EVALUATION OF SPLICE ON THE AURORA 2 AND 3 TASKS

Jasha Droppo, Li Deng, and Alex Acero

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

ABSTRACT

Stereo-based Piecewise Linear Compensation for Environments (SPLICE) is a general framework for removing distortions from noisy speech cepstra. It contains a non-parametric model for cepstral corruption, which is learned from two channels of training data.

We evaluate SPLICE on both the Aurora 2 and 3 tasks. These tasks consist of digit sequences in five European languages. Noise corruption is both synthetic (Aurora 2) and realistic (Aurora 3).

For both the Aurora 2 and 3 tasks, we use the same training and testing procedure provided with the corpora. By holding the back-end constant, we ensure that any increase in word accuracy is due to our front-end processing techniques.

In the Aurora 2 task, we achieve a 76.86% average decrease in word error rate with clean acoustic models, and an overall improvement of 62.63%. For the Aurora 3 task, we achieve a 75.06% average decrease in word error rate for the high-mismatch experiment, and an overall improvement of 47.19%.

1. INTRODUCTION

The Aurora tasks [1] focus on noise-robust distributed speech recognition applications, in which the user has either a plain phone or a smart phone, and speech recognition may be performed by a centralized server. ETSI is in the process of standardizing a front-end and noise robustness techniques for these applications that offer low bit-rate and robustness to noise and channel distortions [2].

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 noise removal algorithm for cepstral enhancement in the presence of additive noise, channel distortion, or a combination of the two. We have previously presented a rigorous MMSE formulation of the algorithm, as well as presenting accuracy results on several tasks with synthetic training data, and both synthetic and realistic test data [3, 4].

In this paper, we report minor new developments of the algorithm and present full sets of evaluation results for the Aurora 2 and 3 noisy digit recognition tasks. The Aurora 3 results represent the first time SPLICE has been successfully trained and tested on non-synthetic signals.

The organization of this paper is as follows. In Section 2, we give a brief review of the basic SPLICE algorithm together with the modifications used in the current implementation. Full experimental results for the Aurora 2 and 3 noisy digit recognition tasks are presented and discussed in Section 3. We summarize in Section 4 that SPLICE works effectively on these corpora, and outline areas for improvement.

2. A REVIEW OF SPLICE

SPLICE is a general framework used to model and remove the effect of any consistent degradation of speech cepstra. It learns a joint probability distribution of noisy and clean cepstra, and uses this distribution to infer clean speech estimates from noisy inputs.

Because it takes noisy cepstra as input, and outputs clean cepstral estimates, SPLICE can be characterized as a cepstral noise removal function. Other algorithms construct rigid functions based on approximations to known causes of distortion, such as additive noise and linear convolutional channels. By contrast, SPLICE does not include any assumptions about how noisy cepstra are produced from clean cepstra, and can model any combination of these affects as well as others, including nonlinear and possibly non-stationary distortions.

The parameters of SPLICE's joint probability distribution are learned from simultaneous recordings of clean and noisy speech. The cepstral degradation is embedded in the statistical relationship between these two channels. SPLICE requires the training data to be partitioned into sets of utterances with similar corruption. In the case of unlabeled training data, unsupervised clustering can be used.

2.1. A Bayesian Formulation

One way to model the joint probability distribution function between the clean speech $\mathbf{x}$ and the distorted speech $\mathbf{y}$ is to train both $p(\mathbf{y})$ and $p(\mathbf{x}|\mathbf{y})$, and combine them to form the joint probability $p(\mathbf{x}, \mathbf{y})$.

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{y}).$$

This approach is, however, infeasible because $p(\mathbf{x}|\mathbf{y})$ will have parameters which are non-linear functions of $\mathbf{y}$.

Instead, SPLICE introduces an auxiliary discrete random variable s, which partitions the acoustic space into local regions, where the relationship between $\mathbf{x}$ and $\mathbf{y}$ is approximately linear within

each region:

$$\begin{aligned} p(\mathbf{x},\mathbf{y}) &= \sum_s p(\mathbf{x}|\mathbf{y},s)p(\mathbf{y}|s)p(s) \\ p(\mathbf{x}|\mathbf{y},s) &= N(\mathbf{x};\mathbf{y}+\mathbf{r}_s,\Gamma_s) \\ p(\mathbf{y}|s) &= N(\mathbf{y};\mu_s,\Sigma_s). \end{aligned}$$

Under this piecewise linear assumption, $p(\mathbf{y})$ is a Gaussian mixture model for the noisy cepstra, and $p(\mathbf{x}|\mathbf{y},s)$ describes a linear rule for producing $\mathbf{x}$ from $\mathbf{y}$. For each local region defined by s, the rule is to take $\mathbf{y}$ and add a correction vector $\mathbf{r}_s$, which produces an estimate of $\mathbf{x}$ with expected variance Γ_s.

2.2. Cepstral Enhancement

One significant advantage of the piecewise linear assumption is the inherent simplicity in deriving and implementing a rigorous minimum mean squared error (MMSE) estimate of clean speech cepstral vectors from their distorted counterparts. The estimate is the conditional expectation of clean speech vector given the observed noisy speech:

$$\begin{aligned} \hat{\mathbf{x}} &= E_x\left[\mathbf{x}|\mathbf{y}\right] \\ &= \sum_s p(s|\mathbf{y})E_x\left[\mathbf{x}|\mathbf{y},s\right] \\ &= \mathbf{y}+\sum_s p(s|\mathbf{y})\mathbf{r}_s. \end{aligned}$$

The MMSE estimate of $\mathbf{x}$ is the noisy speech vector corrected by a linear weighted sum of all mixture component dependent correction vectors.

2.3. Time Smoothing

Although SPLICE processes each frame independently, we know that the output cepstra should have a smoothness constraint. A rigorous smoothness constraint can be defined, but we have seen similar accuracy result from imposing a fixed, empirical filter on the correction vector sequence. For this paper, we used a simple zero-phase, non-causal, IIR filter to smooth the correction vector sequence as described in [5]:

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

2.4. SPLICE Training

We train one complete set of SPLICE parameters for each type of distortion. The noisy speech model $p(\mathbf{y})$ generally contains 256 diagonal Gaussian mixture components, and is trained from noisy data using standard EM techniques.

The parameters $\mathbf{r}_s$ of the conditional PDF $p(\mathbf{x}|\mathbf{y},s)$ can be trained using the maximum likelihood criterion.

$$\mathbf{r}_s = \frac{\sum_n p(s|\mathbf{y}_n)(\mathbf{x}_n-\mathbf{y}_n)}{\sum_n p(s|\mathbf{y}_n)}, \text{ where} \quad (1)$$

$$p(s|\mathbf{y}_n) = \frac{p(\mathbf{y}_n|s)p(s)}{\sum_s p(\mathbf{y}_n|s)p(s)}. \quad (2)$$

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.

2.5. Noise Mean Normalization

Early versions of SPLICE require that the distortions present in the test data match those under which SPLICE was trained. The noise mean normalization (NMN) technique is used to mitigate this effect. In [5], we demonstrate NMN improving recognition accuracy in both mismatched and matched testing conditions.

For every frame of the training and testing data, a noise estimate $\mu^{(n)}$ is created and used to normalize the inputs and outputs of SPLICE. 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. Another benefit is that as the noise estimate improves, even the accuracy of the matched testing conditions improve.

The mathematical theory, algorithm, and implementation detail of the noise estimation technique used for this paper were presented in [6], with more thorough treatment in [7]. The noise estimation algorithm uses iterative stochastic approximation (ISA) and a forgetting mechanism to effectively track non-stationary noise, and is quite robust. All of the parameters, including the forgetting factor, number of iterations, and the variance-dependent learning rate, are identical to those used for Aurora 2 without any tuning for Aurora 3.

2.6. Unsupervised Noise Clustering

If the training set is well labeled with noise type and SNR, as with Aurora 2, partitioning the data to create SPLICE parameters is simple. One set of parameters can be trained for each unique noise label. When the training set is not well labeled, as with Aurora 3, finding a suitable partition of the data is not as simple.

To find the hidden noise classes within the training data, we use a simple form of unsupervised clustering. The beginning and ending of every utterance not occurring in any test set is gathered, and used to train a Gaussian mixture model with six components:

$$p(\mathbf{y}) = \sum_m p(\mathbf{y}|m)p(m) = \sum_m N(\mathbf{y};\nu_m,\psi_m^2)p(m).$$

After training is complete, we use the mixture model, together with Bayes' rule, to choose the best label for each utterance in the training set, according to

$$\hat{m} = \arg\max_m p(m|\mathbf{y}) = \arg\max_m \frac{p(\mathbf{y}|m)p(m)}{\sum_{m'} p(\mathbf{y}|m')p(m')}.$$

These labels partition the training data into the six disjoint sets needed to create six sets of SPLICE parameters for each language.

2.7. Environment Detection

SPLICE conditions its processing on the presence of a known type of distortion, which has been encountered during training, but the testing data consists of unlabeled utterances. To find the appropriate set of SPLICE parameters (a codebook and associated correction vectors) for a given utterance, we evaluate $p(\mathbf{y})$ for each parameter set, and use the most likely condition. The details of this technique were presented in [4].

2.8. Blind Equalization

To account for possible discrepancies in linear channel 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 Aurora 2 Set C dramatically. This is to be expected, because this set is meant to test performance in the presence of a linear convolutional distortion not seen in the training data. CMN also increases the word accuracy on Aurora 3. The data was presumably collected using a variety of automobiles and microphones, so this is also expected.

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.

2.9. Endpointing

The baseline results provided for this evaluation pre-process every utterance, eliminating all but 200ms of non-speech at each end of every utterance. Since this paper is concerned with noise removal in the presence of speech, and not voice activity detection, we attempt to match this perfect endpointing algorithm as closely as possible.

For each language in the Aurora 3 task, we start with an acoustic model trained only on close-talk microphone data. This model is used to perform a forced alignment of all of the close-talk data in the corpus. Finally, these alignments are used to eliminate all but at most 200ms of non-speech from the beginning and end of each utterance. When using the same front end as the baseline (WI007), the resulting average word error rate of 23.33% is very close to the reference 23.48%.

3. EXPERIMENTAL RESULTS

All speech recognition results reported in this paper use HMMs trained in the manner prescribed by the scripts included with the Aurora tasks. When training is complete, we have 16-state whole-word models for each digit in addition to the "sil" and "sp" models. The Aurora 2 models are the standard "complex back-end" with 20 diagonal Gaussian mixture components in each state, and the Aurora 3 models have 3 diagonal Gaussian mixture components.

The cepstral features used in this paper were produced by the reference WI007 front end, with two modifications. These modifications are necessary to maintain compatibility with our in-house noise estimation software. The WI007 baseline uses a log frame energy feature, and computes cepstra based on the magnitude frequency spectrum. We replace these with the DC cepstral coefficient c_0, and the power spectral density.

3.1. Aurora 2

The Aurora 2 task consists of recognizing English digits in the presence of additive noise and linear convolutional distortion. These distortions have been synthetically introduced to clean (TI-Digits) data. Three test sets measure performance against noise types similar to those seen in the training data (set A), different from those seen in the training data (set B), and with an additional convolutional channel (set C).

The Aurora 2 acoustic model training data is perfectly suited for learning the SPLICE parameters. The clean acoustic model training data consists of 8440 utterances. The multi-style acoustic model training data consists of the same utterances synthetically mixed with four different noise types at varying amplitudes, for a total of 17 unique noise conditions. We trained one SPLICE codebook and corresponding correction vectors for each of the 17 conditions.

Table 1. Aurora 2 word error rate results. NMN SPLICE with "iterative stochastic approximation" noise estimate.

Aurora 2 Reference Word Error Rate				
	Set A	**Set B**	**Set C**	**Overall**
Multi	11.93%	12.78%	15.44%	**12.97%**
Clean	41.26%	46.60%	34.00%	**41.94%**
Average	**26.59%**	**29.69%**	**24.72%**	**27.46%**

Aurora 2 Word Error Rate				
	Set A	**Set B**	**Set C**	**Overall**
Multi	6.66%	8.99%	7.85%	**7.83%**
Clean	11.67%	12.25%	13.67%	**12.30%**
Average	**9.17%**	**10.62%**	**10.76%**	**10.07%**

Aurora 2 Relative Improvement				
	Set A	**Set B**	**Set C**	**Overall**
Multi	50.58%	43.80%	53.23%	**48.40%**
Clean	76.96%	80.39%	69.59%	**76.86%**
Average	**63.77%**	**62.10%**	**61.41%**	**62.63%**

Table 1 presents accuracy results for Aurora 2, using NMN SPLICE and noise estimates from the iterative stochastic approximation algorithm [6].

The improvement over the baseline using clean acoustic models is quite dramatic, with a 76.86% reduction in word error rate. When using multi-style acoustic models, we can still reduce the error by 48.40%. The performance for Set C is comparable to the other sets, indicating that this technique is robust to the linear convolutional distortion present in Set C.

3.2. Aurora 3

The Aurora 3 task consists of recognizing Danish, German, Spanish, and Finnish digits in realistic automobile environments.

Each utterance in the corpus consists of two recordings, and is labeled as coming from either a high, low, or quiet noise environment. Each pair of recordings was made simultaneously on a close-talk microphone and a hands-free, far-field microphone. Unlike the Aurora 2 task, which employed artificially created distortions, here each recording contains realistic channel, noise, and reverberation effects. Although they do not contain perfectly clean signals, the close-talk microphone recordings should always exhibit a greater signal-to-noise ratio than the far-field microphone recordings. Hence, we use the close-talk recordings as "clean speech" when training SPLICE.

Three experiments are defined for the evaluation: well-matched, high-mismatch, and mid-mismatch. The experiment

names refer to the relationship between the testing and training data. Both the testing and training data in the well-matched experiment use mixed close-talk and far-field microphone data from all noise classes. The high-mismatch experiment uses only the close-talk data from all noise classes for training, and the high and low far-field data for testing. The mid-mismatch experiment trains acoustic models from the far-field quiet and low noise classes, and uses the far-field high data for testing.

To evaluate SPLICE on this task, we train six complete sets of SPLICE parameters for each language, using unsupervised clustering as described in Section 2.6. These parameters are used to process all of the training and test data for each experiment.

Table 2. Aurora 3 word error rate results. NMN SPLICE with "iterative stochastic approximation" noise estimate.

Aurora 3 Referance Word Error Rate					
	Finnish	**Spanish**	**German**	**Danish**	**Average**
Well (x40%)	7.26%	7.06%	8.80%	12.72%	**8.96%**
Mid (x35%)	19.49%	16.69%	18.96%	32.68%	**21.96%**
High (x25%)	59.47%	48.45%	26.83%	60.63%	**48.85%**
Overall	**24.59%**	**20.78%**	**16.86%**	**31.68%**	**23.48%**

Aurora 3 Word Error Rate					
	Finnish	**Spanish**	**German**	**Danish**	**Average**
Well (x40%)	5.57%	4.20%	5.39%	6.80%	**5.49%**
Mid (x35%)	12.59%	8.56%	12.88%	20.18%	**13.55%**
High (x25%)	8.59%	10.74%	9.48%	16.87%	**11.42%**
Overall	**8.78%**	**7.36%**	**9.03%**	**14.00%**	**9.79%**

Aurora 3 Relative Improvement					
	Finnish	**Spanish**	**German**	**Danish**	**Average**
Well (x40%)	23.28%	40.51%	38.75%	46.54%	**37.27%**
Mid (x35%)	35.40%	48.71%	32.07%	38.25%	**38.61%**
High (x25%)	85.56%	77.83%	64.67%	72.18%	**75.06%**
Overall	**43.09%**	**52.71%**	**42.89%**	**50.05%**	**47.19%**

Table 2 presents accuracy results for Aurora 3, using NMN SPLICE and noise estimates from the iterative stochastic approximation.

The improvement over the baseline in the high-mismatch experiments is large. This is to be expected for any good noise-removal technique, as the original accuracy was quite low, and we are trying to make the noisy cepstra similar to the clean cepstra used to train the acoustic model.

The improvement over the baseline in the mid-mismatch experiments is smaller, but still significant. For this case, the training data consists of cleaned far-field data from the low and quiet noise types. The test data consists of similar data from the high noise type. SPLICE tries to make all of these utterances similar to clean data, and the accuracy improves.

We also see a large relative increase in accuracy for the well-matched experiments. The acoustic models for this experiment are trained on close-talk and cleaned far-field microphone data from all noise types, and the test data consists of a similar mix. The reference word error rate, 8.96%, is already low, and we are able to bring it down to 5.49%.

The Danish language exhibits the worst absolute recognition accuracy for all of our Experiments. Since this is also true of the baseline, we believe that it is an inherent property of the data and not due to any deficiency of SPLICE.

4. 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 Gaussian components for each separate distortion condition. We can leverage this model to identify the type of corruption currently being encountered in each test utterance.

We show in this paper that SPLICE is equally effective for improving the word recognition accuracy of both artificially and realistically distorted speech.

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 [8, 9].

5. 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] D. Pearce, "Enabling new speech driven services for mobile devices: An overview of the ETSI standards activities for distributed speech recognition front-ends," in *Applied Voice Input/Output Society Conference (AVIOS) 2000*, San Jose, CA, May 2000.

[3] L. Deng, A. Acero, L. Jiang, J. Droppo, and X.D. Huang, "High-performance robust speech recognition using stereo training data," in *Proc. 2001 ICASSP*, Salt Lake City, Utah, May 2001, pp. 301–304.

[4] J. Droppo, A. Acero, and L. Deng, "Efficient on-line acoustic environment estimation for FCDCN in a continuous speech recognition system," in *Proc. 2001 ICASSP*, Salt Lake City, UT, May 2001, pp. 209–212.

[5] J. Droppo, L. Deng, and A. Acero, "Evaluation of the SPLICE algorithm on the Aurora 2 database," in *Proc. Eurospeech 2001*, Aalborg, Denmark, September 2001, pp. 217–220.

[6] L. Deng, J. Droppo, and A. Acero, "Recursive estimation of nonstationary noise using a nonlinear model with iterative stochastic approximation," in *Proc. ASRU Workshop*, Madonna di Campiglio, Trento, Italy, December 9–13 2001, 4 pages (CDROM).

[7] L. Deng, J. Droppo, and A. Acero, "Robust speech recognition using iterative stochastic approximation for recursive estimation of nonstationary noise," *IEEE Transactions on Speech and Audio Processing*, to appear.

[8] B. Frey, L. Deng, A. Acero, and T. Kristjansson, "ALGONQUIN: Iterating Laplace's method to remove multiple types of acoustic distortion for robust speech recognition," in *Proc. 2001 Eurospeech*, Aalborg, Denmark, September 2001, pp. 901–904.

[9] J. Droppo, A. Acero, and L. Deng, "A nonlinear observation model for removing noise from corrupted speech log mel-spectral energies," in *Proc. 2002 ICSLP*, Denver, CO, September 2002, (Submitted).