HIGH-PERFORMANCE ROBUST SPEECH RECOGNITION USING STEREO TRAINING DATA

Li Deng, Alex Acero, Li Jiang, Jasha Droppo, and Xuedong Huang

Microsoft Research
One Microsoft Way, Redmond, Washington 98052, USA
{deng,alexac,lij,jdroppo,xdh}@microsoft.com

ABSTRACT

We describe a novel technique of SPLICE for high-performance robust speech recognition. It is an efficient noise reduction and channel distortion compensation technique that makes effective use of stereo training data. In this paper, we present a new version of SPLICE using the minimum-mean-square-error decision, and describe an extension by training clusters of HMMs with SPLICE processing. Comprehensive results using a Wall Street Journal large vocabulary recognition task and with a wide range of noise types demonstrate superior performance of the SPLICE technique over that under noisy matched conditions (19% word error rate reduction). The new technique is also shown to consistently outperform the spectral-subtraction noise reduction technique, and is currently being integrated into the Microsoft MiPad, a new generation PDA prototype.

1. INTRODUCTION

Noise robustness is critical to virtually all types of speech recognition applications. There are two major classes of approaches to noise robustness: the feature-domain approaches (e.g., [1]) and model-domain ones (e.g.,[1]). Since the model-domain approaches aim at transforming the HMM parameters so as to match the noisy-speech statistics, their performance is typically limited by that achieved under a matched noisy condition. In our recent work [3], we showed that such a limit can be beaten by a novel feature-domain approach where noise reduction is performed on both training and test data and noise adaptive training is used to cover a wide range of anticipated noisy environments.

In [3], we described a novel noise-reduction algorithm named SPLICE (*Stereo-based Piecewise LInear Compensation for Environments*) for the first time. SPLICE was shown to be consistently superior to spectral subtraction, especially for nonstationary noises. In this paper, we will present an improvement of SPLICE from the previous approximate-MAP decision rule to the current minimum mean square error (MMSE) rule. We then describe an extension of SPLICE by training clusters of HMMs using the SPLICE-processed training data. Comprehensive results from large vocabulary speech recognition on the WSJ task with a wide range of noise types will be presented in this paper to demonstrate high performance of the above newly developed techniques. In particular, we will show highly reliable results of using vector quantization (VQ) distortion as a metric to automatically detect the noise type and level for test utterances. This provides a key to solving practical problems associated with using the SPLICE algorithms in the deployment of robust speech recognizers.

2. ASSUMPTIONS, LEARNING, AND MMSE RULE IN SPLICE

SPLICE assumes that the noisy speech cepstral vector, $\mathbf{y}$, is distributed according to a mixture of Gaussians. That is, it partitions the acoustic space in terms of the noisy speech, in contrast to some earlier algorithms (such as FCDCN [1]) that partitioned the acoustic space in terms of the clean speech cepstral vector $\mathbf{x}$. One main advantage of this new partitioning is that it obtains a more uniform and desirable division of the cepstral space directly for the observable data $\mathbf{y}$. In addition, the cepstral enhancement algorithm becomes slightly more efficient in computation. In the current implementation, the parameters of the mixture of Gaussians for $\mathbf{y}$ are determined by performing VQ followed by training each of the means and variances in the mixture using the training vectors classified into the corresponding VQ codewords.

SPLICE further assumes that a clean speech cepstral vector $\mathbf{x}$ and its corresponding noisy speech counterpart $\mathbf{y}$ are piecewise linearly related according to

$$\mathbf{x} = \mathbf{y} + \mathbf{r}(\mathbf{y}) \approx \mathbf{y} + \mathbf{r}_{\mathrm{i}(\mathbf{y})}. \qquad (1)$$

where $\mathrm{i}(\mathbf{y})$ is an index, to the correction vector $\mathbf{r}$, of the mixture component that $\mathbf{y}$ belongs to.

Given these assumptions, the cepstral enhancement algorithm using the MMSE rule gives:

$$\begin{aligned}\hat{\mathbf{x}} &= \int_{\mathbf{x}} \mathbf{x} p(\mathbf{x}|\mathbf{y}) d\mathbf{x} \approx \int_{\mathbf{x}} (\mathbf{y} + \mathbf{r}_{\mathrm{i}(\mathbf{y})}) p(\mathbf{x}|\mathbf{y}) d\mathbf{x} \\ &= \mathbf{y} + \int_{\mathbf{x}} \mathbf{r}_{\mathrm{i}(\mathbf{y})} p(\mathbf{x}|\mathbf{y}) d\mathbf{x} = \mathbf{y} + \int_{\mathbf{x}} \sum_{i} \mathbf{r}_{\mathrm{i}(\mathbf{y})} p(\mathbf{x}, \mathrm{i}|\mathbf{y}) d\mathbf{x} \qquad (2) \\ &= \mathbf{y} + \sum_{\mathrm{i}} \mathbf{r}_{\mathrm{i}(\mathbf{y})} \int_{\mathbf{x}} p(\mathbf{x}, \mathrm{i}|\mathbf{y}) d\mathbf{x} = \mathbf{y} + \sum_{\mathrm{i}} \mathbf{r}_{\mathrm{i}(\mathbf{y})} p(\mathrm{i}|\mathbf{y})\end{aligned}$$

The posterior probability above is computed from Bayes rule using the trained parameters in the mixture of Gaussians for $\mathbf{y}$. This MMSE rule generalizes the previous approximate-MAP rule in [3] by providing soft weights based on codeword (Gaussian component in the mixture) posterior probabilities rather than the 0-1 hard decision. (We found so far that the two decision rules perform similarly in speech recognition experiments.)

All the correction vectors, $\mathbf{r}_{i(y)}$, in Eqn. (2) are learned from stereo recordings for both the clean and noisy speech data. Minimizing the weighted square error of

$$E = \sum_t p(i|\mathbf{y}_t)(\hat{\mathbf{x}} - \mathbf{x})^2 = \sum_t p(i|\mathbf{y}_t)(\mathbf{y} + \mathbf{r}_{i(y)} - \mathbf{x})^2 \quad (3)$$

by setting $\frac{\partial E}{\partial \mathbf{r}_i} = 0$, we obtain the estimate:

$$\hat{\mathbf{r}}_{i(y)} = [\sum_t p(i \mid \mathbf{y}_t)]^{-1} \sum_t p(i \mid \mathbf{y}_t)(\mathbf{x}_t - \mathbf{y}_t), \quad (4)$$

where the summation is over all frames of the stereo training data.

3. CLUSTERING HMMs WITH SPLICE PROCESSING

To handle different types of noise, Noise Adaptive Training (NAT) was proposed in [3]. Like multi-style training, NAT pools all noise data together after applying noise reduction algorithms such as SPLICE, and trains a set of models that are robust across a wide range of noise types and levels. NAT has been found to work well in many cases [3]. However, when the characteristics of noise are very different, the performance of NAT often degrades. Therefore, we propose to use clustering techniques to increase the resolution of models.

In [5], subword-dependent speaker clustering was used to model speaker variation explicitly. It is different from traditional speaker clustering as the clustering on each subword or subphonetic unit could be different. This technique has been applied to improve the NAT model resolution in the current work. Since we know that the impact of noise and noise reduction on different phonetic units will be different, subword-dependent clustering will be able to model noise-reduced speech more efficiently.

The procedure for training subword-dependent NAT clustered models is as follows:

1. Train a set of initial single Gaussian context-dependent model for each noise condition (type and level).
2. For models under all noise conditions, use a bottom-up clustering technique to merge a pair of Gaussians (in the same senone) with a minimum likelihood loss over all the training data; repeat until a desired number of instances of senones is achieved.
3. Output the clustering information to a mapping table. The table contains the information of: a) how many instances each senone will be allocated and b) for each instance, which noise-conditioned data will contribute to it.
4. Based on the clustering mapping table, context-dependent Gaussian mixture models are trained.

The likelihood loss computation in the clustering step is carried out as follows. Assume we have two Gaussians $G1 = N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ with EM counts c_1 and $G2 = N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ with EM counts c_2, modeling the same set of data. When we merge these two Gaussians, the new Gaussian has the following EM count, mean, and variance:

$$c = c_1 + c_2 \quad (5)$$

$$\boldsymbol{\mu} = \frac{c_1\boldsymbol{\mu}_1 + c_2\boldsymbol{\mu}_2}{c} \quad (6)$$

$$\boldsymbol{\Sigma} = \frac{c_1\{\boldsymbol{\Sigma}_1 + (\boldsymbol{\mu}_1 - \boldsymbol{\mu})(\boldsymbol{\mu}_1 - \boldsymbol{\mu})^t\}}{c} + \frac{c_2\{\boldsymbol{\Sigma}_2 + (\boldsymbol{\mu}_2 - \boldsymbol{\mu})(\boldsymbol{\mu}_2 - \boldsymbol{\mu})^t\}}{c} \quad (7)$$

The likelihood loss over the data due to merging $G1$ and $G2$ is:

$$\Delta_{1+2} = \frac{c \log|\boldsymbol{\Sigma}| - c_1 \log|\boldsymbol{\Sigma}_1| - c_2 \log|\boldsymbol{\Sigma}_2|}{2} \quad (8)$$

For decoding with the subword-dependent clustered models, all instances of the subword unit is treated as parallel states in the topology. For example, with senone-dependent clustered models in our experiments here, all instances of the same senone are scored and the maximum likelihood is used as the score for that senone. Experimental results using this clustering technique will be presented in Section 4.4.

4. ROBUST SPEECH RECOGNITION EXPERIMENTS

A series of large vocabulary speech recognition experiments are carried out to evaluate the various improved and extended SPLICE methods discussed above. The baseline system uses a version of the Microsoft continuous-density HMMs (Whisper) [1][3]. The system uses 6000 tied HMM states and 20 Gaussians per state. All experiments use two-mean cepstrum normalization. The recognition task is 5000-word vocabulary, continuous speech recognition in the Wall Street Journal (WSJ) database. A fixed, bigram language model is used. The training set consists of 16,000 female sentences, and the test set of 167 female sentences. The noise added to the clean WSJ training and test speech data is collected from a number of live locations and sources, including restaurants, airport, lobby, noisy offices, cafeteria conversations (babbles), coughing, keyboard typing, water running, plane engine, plane cabinet, telephone dialing, etc. The baseline error rate under the clean acoustic environment is 4.87%.

4.1 Automatic detection of noise type and level

In contrast to the FCDCN method [1] where the acoustic-space partitioning is performed for clean speech ($\mathbf{x}$) via VQ, the current SPLICE method does the partitioning for noisy speech ($\mathbf{y}$). This requires that multiple codebooks or multiple Gaussian mixture models be trained for noisy speech corrupted by separate types and levels of noise. If the noise type and level can be reliably detected from the noisy data alone, then a most appropriate VQ codebook (or a mixture model) can be used for carrying out the SPLICE processing. This concept is similar to that of MFCDCN described in [6].

The simplest technique for the noise detection is to use the VQ distortion measure to score the noisy speech with unknown

noise type and level against a set of pre-trained codebooks and to choose the noise type and level based on the minimum VQ distortion. Such distortion values, with means and standard deviations (in parentheses) computed over 167 test sentences, are shown in Table 1. They encompass three types of noise (babble, office, and white noise) with two to three different SNR levels, and are randomly extracted from a much larger set of similar results we have obtained. It is observed that the minimum VQ distortion (bold figures) is very reliable for discriminating different noise types. Within a noise type, discrimination of noise levels is also reasonably good. This method of using the VQ distortion metric for automatic noise type and level detection is much more efficient than other methods using a multiple-state HMM for the noise statistics [7].

Code book / *Test data*	babble 10 dB	babble 20 dB	Office -10 dB	Office 0 dB	White 10 dB	White 15 dB	White 20 dB
Babble SNR 10 dB	**0.19** (0.02)	0.29 (0.03)	1.21 (0.06)	1.11 (0.06)	1.42 (0.08)	1.21 (0.11)	1.11 (0.12)
Babble 20 dB	0.56 (0.12)	**0.25** (0.02)	1.39 (0.09)	0.99 (0.08)	1.60 (0.13)	1.36 (0.08)	1.07 (0.08)
Office -10 dB	1.13 (0.05)	1.16 (0.04)	**0.15** (0.02)	0.21 (0.03)	1.94 (0.11)	1.85 (0.11)	1.83 (0.11)
Office 0 dB	1.11 (0.05)	0.96 (0.04)	0.39 (0.04)	**0.21** (0.02)	1.90 (0.14)	1.74 (0.10)	1.56 (0.09)
White 10 dB	1.40 (0.26)	1.23 (0.09)	4.07 (0.19)	3.78 (0.18)	**0.09** (0.07)	0.10 (0.07)	0.14 (0.07)
White 15 dB	1.28 (0.28)	1.16 (0.09)	3.63 (0.22)	3.39 (0.21)	0.16 (0.06)	**0.11** (0.07)	0.13 (0.07)
White 20 dB	1.13 (0.29)	1.06 (0.10)	3.20 (0.27)	2.97 (0.24)	0.42 (0.07)	0.20 (0.06)	**0.14** (0.06)

Table 1. VQ distortion (including standard deviation in parentheses) for each noisy test set against a range of VQ codebooks trained on a data set corrupted by three types of noise.

4.2 Results for in-task SPLICE processing

In-task SPLICE processing refers to the scenario where it is assumed that the noise type corrupting the test data has also been contained in the training set, both subject to the same SPLICE processing. Cross-task SPLICE processing does not require such an assumption. Given the highly reliable noise type and level detection using the VQ distortion metric already shown, the above "in-task" assumption should not cause serious difficulties in practical applications of SPLICE.

In Table 2 we list word error rates (percent accuracy WER) for 14 types of natural noise (column 1) with fixed SNR of 10 dB using three types of in-task SPLICE processing (columns 4-6):

a) **SPLICE test-only** --- clean speech models are used to score SPLICE-processed test data;
b) **SPLICE-SPLICE** --- both training and test data are subject to the same SPLICE processing; and
c) **NAT-SPLICE** --- multi-style training is used to train one single set of HMMs using all in-task, SPLICE-processed training data.

Note that for SPLICE-SPLICE, many sets of HMMs (one set for each noise condition) are needed, and for NAT-SPLICE only one set is needed.

For comparison purposes, we also list in Table 2 the WERs for the mismatched (column 2) and noisy matched (column 3) conditions. While the performance obtained with SPLICE processing only on the test data is short of that under the noisy matched condition, the use of the HMMs trained with SPLICE-processed data (SPLICE-SPLICE) significantly outperforms the latter. (In only one out of the 14 cases SPLICE-SPLICE is slightly worse.) Table 2 also shows that the use of NAT for the SPLICE-processed data across all 14 noise types gives very small performance degradation compared with the corresponding SPLICE-SPLICE performance. This suggests that as long as a sufficiently rich set of noisy speech data are used for SPLICE processing and NAT training, the recognizer remains robust for an unknown noise and channel distortion environment.

Exps / Noise	Mis-match	Noisy match	SPLICE test-only	SPLICE-SPLICE	NAT SPLICE
PhoneDial	6.99	6.46	6.68	6.17	6.13
Keyboard	16.80	10.41	11.37	7.50	7.94
Coughing	22.71	20.31	21.34	12.63	12.78
Engine	30.17	9.34	19.05	9.23	10.34
Cafeteria	12.44	6.79	9.60	6.87	7.83
LoudRoom	31.06	9.64	15.58	8.83	9.60
Airport	31.31	10.56	18.65	10.01	11.19
Restaurant	12.22	7.75	9.64	7.16	7.46
QuietRoom	14.81	7.02	10.97	6.87	8.27
Lobby	32.50	10.75	16.51	9.79	10.52
Water	33.79	10.08	13.70	8.46	9.25
Talk	36.15	12.08	23.79	11.23	11.89
PhoneDial2	6.50	6.43	6.46	5.58	6.24
Engine2	28.77	9.71	21.16	9.64	11.34
AVERAGE	**22.59**	**10.57**	**14.61**	**8.57**	**9.34**

Table 2. Table 2: WERs (%) for 14 types of natural noise (SNR= 10 dB) using various types of SPLICE processing.

For these 14 types of noise, we also evaluated a spectral subtraction (SS) technique, with its implementation described in [3], in place of SPLICE in an otherwise identical manner. For both the SS-test-only and SS-SS scenarios, the SS are shown to produce significantly more errors than its SPLICE counterpart.

4.3 Results for cross-task SPLICE processing

Despite the practical value of the SPLICE technique provided by the success of in-task NAT and by high accuracy of noise type detection, a most rigorous test of the SPLICE strength is to perform cross-task experiments where the noise types in the training set are disjoint from those used to corrupt the test set. We designed such experiments where the first eight types of

noise in Table 2 plus five additional types (synthetic white noise, office computer noise, babble sound, and roller coaster noise, which were described in [3]) were used to corrupt the training data. The remaining six types of noise in Table 2 were used to corrupt the test data. The WERs in the cross-task NAT-SPLICE experiments are listed in column 3 of Table 3, in comparison with the corresponding in-task WERs in column 2. Rather small performance degradation is observed going from in-task testing to cross-task testing. This demonstrates a highly desirable property of the SPLICE technique.

Exps / Noise Type	In-task NAT-SPLICE	Cross-task NAT-SPLICE	Cross-task Cluster-SPLICE
QuietRoom	8.27	8.60	9.05
Lobby	10.52	11.30	10.34
Water	9.25	9.27	9.05
Talk	11.89	14.00	12.56
PhoneDial2	6.24	5.98	6.35
Engine2	11.34	12.22	10.93
AVE.	**9.59**	**10.23**	**9.71**

Table 3. WER comparisons for (A) in-task and cross-task (columns 2 & 3) noise adaptive training using SPLICE processing; and (B) one-cluster versus two-cluster HMMs (columns 3 & 4) both with the same SPLICE processing.

4.4 Results for clustered HMMs with SPLICE processing

The experimental results for the clustering technique described in Section 3 are shown in the last column of Table 3. The number of clusters in the experiment is two. An average of 5% error rate reduction is achieved going from one cluster (NAT, column 2) to two clusters. The cross-task clustered models provide a performance close to the in-task non-clustered counterpart. The price paid for the 5% performance improvement is twice of the memory storage for the HMMs and somewhat higher cost in decoding.

5. DISCUSSION AND CONCLUSION

This paper describes our continuing work on noise robust speech recognition for the purpose of deploying the recognizers in realistic acoustic environments. The results reported in this paper have demonstrated that our new feature-domain processing technique of SPLICE has beaten the performance limit set by the conventional wisdom --- that is, "the best option when dealing with noisy speech would be to retrain the system so as to create the matched noisy condition". Our SPLICE experiments have produced an average of 19% lower WER than this "limit". In addition to the performance gain, our SPLICE technique is shown to be "practical" (via the use of the automatic noise type detection and of NAT). In contrast, the matched noisy condition retraining is unattainable in practice because the noise properties are typically unknown in advance.

As mentioned, one key issue for practical deployment of SPLICE is the choice of an appropriate VQ codebook for the SPLICE processing. The results shown in Section 4.1 for automatic noise type detection resolved this issue. While these results were obtained at the sentence level, an on-line version [4] showed similarly good results. Further, while all the results reported in this paper were obtained from the noisy speech data created by adding (natural) noise into the clean speech waves, use of live recorded noisy speech using our MiPad device have also produced similarly good results (for detailed experiments and results, see also [4]).

The SPLICE technique presented in this paper is expected to revive a class of stereo-based techniques [2] for robust speech recognition, which have been put into dormancy for many years. Two critical innovations responsible for this are the ideas of modeling residual noise from noise reduction and of noise adaptive training, both of which were presented in [3] recently.

While analyzing why the SPLICE technique has been able to consistently produce superior performance over that under the matched noisy condition, we observed that the MFCC distributions in the HMMs trained using the SPLICE-processed data are often significantly more separated across confusable phone classes than distributions trained with (matched) noisy data. This suggests that by explicitly forcing phonetic discrimination in training the HMMs jointly with training the SPLICE parameters, we can further enhance the phonetic discriminative power of our robust recognizer and hence its performance under adverse acoustic environments.

REFERENCES

[1] A. Acero, L. Deng, T. Kristjansson, and J. Zhang. "Noise HMM adaptation using vector Taylor series for noisy speech recognition," Proc.ICSLP, Oct. 2000, pp. 869-872.

[2] A. Acero. Acoustic and Environmental Robustness in Automatic Speech Recognition, Kluwer Academic, 1993.

[3] L. Deng, A. Acero, M. Plumpe, and X.D. Huang. "Large-vocabulary speech recognition under adverse acoustic environments," Proc. ICSLP, Oct. 2000, pp. 806-809.

[4] J. Droppo, A. Acero, and L. Deng, "Efficient on-line acoustic environment estimation for FCDCN in a continuous speech recognition system," Submitted to ICASSP 2001.

[5] L. Jiang and X.D. Huang, "Subword-dependent speaker clustering for improved speech recognition", Proc. ICSLP, Oct. 2000, Vol. III, pp. 137-140.

[6] F. H. Liu, R. Stern, A. Acero, and P. Moreno, "Environment normalization for robust speech recognition using direct cepstral comparison," Proc. ICASSP, 1991, pp. 893-896.

[7] H. Sameti, H. Sheikhzadeh, L. Deng, and R. Brennan. "HMM-based strategies for enhancement of speech embedded in non-stationary noise", IEEE Trans. Speech and Audio Processing, Vol.6, Sept.1998, pp. 445-455.