Maximum Mutual Information SPLICE Transform for Seen and Unseen Conditions

Jasha Droppo and Alex Acero

Speech Technology Group
Microsoft Research, Redmond, WA, USA
{jdroppo,alexac}@microsoft.com

Abstract

SPLICE is a front-end technique for automatic speech recognition systems. It is a non-linear feature space transformation meant to increase recognition accuracy. Our previous work has shown how to train SPLICE to perform speech feature enhancement. This paper evaluates a maximum mutual information (MMI) based discriminative training method for SPLICE. Discriminative techniques tend to excel when the training and testing data are similar, and to degrade performance significantly otherwise. This paper explores both cases in detail using the Aurora 2 corpus. The overall recognition accuracy of the MMI-SPLICE system is slightly better than the Advanced Front End standard from ETSI, and much better than previous SPLICE training algorithms. Most notably, it achieves this without explicitly resorting to the standard techniques of environment modeling, noise modeling or spectral subtraction.

1. Introduction

Automatic speech recognition systems without explicit provisions for noise robustness degrade quickly in the presence of additive noise. As a consequence, how to best add noise robustness to such systems is an area of active research.

SPLICE (stereo piecewise linear compensation for environment) [1] is a powerful method for normalizing and enhancing features. Given enough parameters, it can approximate any feature transformation with a high degree of precision.

In its original ML-SPLICE formulation, two synchronous feature streams were needed to train the SPLICE parameters. By using noisy and clean feature streams, SPLICE learned an effective speech enhancement function.

There has been recent interest in producing discriminatively trained linear transformations of the feature space, such as MCMIP[2], MCELR[3] and MPE-HLDA[4]. By tying the transform parameters to recognition accuracy, these methods achieve modest improvements over LDA and HDA[5]. But, being linear transformations, they are quite limited.

A natural extension is to discriminatively train a non-linear transformation of the feature space. This is the approach taken by MCE-SPLICE[6], fMPE[7], and the method presented in this paper, MMI-SPLICE. For a comparison of SPLICE and fMPE, see [8].

MMI-SPLICE is much like SPLICE, but without the need for target clean features. Instead of learning a speech enhancement function, MMI-SPLICE learns to increase recognition accuracy directly with a maximum mutual information objective function. It retains the concept of source and target feature spaces from ML-SPLICE, but now the target is allowed to take whatever values increase the MMI objective function.

The overhead for the SPLICE runtime is small. Most of the effort is in calculating a few hundred Gaussian posteriors. Because MMI-SPLICE and ML-SPLICE only differ in the way they are trained, MMI-SPLICE shares these light runtime requirements.

This paper presents a suitable training algorithm for MMI-SPLICE, with results on the Aurora 2 corpus[9]. This corpus is interesting because it lets us see how the algorithm behaves with data that is similar to the training data (Set A), as well as test sets that differ in noise type (Set B) and channel (Set C).

A significant consideration in designing any discriminative training algorithm is over-training. Two different over-training conditions are examined. First, how well does MMI-SPLICE generalize when the conditions in the training set match those in the test set? Experimental results on test set A demonstrate that, for this case, word error rate can be improved substantially. The second case is when the conditions in the training set do not closely match those in the test set. Experimental results on test set B show that, even for mismatched noise conditions, small gains are possible.

This paper is organized as follows. Section 2 covers the SPLICE transformation and demonstrates hot to use a gradient-based technique to do MMI training of the offset parameters. Section 3 describes the experimental setup, as well as how the parameter update equations are approximated. Section 4 discusses the behavior of the complete system to both seen and unseen noise conditions.

2. The SPLICE Transform

The SPLICE transform was introduced as a method for overcoming noisy speech [1]. SPLICE takes noisy acoustic observations y, and produces clean estimates $\hat{x}$.

The relationship between x and y is modeled as a constrained Gaussian mixture model (GMM). An auxiliary variable m is introduced to index the hidden state of the GMM.

$$\begin{aligned} p(x,y,m) &= p(x|y,m)p(y|m)p(m) \\ p(y|m) &= N(y;\mu_m,\sigma_m) \\ p(x|y,m) &= N(x;A_m y + b_m,\gamma_m) \end{aligned}$$

The SPLICE transform $f_\lambda(y)$ is the MMSE estimate of x, given y and the model parameters λ. In effect, the GMM induces a piecewise linear mapping from y to x.

$$\begin{aligned} \hat{x} = f_\lambda(y) &= E[x|y] \\ &= \sum_s E[x|y,m]p(m|y) \\ &= \sum_m (A_m y + b_m)p(m|y) \end{aligned} \quad (1)$$

In most SPLICE implementations, the conditional mapping of y to x given s is a simple offset ($A_s = I$), although including the rotation A_s can provide better recognition accuracy with fewer mixture components.

In [1], a maximum likelihood training procedure was used which assumes both x and y are observable during training. When y is a distorted, noisy version of clean speech x, this ML-SPLICE (maximum likelihood SPLICE) learns a transformation for enhancing noisy speech.

In this work, a discriminative training procedure is introduced. This eliminates the necessity of providing observable clean features x. Where ML-SPLICE learns a transformation to a fixed oracle feature space, MMI-SPLICE is only concerned with improving the accuracy of the end-to-end system. Although the training procedure is drastically different, the runtime code remains unchanged.

2.1. Gaussian Mixture Model Training

All of the results in this paper were obtained from Gaussian mixture models (GMM) with a tied covariance structure. Mean parameters were initialized by selecting K vectors uniformly spaced throughout the noisy training data. The variance parameters were initialized to unit covariance. Ten iterations of expectation-maximization training were performed to refine the model parameters.

2.2. SPLICE Offset Parameter Training

The SPLICE offset parameters were trained to maximize a MMI criteria over the noisy training data. The initial value chosen for the offset parameters was zero, corresponding to an identity transform of the acoustic data.

For R training observation sequences $\{\mathcal{Y}_1, \ldots, \mathcal{Y}_R\}$ with transcriptions $\{w_1, \ldots, w_R\}$, the MMI objective function is the composition of three functions: The global objective function, the per-utterance objective function, and the feature transformation.

$$\mathcal{F} = \sum_r \mathcal{F}_r \tag{2}$$

$$\mathcal{F}_r = \ln \frac{p(\mathcal{X}_r, w_r)}{\sum_w p(\mathcal{X}_r, w)} \tag{3}$$

$$\mathcal{X}_r = f_\lambda(\mathcal{Y}_r) \tag{4}$$

The global objective function $\mathcal{F}$ is a linear sum of the objective function for each utterance, $\mathcal{F}_r$.

The per-utterance objective function is the log of the conditional probability of the correct transcription, under the current acoustic model, given the acoustics. Ideally, the sum in the denominator of Eq. 3 is taken over all permissible transcriptions for the current utterance.

The features $\mathcal{X}_r$ are the SPLICE-transformed input to the speech recognition system for utterance r.

A direct optimization of Eqs. 2-4 with respect to the model parameters λ is intractable. Instead, we use a gradient-based linear method.

To use a gradient-based method, it is necessary to compute the partial derivative of $\mathcal{F}$ with respect to the free model parameters. Fortunately, the structure of the objective function allows a simple application of the chain rule.

Every $\mathcal{F}_r$ is a function of many acoustic model state conditional probabilities $p(x_t^r|s_t^r)$. Each of these is, in turn, a function of the SPLICE-transformed features x_{it}^r. And, each transformed feature is a function of the SPLICE parameters λ.

$$\frac{\partial \mathcal{F}}{\partial \lambda} = \sum_{r,t,s_t^r,i} \frac{\partial \mathcal{F}_r}{\partial \ln p(x_t^r|s_t^r)} \frac{\partial \ln p(x_t^r|s_t^r)}{\partial x_{it}^r} \frac{\partial x_{it}^r}{\partial \lambda} \tag{5}$$

The first term in Eq. 5 captures the sensitivity of the objective function to individual acoustic likelihoods in the model. It can be shown to be equal to the difference of the conditional and unconditional posterior, with respect to the correct transcription. These are simply the numerator and denominator terms that occur in standard MMI estimation[10].

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial \ln p(x_t^r|s_t^r)} &= p(s_t^r|\mathcal{X}_r, w_r) - p(s_t^r|\mathcal{X}_r) \\ &= \gamma_{s_t^r}^{\text{num}} - \gamma_{s_t^r}^{\text{den}} \end{aligned} \tag{6}$$

The second term in Eq. 5 captures the sensitivity of individual likelihoods in the acoustic model with respect to the SPLICE-transformed features. Computing this differential is a simple matter.

$$\frac{\partial \ln p(x_t^r|s_t^r)}{\partial x_t^r} = -\Sigma_{s_t^r}^{-1}(x_t^r - \mu_{s_t^r}) \tag{7}$$

The final term in Eq. 5 captures the relationship between the transformed features and the SPLICE parameters. In this paper, only the SPLICE offset parameters are adapted. The differential follows from the definition of the SPLICE transform, Eq. 1.

$$\begin{aligned} \frac{\partial x_{it}^r}{\partial b_{jm}} &= \frac{\partial}{\partial b_{jm}} \left(y_{it}^r + \sum_{\acute{m}} b_{i\acute{m}} p(\acute{m}|y_t^r) \right) \\ &= 1(i=j) p(m|y_t^r) \end{aligned} \tag{8}$$

Here, $1()$ is an indicator function that takes the value 1 when its argument is true, and zero otherwise.

Finally, the complete gradient can be expressed concisely.

$$\frac{\partial \mathcal{F}}{\partial b_m} = \sum_{r,t,s_t^r} p(m|y_t^r) \left(\gamma_{s_t^r}^{\text{num}} - \gamma_{s_t^r}^{\text{den}} \right) \Sigma_{s_t^r}^{-1}(\mu_{s_t^r} - x_t^r) \tag{9}$$

The gradient is then used to update the SPLICE parameters in such a way as to increase the objective function. Any gradient ascent method can be used, such as conjugate gradient or BFGS, but for this paper we chose to mimic the dynamic gradient scaling detailed in [7].

3. Experimental Setup

3.1. The AURORA 2 Task

The experiments presented here were based on the data, code, and training scripts provided within the Aurora 2 task[9]. The task consists of recognizing strings of English digits embedded in a range of artificial noise conditions.

The acoustic model (AM) used for recognition was trained with the standard "complex back-end" Aurora 2 scripts on the multi-condition training data. This data consists of 8440 utterances, and includes all of the noise types seen in test set A, at a subset of the SNR levels.

The AM contains eleven whole word models, plus `sil` and `sp`, and consists of a total of 3628 diagonal Gaussian mixture components, each with 39 dimensions.

All results presented in this paper include whole-utterance cepstral mean normalization and automatic gain normalization.

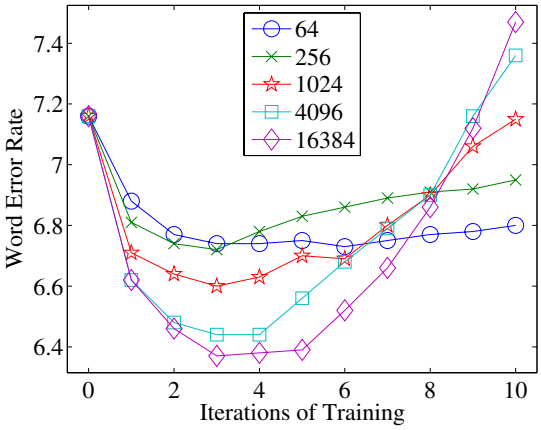

Figure 1: *Performance on test set A, which matches the multi-style training data. Increasing GMM size improves peak performance, as well as accentuating over-training behavior.*

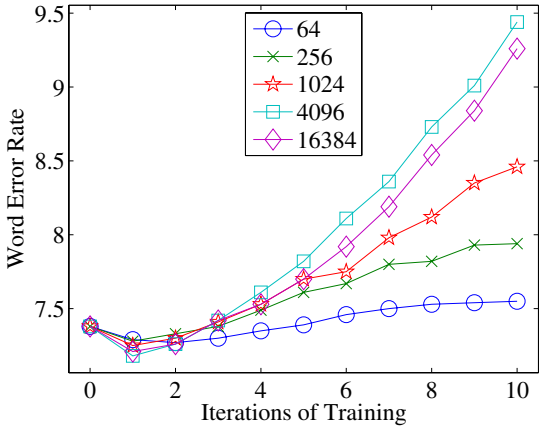

Figure 2: *Performance on test set B, which contains unseen noise types. The first MMI-SPLICE training iterations decrease WER, but subsequent iterations rapidly degrade the system.*

3.2. Approximating the Gradient with Lattices

Eq.6 requires computation of acoustic model state and mixture component posterior probabilities. Since exact computation can be somewhat resource intensive, the posteriors were approximated on word lattices generated by the baseline maximum likelihood multi-condition acoustic model. The time marks in the lattices were held fixed, and forward-backward was used within each arc to determine arc conditional posterior probabilities.

Denominator lattices were used to compute the posterior $p(s_t^r|\mathcal{X}_r)$. These were generated from unprocessed acoustic data, the full digit language model, and the multi-condition acoustic model. The `HVite` recognizer was used to produce a lattice equivalent to an N-best recognition with five tokens.

Numerator lattices were used to compute the posterior $p(s_t^r|\mathcal{X}_r, w_r)$. These were generated from unprocessed acoustic data, the correct transcription, and the multi-condition acoustic model. The `HVite` recognizer was used to generate a forced alignment of the correct transcription, which was then transformed into a single-path lattice.

4. Results

Recognition experiments were performed on test sets A, B, and C, with SPLICE models ranging in size from 64 components to 16384 components

4.1. Test Set A

The MMI criterion optimizes the end-to-end recognition system with respect to the data seen in the training set. The system should perform best on test data which is similar to the training set.

Figure 1 shows word error rate (WER) measures for test set A, which is fairly well matched to the multi-condition training data. The baseline WER is 7.16%, which is improved upon by all tested configurations under eight iterations.

The 64-component model is interesting, in that it doesn't appear to suffer from over-fitting the training data. It gets under 6.8% WER in just two iterations, and then flattens out. This shows that if you have similar training and testing data, a small SPLICE model can produces reasonable performance gains for unseen test data, while being quite robust to over-training.

As the model size is increased, three trends become apparent. First, peak performance constantly improves. Second, the number of iterations to achieve peak performance increases. Finally, the effect of over-training becomes quite dramatic.

The best performing configuration corresponds to the largest model size, 16384 mixture components. At this size, the SPLICE model has over twice as many parameters as the back-end acoustic model. A peak performance of 6.37% occurs after three iterations. This represents a relative error rate reduction of 11% over the baseline.

4.2. Test Set B

The additive noises present in test set B have similar long-term spectra to the noises in the multi-condition training data, but were derived from different sources. Examining system performance on this test set illustrates the system's behavior against unseen noise conditions.

One would expect a discriminative technique to fail when presented with test data that is dissimilar to the training set. This was not entirely the case for set B. Against expectations, the first iterations were actually slightly better than the baseline.

Figure 2 shows the WER measures for test set B. The baseline WER is 7.38%. All tested configurations improve the WER slightly in the first two iterations, and then deteriorate with further training.

The 64-component model is the most immune to over-training. Each of the first four iterations improve upon the baseline. Even after ten iterations, the degradation is still reasonable. The peak performance for the 64-component model occurs after two iterations, an accuracy of 7.27%, a reduction of 1.5% relative from the baseline.

As the model size is increased, over-training becomes more apparent. The MMI-SPLICE parameters are optimizing against noise types that do not occur in this test set. After only four iterations, all of the configurations have lost whatever gain they had achieved, and most are worse than the baseline.

The best performance is achieved by the largest models, after only one iteration. The model with 16384 components has a WER of 7.21%, which represents only a 2.3% relative reduction

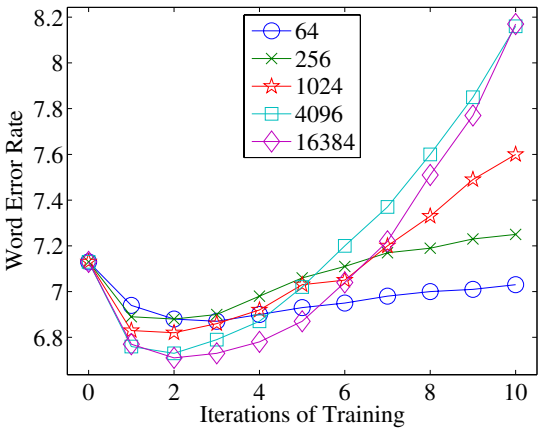

Figure 3: *Performance averaged over test sets A, B, and C. Larger models produce better peak accuracies.*

over the baseline.

If your training data and testing data contain dissimilar noise conditions, a small gain can still be achieved with MMI-SPLICE. The best solution is to use either a small number of components in the model, or to limit to one iteration to avoid over-training.

4.3. Average Performance

Figure 3 shows how the global average error rate metric for Aurora 2 is affected by model size and number of iterations. This average error rate is the standard weighted mean of the performance in sets A, B, and C. Set C contains a subset of noises found in sets A and B, with an additional linear filter applied, and is not presented separately in this paper.

For less than five iterations, the shape graph is dominated by the performance on Set A. In this region, Set A is experiencing great performance gains (ranging from 5% to 10% relative), and set B is either gaining or losing less than 2% relative.

As a result, the dominant trends are quite similar to the discussion of set A above. There is a broad valley in the error rate curve that a system designer can target.

The 16384-component model performs best. It has an average WER of 6.7% after two iterations, which is a 5.9% relative error rate reduction from the baseline. This also compares favorably with the advanced front end definition [11], which has a 6.8% WER.

The best ML-SPLICE result published on this task has a 7.83% WER[1]. The MMI-SPLICE training does much better, despite the fact that it uses only a single channel of training data, and has significantly fewer parameters.

An alternative method for training the SPLICE parameters, based on minimum classification error (MCE), was developed in [6]. That paper presented results against the Aurora 2 simple back-end configuration. With that configuration, the baseline error rate is 10.3%, the MCE-SPLICE error rate is 9.0%, and the MMI-SPLICE error rate is 8.4%.

5. Summary

We have presented a framework for training a discriminative front-end for noise robust speech recognition, and evaluated it on the Aurora 2 task. Our results indicate:

- If training and testing data are somewhat matched, large improvements are possible using MMI-SPLICE.
- For mismatched data, small improvements are possible, but over-training quickly becomes a problem.
- If you can anticipate some test conditions with your training data, the large gains in anticipated noises can swamp small losses in unanticipated noises.

6. Acknowledgments

The authors would like to thank Asela Gunawardana, who provided the reference lattice-based MMI code that was used as a basis for the MMI-SPLICE training, as well as a sympathetic ear.

7. References

[1] J. Droppo, L. Deng, and A. Acero, "Evaluation of SPLICE on the Aurora 2 and 3 tasks," in *Proc. ICSLP 2002*, Denver, USA, September 2002, pp. 29–32.

[2] M. Omar and M. Hasegawa-Johnson, "Maximum conditional mutual information projection for speech recognition," in *Proc. Eurospeech 2003*, Geneva, Switzerland, September 2003, pp. 505–509.

[3] X. He and W. Chou, "Minimum classification error linear regression for acoustic model adaptation of continuous density HMMs," in *Proc. 2003 ICASSP*, vol. 1, Hong Kong, April 2003, pp. 556–561.

[4] B. Zhang and S. Matsoukas, "Minimum phoneme error based heteroscedastic linear discriminant analysis for speech recognition," in *Proc. 2005 ICASSP*, vol. 1, Philadelphia, USA, March 2005, pp. 925–929.

[5] N. Kumar and A. Andreou, "On generalizations of linear discriminant analysis," Johns Hopkins University, Tech. Rep. JHU/ECE-9607, 1996.

[6] J. Wu and Q. Huo, "An evinronment compensated minimum classification error training approach and its evaluation on aurora2 database," in *Proc. ICSLP 2002*, Denver, USA, September 2002, pp. 453–457.

[7] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Solatu, and G. Zweig, "FMPE: Discriminatively trained features for speech recognition," in *Proc. 2005 ICASSP*, vol. 1, Philadelphia, USA, March 2005, pp. 961–964.

[8] L. Deng, J. Wu, J. Droppo, and A. Acero, "Analysis and comparison of two speech feature extraction/compensation algorithms," *IEEE Signal Processing Letters*, 2005, accepted for publication.

[9] 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*, Paris, France, September 2000.

[10] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge University, 2003.

[11] D. Macho, L. Mauuary, B. Noé, Y. M. Cheng, D. Ealey, D. Jouvet, H. Kelleher, D. Pearce, and F. Saadoun, "Evaluation of a noise-robust DSR front-end on Aurora databases," in *Proc. ICSLP 2002*, Denver, USA, September 2002, pp. 17–20.