caused by the mismatch between training and test environments. Also in set C, the performance degraded due to the channel mismatch. By applying CMN, utterance by utterance, the performance was significantly improved from the baseline, particularly in set C, showing the CMN's capability to normalize channel variation. The CMLLR speaker normalization further improved the performance. The method is more flexible than CMN in that it can rotate, scale, or shear the space as well as just shift it. This flexibility boosted the capability to normalize speaker variation. On the other hand, SPLICE did not improve the performance on average. Although the performance improved in known noise environments, a significant drop was observed in unknown ones. When the noise type is unknown to the system, SPLICE is forced to select an environment from 20, no matter how different the one is compared with the current environment, and input vectors are mapped to somewhere irrelevant as a result. Nevertheless, SPLICE can build a new environment model on the fly by learning a new map using a stereo data generated from the current noise. Hence, the weakness in unknown environments can be covered to some extent. When CMN is used as a preprocessor of SPLICE, the average word error rate was reduced by about 25% relative to the basic SPLICE. The proposed method (i.e., speaker-normalized SPLICE with CMLLR) achieved the best result among all, reducing the error by about 38% relative to the basic SPLICE, and 18% relative to the CMN+SPLICE.

Table 2. Comparative performance of the feature enhancers in the Aurora2 evaluation (word accuracy in %).

	A	B	C	Ave
Baseline	91.36	89.53	89.18	90.19
CMN	92.22	91.99	92.77	92.24
CMLLR	93.92	93.35	93.66	93.64
SPLICE	93.29	86.88	89.32	89.93
CMN+SPLICE	93.62	91.15	92.61	92.43
CMLLR+SPLICE	94.58	92.98	93.82	93.79

4. CONCLUSION

In this paper, we proposed the speaker-normalized SPLICE approach for enhancing features in noisy conditions. The method is a two-stage feature enhancer; in the first stage, CMLLR is used to normalize speaker characteristics; in the second, SPLICE is used to normalize the effect of noises. By the introduction of the speaker-normalization step, feature vector variation caused by speaker changes is removed, and the SPLICE mappings (i.e., shifts of feature vectors induced by noises) can be learned more clearly. The proposed method achieved a word error rate reduction of 38% relative to the conventional SPLICE without any preprocessing, and 18% relative to the SPLICE with CMN preprocessing.

5. REFERENCES

[1] Y. Gong, "Speech recognition in noisy environments: A survey," *Speech Communication*, vol. 16, pp. 261–291, 1995.

[2] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean square error short-time spectral amplitude estimator," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[3] P. Moreno, *Speech Recognition in Noisy Environments*, Ph.D. thesis, Carnegie Mellon University, 1996.

[4] L. Deng, A. Acero, M. Plumpe, and X. Huang, "Large-vocabulary speech recognition under adverse acoustic environments," in *Proc. ICSLP*, 2000.

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

[6] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, 1998.

[7] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. ICSLP*, 1996.

[8] G. Stemmer, F. Brugnara, and D. Giuliani, "Adaptive training using simple target models," in *Proc. ICASSP*, 2005.

[9] D. Pearce and H.-G. Hirsh, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ICSLP*, 2000.