Evaluation of the SPLICE Algorithm on the Aurora2 Database

Jasha Droppo, Li Deng, and Alex Acero

Microsoft Research
One Microsoft Way
Redmond, WA, USA
{jdroppo,deng,alexac}@microsoft.com

Abstract

This paper describes recent improvements to SPLICE, Stereo-based Piecewise Linear Compensation for Environments, which produces an estimate of cepstrum of undistorted speech given the observed cepstrum of distorted speech. For distributed speech recognition applications, SPLICE can be placed at the server, thus limiting the processing that would take place at the client. We evaluated this algorithm on the Aurora2 task, which consists of digit sequences within the TIDigits database that have been digitally corrupted by passing them through a linear filter and/or by adding different types of realistic noises at SNRs ranging from 20dB to -5dB. For clean acoustic models, we achieve a 67.39% average decrease in word error rate over all test sets. For retrained multi-style acoustic models, the average decrease is 27.87%. The average relative word error rate reduction is 47.63%.

1. Introduction

There has been a great deal of interest recently in standardizing distributed speech recognition applications in which the user can have either a plain phone or a smart phone and speech recognition is done at a centralized server. Because of bandwidth limitations, one possibility is to have a cellular phone use a standard codec to transmit the speech to the server, which decompresses it and recognizes it. Since ASR systems only need some features of the speech signal, such as mel-cepstrum, more bandwidth can be saved by transmitting only those features. ETSI has been accepting proposals for Aurora [1], an effort to standardize a front-end for distributed speech recognition applications that offers low bitrate and is robust to noise and channel distortions.

In a distributed speech recognition system, the SPLICE technique described in this paper may either be applied within the front end on the client device, or on the server. Implementation on the server has several advantages. Computational complexity becomes less of an issue, and continuing improvements can be made that benefit devices already deployed in the field.

SPLICE is a frame-based bias removal algorithm for cepstrum enhancement under additive noise distortion, channel distortion, or a combination of the two. In [2] we reported the approximate MAP formulation of the algorithm, and more recently [3][4] described the MMSE formulation of the algorithm with a much wider range of naturally recorded noise including both artificially mixed speech and noise and naturally recorded noisy speech. In this paper, we report some new developments of the algorithm including temporal smoothing and noise mean normalization, and present full sets of evaluation results for AURORA2 digit-sequence recognition.

The SPLICE algorithm assumes no explicit noise model, and the noise characteristics are embedded in the piecewise linear mapping between the "stereo" clean and distorted speech cepstral vectors. The piecewise linearity is intended to approximate the true nonlinear relationship between the two. The nonlinearity between the cepstral vectors of clean and distorted (including additive noise) cepstra arises due to the use of the logarithm in computing the cepstra. Because of the use of the stereo training data that provide accurate estimates of the bias or correction vectors without the need for an explicit noise model, the SPLICE algorithm is potentially able to effectively handle a wide range of difficult distortions, including nonstationary distortion, joint additive and convolutional distortion, and even nonlinear distortion of the original time-series. A key requirement for the success of earlier versions of the SPLICE is that the distortion conditions under which the correction vectors are learned from the stereo data are similar to those that corrupt the test data. Our recent work on noise mean normalization has greatly reduced this requirement.

This organization of this paper is as follows. In Section 2, we give a brief review of the basic SPLICE algorithm. The extension of the basic SPLICE to its dynamic, temporally smoothed version is presented in Section 3. A method for making SPLICE work much better on unseen noise conditions is presented in Section 4. Full results of digit-sequence recognition for AURORA2 are presented and discussed in Section 5.

2. A Review of SPLICE

Given the general model of distortion from a clean cepstral vector, $\mathbf{x}$, into a noisy one, $\mathbf{y}$, we describe the probabilistic formulation of the basic (frame independent) version of the SPLICE algorithm below.

2.1. A Model of Speech and its Degradation

The first assumption is that the noisy speech cepstral vector follows the distribution of mixture of Gaussians:

$$\begin{aligned} p(\mathbf{y}) &= \sum_s p(\mathbf{y}|s)p(s), \text{ where} \\ p(\mathbf{y}|s) &= N(\mathbf{y}; \mu_s, \Sigma_s) \end{aligned}$$

The discrete state variable s denotes the discrete random variable taking the values 1, 2, ..., N, one for each region over which the piecewise linear approximation between the clean cepstral vector $\mathbf{x}$ and distorted cepstral vector is made. One distribution $p(\mathbf{y})$ is trained for each separate distortion condition (not indexed for clarity), and can be thought as a "codebook" with a total of N codewords (means) and their variances.

The second assumption made by the SPLICE is that the conditional probability density function (PDF) for the clean vector $\mathbf{x}$ given the noisy speech vector, $\mathbf{y}$, and the region index, s, is Gaussian whose mean vector is a linear transformation of the noisy speech vector $\mathbf{y}$. In this paper, we take a simplified form of this linear transformation by making the rotation matrix to be the identity matrix, leaving only the bias or correction vector. Thus, the conditional PDF is assumed to have the form,

$$p(\mathbf{x}|\mathbf{y}, s) = N(\mathbf{x}; \mathbf{y} + \mathbf{r}_s, \Gamma_s). \quad (1)$$

2.2. Cepstral Enhancement

One significant advantage of the above two basic assumptions made in the SPLICE is the inherent simplicity in deriving and implementing the rigorous MMSE estimate of clean speech cepstral vectors from their distorted counterparts. The MMSE is the following conditional expectation of clean speech vector given the observed noisy speech:

$$\hat{\mathbf{x}}_{\text{MMSE}} = E_x\left[\mathbf{x}|\mathbf{y}\right] = \sum_s p(s|\mathbf{y}) E_x\left[\mathbf{x}|\mathbf{y}, s\right]. \quad (2)$$

Using Eq. 1, it is clear that:

$$E_x\left[\mathbf{x}|\mathbf{y}, s\right] = \mathbf{y} + \mathbf{r}_s, \quad (3)$$

which, inserted into Eq. 2, results in

$$\hat{\mathbf{x}}_{\text{MMSE}} = \mathbf{y} + \sum_s p(s|\mathbf{y})\mathbf{r}_s, \quad (4)$$

so that the MMSE estimate of $\mathbf{x}$ is the noisy speech vector corrected by a linear weighted sum of all codeword-dependent bias vectors.

A faster implementation can be achieved by approximating the weights $p(s|\mathbf{y})$ according to

$$\hat{p}(s|\mathbf{y}) \cong \begin{cases} 1 & s = \text{argmax}_s p(s|\mathbf{y}) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

so that this approximation turns the MMSE estimate to the approximate MAP estimate [2] that consists of two sequential steps of operation. First, finding optimal codeword s using the VQ codebook based on the parameters (μ_s,Σ_s), and then adding the codeword-dependent vector $\mathbf{r}_s$ to the noisy speech vector. We have found empirically that the above VQ approximation does not appreciably affect recognition accuracy. All results presented in this paper use this approximation.

2.3. SPLICE Training

Since the noisy speech PDF $p(\mathbf{y})$ is assumed to be a mixture of Gaussians, the standard EM algorithm can be used to train μ_s and Σ_s on noisy speech. Initial values of the parameters are determined by a VQ clustering algorithm.

If stereo data is available, the parameters $\mathbf{r}_s$ of the conditional PDF $p(\mathbf{x}|\mathbf{y}, s)$ can be trained using the maximum likelihood criterion:

$$\mathbf{r}_s = \frac{\sum_n p(s|\mathbf{y}_n)(\mathbf{x}_n - \mathbf{y}_n)}{\sum_n p(s|\mathbf{y}_n)}, \text{ where} \quad (6)$$

$$p(s|\mathbf{y}_n) = \frac{p(\mathbf{y}_n|s)p(s)}{\sum_s p(\mathbf{y}_n|s)p(s)} \quad (7)$$

where this training procedure requires a set of stereo (two channel) data. One channel contains the clean utterance, and the other contains the same utterance with distortion, where the distortion represented by the correction vectors is estimated above. The two-channel data can be collected, for example, by simultaneously recording utterances with one close-talk and one far-field microphone.

For the Aurora work reported in this paper, the SPLICE parameters were trained using identical utterances from the clean training set and the multistyle training set. This effectively tunes our cepstral enhancement parameters on the noise types from set A, keeping sets B and C as unseen conditions.

Note that the correction vectors $\mathbf{r}_s$ can also be estimated without the need of stereo data, at the expense of modest loss in accuracy [5].

2.4. Environmental Model Selection

The SPLICE algorithm described so far requires that the mixture of Gaussians for the noisy speech be conditioned on a specific noise type and level. To satisfy this requirement, we developed an effective on-line environmental selection method, which has been described in detail in [4].

We apply this method to the AURORA2 evaluation as follows. Seventeen separate mixture models are trained, one for each of the combinations of noise type and level in the multicondition training set. The on-line decision for selecting the environmental model e is made by first producing a local estimate of $p(\mathbf{y}_i|e)$ and then smoothing it over time.

2.5. Blind Equalization

In principle, when the training data for the SPLICE contain similar convolutional distortions to those in the test data, the methods described thus far can effectively remove that distortion. However, for data in set C, the convolutional distortion is unknown, so the stereo data needed for Eq. 6 is unavailable.

To account for this possible discrepancy between training and testing data, all of the experiments reported in this paper use a simple offline cepstral mean normalization (CMN) procedure. After each utterance is processed, we subtract from each frame the mean cepstrum computed over the entire utterance.

This procedure is of course not optimal, but increases the performance on set C dramatically.

Also under investigation are joint optimization techniques which integrate blind equalization directly into SPLICE. In principle, using the speech model already present in SPLICE should produce even better results.

3. Dynamic SPLICE

In this section, we present a new version of SPLICE that not only minimizes the static deviation from the clean to noisy cepstral vectors (as in the basic version of the SPLICE described in Section 2), but also seeks to minimize the deviation between the delta parameters.

The basic SPLICE (optimally) processes each frame of noisy speech independently. An obvious extension is to jointly process a segment of frames. In this way, although the deviation from the clean to noisy speech cepstra for an individual frame may be undesirably greater than that achieved by the basic, static SPLICE, the global deviation that takes into account the differential frames and the whole segment of frames can be reduced compared with the basic SPLICE.

We have implemented the above idea of dynamic SPLICE by temporally smoothing the bias vectors obtained from the basic, static SPLICE described in Section 2. This is an empir-

ical way of implementing the rigorous solution which would use a more realistic model for the time-evolution of the clean speech dynamics. Using the discrete state, we would model $p(\mathbf{x}_n|\mathbf{y}_n, s_n, s_{n-1})$, or using the continuous clean speech vector estimate we would model $p(\mathbf{x}_n|\mathbf{y}_n, s_n, \mathbf{x}_{n-1})$.

An efficient way to implement an approximate dynamic SPLICE, as is used in the current AURORA2 evaluation, is to independently time-filter each component of the cepstral bias vector $\mathbf{r}_{s_n}$. We have achieved significant performance gains using this efficient heuristic implementation.

In our specific implementation, we used a simple zero-phase, non-causal, IIR filter to smooth the cepstral bias vectors. This filter has a low-pass characteristic, with the system transfer function of

$$H(z) = \frac{-0.5}{(z^{-1} - 0.5)(z - 2)}. \quad (8)$$

This transfer function is the result of defining an objective function of the summation of the static and dynamic deviations from clean speech to noisy speech vectors. The optimal solution that minimizes this objective function is of the form of Eq. 8, where the constants are functions of the variances of our model.

In preliminary testing, using Eq. 8 instead of the exact solution produces similar results, at a lower computational cost. The results presented in this paper use this approximation rather than the rigorous solution.

4. Noise Mean Normalization

The SPLICE mapping from the noisy cepstrum $\mathbf{y}$ to the cleaned cepstrum $\hat{\mathbf{x}}$ described so far depends directly on the type and level of noise added to the clean speech. In this section, we describe an enhancement which we call Noise Mean Normalization (NMN).

Incorporating NMN into SPLICE decreases the dependency of the SPLICE mapping on the noise statistics. By employing NMN SPLICE, recognition performance on unseen noise types should improve, which is validated by experimental results in Section 5.

4.1. Procedure

Previously, SPLICE has implicitly characterized the relationship between $\mathbf{x}$ and $\mathbf{y}$ for a given noise type in the training set and assumed similar relationships existed in the testing set. This assumption is often invalid, since the testing set noise can be quite different from the training set noise (e.g. set A versus set B). NMN SPLICE instead characterizes the relationship between $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$, where

$$\begin{aligned} \bar{\mathbf{x}} &= \mathbf{x} - \mu \\ \bar{\mathbf{y}} &= \mathbf{y} - \mu, \text{ and} \\ \mu &= \text{predicted value of } \mathbf{n}. \end{aligned}$$

For every frame of the training and testing data, a noise estimate μ is created. It has been our experience that even simple estimates, such as taking the mean of the first ten frames of the utterance, show improvements over unmodified versions of SPLICE.

The training procedure differs slightly from Section 2.1. Instead of modeling $p(\mathbf{y})$, we instead build a prior distribution for $\bar{\mathbf{y}} = \mathbf{y} - \mu$.

The cepstral enhancement procedure consists of first finding $\bar{\mathbf{x}}$ from $\bar{\mathbf{y}}$ using the SPLICE mapping, and then computing the clean speech estimate as the sum of μ and $\bar{\mathbf{x}}$.

4.2. Preliminary Analysis

When the training and testing noises are identical, and the estimate $\mu = \mathbf{0}$, NMN SPLICE is identical to SPLICE. As the estimate μ improves, there is less uncertainty about the noise in each frame, and word error rate decreases.

For the cross-condition case, where training and testing noises are dissimilar, our analysis of the behavior of NMN SPLICE considers the limiting case of low instantaneous signal to noise ratio (SNR) regions of the utterance, which account for most of the errors.

In regions of low SNR, the noisy signal $\mathbf{y}$ will consist primarily of noise. If the noise estimation were perfect, then the transformed input to NMN SPLICE would be

$$\bar{\mathbf{y}} = \log(1 + \exp(\mathbf{x} - \mathbf{n})) \approx \exp(\mathbf{x} - \mathbf{n}).$$

In contrast, SPLICE without NMN has the following relationship for low SNR:

$$\mathbf{y} \approx \exp(\mathbf{x} - \mathbf{n}) + \mathbf{n}.$$

Clearly, when $\mathbf{n} \gg \mathbf{x}$, then $\bar{\mathbf{y}}$ is much less dependent than $\mathbf{y}$ on the value of $\mathbf{n}$. Consequently, in low SNR regions, a codebook built using $\bar{\mathbf{y}}$ is less sensitive to the noise type compared to a codebook built on $\mathbf{y}$.

Therefore, in these low SNR regions, the testing data will closely resemble the training data, and the noise will be appropriately suppressed.

4.3. Noise Estimation

A necessary component, therefore, of NMN SPLICE is the noise estimation algorithm. First, in Section 5.2, we explore an upper bound for the performance of NMN splice on this task. We derive the actual noise signals for each utterance from the clean and noisy data available in the test set. From these noise signals, the true noise cepstra are computed.

Then, in Section 5.3 we provide results where the noise estimate is generated using iterative stochastic approximation[6]. This method employs strong speech and noise models, resulting in high quality noise estimates.

5. Experimental Results Using AURORA2

The speech recognition results reported in this paper are produced by the reference Aurora front-end version 2.0, using c_0 instead of log energy, and modified to use power spectral density instead of magnitude spectrum in its computations. We found this configuration to be slightly superior to the default.

All experiments unless otherwise noted were performed using the blind equalization technique described in Section 2.5 and the smoothing technique of Section 3.

SPLICE codebooks and correction vectors were trained using similar utterances from the clean and multi-sytle training sets. The noisy speech model consisted of a mixture of 256 Gaussians with diagonal covariance matrices, though we have observed improved accuracy for some nonstationary noise types with more Gaussians.

5.1. SPLICE

Figure 1 is a summary of the full results for the SPLICE on the Aurora2 corpus, in the absence of noise mean normalization.

Since the SPLICE parameters were trained on fixed noise conditions that are included in Set A, this set exhibits the best

Aurora 2 Multicondition Training - Results

	A					B					C				Percentage
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	Overall	Improvement
Clean	98.83	98.85	98.69	98.98	98.84	98.83	98.85	98.69	98.98	98.84	98.93	99.00	98.97	98.86	22.77%
20 dB	98.53	98.40	98.42	98.40	98.44	98.25	97.76	98.30	98.27	98.15	98.43	98.10	98.27	98.29	34.01%
15 dB	97.82	97.55	98.09	97.56	97.76	96.71	97.37	96.09	96.67	96.71	97.54	96.52	97.03	97.19	24.05%
10 dB	96.13	95.22	96.63	95.50	95.87	94.07	94.53	91.59	92.35	93.14	94.66	93.56	94.11	94.42	10.44%
5 dB	92.82	86.88	92.07	90.59	90.59	83.70	83.37	82.20	80.28	82.39	85.57	82.01	83.79	85.95	4.33%
0 dB	79.89	62.67	77.21	76.40	74.04	54.65	55.17	54.70	53.66	54.55	61.47	52.63	57.05	62.85	8.62%
-5dB	48.54	21.80	44.32	51.25	41.48	10.99	19.59	10.59	17.09	14.57	26.59	17.84	22.22	26.86	2.72%
Average	93.04	88.14	92.48	91.69	91.34	85.48	85.64	84.58	84.25	84.98	87.53	84.56	86.05	87.74	
	38.08%	1.61%	44.23%	30.58%	28.92%	0.58%	-10.79%	-24.81%	-5.10%	-9.37%	25.60%	1.62%	14.00%		9.92%

Aurora 2 Clean Training - Results

	A					B					C				Percentage
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	Overall	Improvement
Clean	98.99	99.15	99.08	99.07	99.07	98.99	99.15	99.08	99.07	99.07	98.99	99.21	99.10	99.08	3.77%
20 dB	98.10	98.43	98.54	98.30	98.34	98.59	97.55	98.69	98.40	98.31	97.94	97.85	97.90	98.24	62.46%
15 dB	96.78	97.40	97.85	96.95	97.25	97.51	96.37	97.26	97.13	97.07	96.38	96.22	96.30	96.99	74.79%
10 dB	93.71	93.71	94.93	93.30	93.91	94.29	92.53	93.29	92.22	93.08	91.83	91.11	91.47	93.09	77.96%
5 dB	87.81	81.80	86.73	84.57	85.23	81.73	79.23	81.51	78.43	80.23	77.99	76.39	77.19	81.62	69.32%
0 dB	67.76	53.93	63.91	66.74	63.09	51.61	50.70	52.79	51.59	51.67	47.41	44.95	46.18	55.14	45.67%
-5dB	35.25	20.01	30.93	36.44	30.66	12.93	20.62	12.02	17.34	15.73	21.49	16.72	19.11	22.38	15.15%
Average	88.83	85.05	88.39	87.97	87.56	84.75	83.28	84.71	83.55	84.07	82.31	81.30	81.81	85.01	
	63.40%	70.18%	70.54%	65.24%	67.83%	67.82%	56.54%	67.29%	62.93%	64.00%	47.72%	44.82%	46.27%		62.48%

Figure 1: *SPLICE results on AURORA (Without NMN)*

NMN SPLICE-True Noise

Absolute performance				
Training Mode	Set A	Set B	Set C	Overall
Multicondition	96.48	97.22	96.27	96.74
Clean Only	93.81	95.58	93.47	94.45
Average	95.15	96.40	94.87	**95.59**

Performance relative to Mel-cepstrum				
Training Mode	Set A	Set B	Set C	Overall
Multicondition	71.12%	79.77%	77.00%	76.01%
Clean Only	84.00%	90.01%	80.72%	86.11%
Average	77.56%	84.89%	78.86%	**81.06%**

Figure 2: *NMN SPLICE with true noise cepstra*

performance among all the three sets. While Set A only contains four types of noises, we have experimented up to fourteen noise types, giving similarly good results on a Wall Street Journal task. (We call the algorithm applied to this experimental setup as in-task or in-condition SPLICE in [3].)

To examine the SPLICE's ability to perform in unseen noise conditions, we applied the SPLICE parameters developed for set A, without modification, to enhance the cepstra in sets B and C. This experimental setup does not allow the in-condition SPLICE to apply. We called this more difficult experimental setup, where the noisy condition in the stereo training data is unseen in the test data, the cross-task or cross-condition SPLICE in [3]. From the results in Figure 1, we observe reasonable performance improvement over the baseline, consistent for all noise conditions, when using the clean acoustic model. This improvement is less than that achieved for set A. This indicates that the bias vectors learned from set A's stereo data may not be representative of those required to transform the noisy speech to clean speech in set B.

5.2. NMN SPLICE–Perfect Knowledge

For this first NMN result, we consider the case where the noise cepstrum is known for each frame of data. Note that even in this case, it is impossible to infer the clean speech to an arbitrary precision, because there is still uncertainty in the mixing of speech and noise.

Figure 2 was produced by computing the true noise cepstral sequence for each utterance, and using those values to normalize frames before processing with SPLICE. Notice that the relative improvement for set B is even higher than that for set A. The addition of NMN to SPLICE not only makes it more robust to unseen noise types, NMN allows SPLICE to outperform itself even on these unseen noises.

This result, which uses a perfect estimate of the noise, provides us with an upper bound we can not expect to exceed using NMN SPLICE.

5.3. NMN SPLICE–Iterative Stochastic Approximation

We recently developed and implemented a novel technique which uses iterative stochastic approximation and the "forgetting" mechanism to effectively track nonstationary noise. Using a number of empirically verified assumptions associated with the implementation simplification, the efficiency of this algorithm has been improved to close to real time for noise tracking. The mathematical theory, algorithm, and implementation detail of this iterative stochastic approximation technique will be written and submitted to ASRU01 [6].

Figure 3 contains the best results we have obtained so far using the iterative stochastic approximation for nonstationary noise estimation in the framework of NMN SPLICE discussed in this paper.

As expected, when we use this fast on-line estimate of noise cepstra, performance on set A increases slightly and performance on set B increases dramatically.

6. Summary and Discussion

The SPLICE algorithm, as described in this paper, is an efficient algorithm that can be run either on the client or the server in a distributed speech recognition system. It models cepstra of noisy speech as a mixture of Gaussians. We can leverage this model to identify the type of corruption currently being encountered, and to compensate for an unknown linear filter. By incor-

Aurora 2 Multicondition Training - Results

	A					B					C				Percentage
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	Overall	Improvement
Clean	98.59	98.70	98.69	98.95	98.73	98.59	98.70	98.69	98.95	98.73	98.65	98.82	98.74	98.73	13.58%
20 dB	98.53	98.64	98.51	98.64	98.58	98.46	97.91	98.60	98.58	98.39	98.40	98.25	98.33	98.45	40.27%
15 dB	97.64	98.07	98.33	97.69	97.93	97.79	97.49	97.44	97.47	97.55	97.88	97.16	97.52	97.70	36.95%
10 dB	95.98	96.37	96.84	95.65	96.21	95.27	94.41	95.11	95.12	94.98	95.79	93.80	94.80	95.43	25.78%
5 dB	92.08	88.94	92.78	90.25	91.01	87.63	88.06	88.16	87.04	87.72	90.97	85.85	88.41	89.18	25.01%
0 dB	78.02	65.57	76.83	74.42	73.71	65.37	68.23	69.49	65.57	67.17	72.67	65.42	69.05	70.16	26.06%
-5dB	46.02	26.12	36.53	43.13	37.95	26.22	34.34	32.12	30.70	30.85	38.23	30.08	34.16	34.35	12.82%
Average	92.45	89.52	92.66	91.33	91.49	88.90	89.22	89.76	88.76	89.16	91.14	88.10	89.62	90.18	
	32.85%	13.01%	45.52%	27.57%	30.15%	24.04%	16.83%	17.14%	24.99%	21.05%	47.14%	24.13%	36.01%		27.87%

Aurora 2 Clean Training - Results

	A					B					C				Percentage
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	Overall	Improvement
Clean	99.11	99.12	99.11	99.04	99.10	99.11	99.12	99.11	99.04	99.10	99.14	99.12	99.13	99.10	6.11%
20 dB	98.16	98.52	98.72	98.27	98.42	98.65	97.58	98.81	98.70	98.44	98.34	98.04	98.19	98.38	65.20%
15 dB	96.65	97.64	98.09	96.61	97.25	97.88	96.89	97.97	97.84	97.65	96.81	96.40	96.61	97.28	76.33%
10 dB	93.77	94.68	95.71	93.09	94.31	94.75	93.44	95.85	94.60	94.66	93.18	91.23	92.21	94.03	80.42%
5 dB	87.47	84.46	88.46	85.53	86.48	85.08	83.71	87.03	84.94	85.19	84.31	80.35	82.33	85.13	75.04%
0 dB	65.92	57.13	63.67	63.78	62.63	59.72	57.83	63.11	57.42	59.52	59.23	52.90	56.07	60.07	51.64%
-5dB	32.42	21.25	23.80	32.58	27.51	22.75	24.58	27.83	23.94	24.78	27.17	21.37	24.27	25.77	18.82%
Average	88.39	86.49	88.93	87.46	87.82	87.22	85.89	88.55	86.70	87.09	86.37	83.78	85.08	86.98	
	61.96%	73.03%	71.90%	63.75%	68.48%	73.03%	63.33%	75.52%	70.02%	70.83%	59.73%	52.14%	55.93%		67.39%

Figure 3: *NMN SPLICE with "iterative stochastic approximation" noise estimate*

porating both the dynamic and NMN modifications, the word error rate decreases significantly across both seen and unseen distortion conditions.

One significant contribution of this work is to show that as long as the noise estimation is reasonably performed, the NMN SPLICE can achieve high accuracy in even unseen noise conditions. In particular, the noise estimation methods described in this paper have not taken into account the convolutional distortion. Nevertheless, the results for set C containing convolutional distortion are comparable to their undistorted counterparts.

Our current work involves improving the noise estimation algorithm to further enhance the performance of NMN SPLICE. We are also investigating direct parametric methods for noise removal[7].

7. References

[1] H. G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy condidions," in *ISCA ITRW ASR2000 "Automatic Speech Recognition: Challenges for the Next Millennium"*, Paris, France, September 2000.

[2] Li Deng, Alex Acero, Mike Plumpe, and X.D. Huang, "Large vocabulary continuous speech recognition under adverse conditions," in *Proceedings of the ICSLP*, Beijing, October 2000, vol. 3, pp. 806–809.

[3] Li Deng, Alex Acero, Li Jiang, Jasha Droppo, and X.D. Huang, "High-performance robust speech recognition using stereo training data," in *Int. Conf. On Acoustics, Speech and Signal Processing*, Salt Lake City, Utah, May 2001.

[4] Jasha Droppo, Alex Acero, and Li Deng, "Efficient on-line acoustic environment estimation for FCDCN in a continuous speech recognition system," in *Int. Conf. On Acoustics, Speech and Signal Processing*, Salt Lake City, Utah, May 2001.

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

[6] Li Deng, Jasha Droppo, and Alex Acero, "Recursive estimation of nonstationary noise using a nonlinear model with iterative stochastic approximation," Submitted to *Proceedings of Automatic Speech Recognition and Understanding*, Italy, December 2001.

[7] Brendan Frey, Li Deng, Alex Acero, and Trausti Kristjansson, "ALGONQUIN: Iterating Laplace's method to remove multiple types of noise and channel distortion from log-spectra in robust speech recognition," in *Eurospeech*, 2001.