Monte Carlo Model-Space Noise Adaptation for Speech Recognition

Daniel Povey and Brian Kingsbury

IBM T. J. Watson Research Center, Yorktown Heights, NY, USA
{dpovey,bedk}@us.ibm.com

Abstract

We describe a Monte Carlo method for model-space noise adaptation of Gaussian mixture models (GMMs). This method combines a single-Gaussian noise model with the GMM speech model to produce an adapted model. It is similar to Parallel Model Combination or model-space *Joint*, except that it applies to spliced and projected MFCC features rather than to MFCC plus dynamic features. We demonstrate the necessity of re-estimating the noise using both the silence and speech frames rather than just estimating it from silence frames, and obtain improvements on a *matched* test set without added noise using a system that includes all standard adaptation techniques.

Index Terms: speech recognition, noise adaptation

1. Introduction

In this paper we introduce a Monte Carlo method for model-space adaptation of Gaussian mixture speech models to noise conditions, given a Gaussian representing the noise. The method is applicable to speech features derived by splicing frames of MFCC features followed by a projection, and may be compared with techniques such as Parallel Model Combination [1] and *Joint* [4], which perform similar adaptation for features derived from MFCC features and their temporal derivatives. We also show how to re-estimate the noise Gaussian to maximize the combined model's likelihood, and we demonstrate that doing so rather than just estimating the noise from the silence frames is essential.

Our method is similar to Data-driven Parallel Model Combination (DPMC) [1], which similarly uses Monte Carlo techniques to combine speech and noise; the main difference is that DPMC re-estimates entire HMM states together rather than just Gaussians and it is harder to envisage efficient approximations to DPMC for this reason. Unlike [1], we also cover noise re-estimation and general feature transforms.

2. Noise modeling

We can classify noise modeling techniques into four different types, as illustrated in Figure 1:

i. Noise removal, which produces a point estimate of the clean speech that is independent of the speech state, e.g., SPLICE [6].

ii. Noise removal with uncertainty, which produces a distribution over clean speech, e.g., *Joint* uncertainty decoding [3][1] and SPLICE with uncertainty [8].

iii. Model adaptation, in which the speech model is combined with model of the noise to obtain a speech-plus-noise model, e.g., Vector Taylor Series [2], Parallel Model Combination [1] or model-based *Joint* [4].

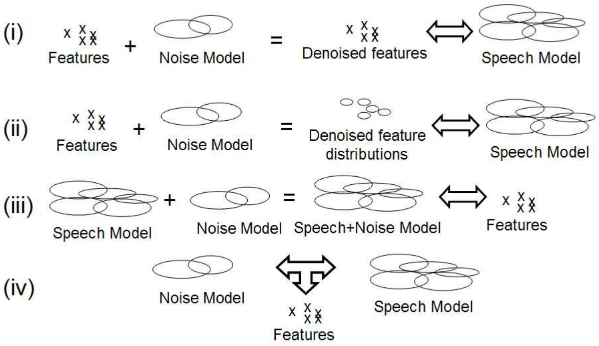

Figure 1: Four categories of noise modeling technique

iv. Joint modeling of speech and noise, in which the likelihood of an observation under a combined speech-plus-noise model is computed from a noise model and a speech model without ever computing the (non-Gaussian) combined model itself. This was used for separation of speech from speech in [10]; these techniques are currently practical only for log-FFT features, not MFCC. Algonquin [9] is one such algorithm.

The four types of techniques each have advantages and disadvantages. Noise removal (i) can never be exact because the estimate of the clean speech should depend on the speech state; however, it is the easiest technique to combine with other modeling techniques such as discriminative training. The other three types of noise modeling (ii, iii, and iv) can, in principle, provide an exact estimate: for a single-state noise model, the three types of method are in a sense equivalent, but they lead to different classes of approximations. The Monte Carlo adaptation method we describe here is a model adaptation (iii) approach.

3. Monte Carlo Model-Space Noise Adaptation

Our Monte Carlo method for noise adaptation produces a new speech-plus-noise model for each utterance by combining every Gaussian mixture component in a speech model with a single, diagonal Gaussian model of the noise. Because the noise model has very few parameters, it can be re-estimated for each utterance. We combine the Gaussians (one speech component and the noise model) by drawing pairs of samples, one from each Gaussian, lifting the samples into the spliced Mel-frequency power domain, adding up the signal power in each Mel bin, projecting back to the recognition feature space, and computing the mean and (diagonal) variance of the samples.

Our raw MFCC features are computed using 25-ms windows with a 10-ms shift and a bank of 40 partially overlapping triangular filters that are equally spaced on the Mel scale. If VTLN is used, the VTLN warping changes both the loca-

[1] With caveats: the distribution depends on the speech state

tions and widths of the filters. The log of the power in each Mel bin is computed, and a cosine transform that computes the 13 lowest-order coefficients (including c_0) produces the MFCC features. The recognition features are obtained by normalizing the MFCC features per speaker to have zero mean (and unit variance when VTLN is used), splicing across 9 frames to obtain 117-dimensional features, and then projecting to 40 dimensions using an LDA transform where the classes are the context-dependent states in our acoustic model. The 40 retained dimensions are diagonalized using a global semi-tied covariance (STC) and speaker-based constrained MLLR (CMLLR).

The cosine transform, variance normalization (if used), LDA projection, and rotation with STC and CMLLR (if used) may be summarized in a single, large matrix that transforms spliced log Mel powers to the recognition features. We approximate this transformation matrix using a square matrix as we will need to invert it. First, we keep all dimensions rejected by the LDA projection and model them using a single zero-mean, unit variance Gaussian. This model is appropriate because we use cepstral mean subtraction and the LDA transformation is normalized so it produces features with unit variance. The dimension loss caused by truncating the 40 MFCC coefficients to 13 is unavoidable, so we imagine that our features were computed from 13 bins instead of 40. Denoting the matrix that computes the 13-dimensional cosine transform as $\mathbf{C}$, the diagonal matrix of multiplicative factors in any cepstral variance normalization as $\mathbf{V}$, the LDA transformation (including rejected dimensions) as $\mathbf{T}$, the semi-tied covariance transformation matrix as $\mathbf{S}$, and the square part of any CMLLR transformation as $\mathbf{A}$, the total transformation (from the imagined 13 Mel bins) is:

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \\ & \mathbf{I}_{77} \end{pmatrix} \begin{pmatrix} \mathbf{S} & \\ & \mathbf{I}_{77} \end{pmatrix} \mathbf{T} \begin{pmatrix} \mathbf{VC} & & \\ & \mathbf{VC} & \\ & & \dots \end{pmatrix} \quad (1)$$

Constant offsets due to cepstral mean subtraction and CMLLR may be ignored as they would not affect our algorithm.

To combine a pair of 40-dimensional diagonal speech and noise Gaussians, we extend them to 117 dimensions using zero means and unit variances for the rejected dimensions, draw N (e.g., N=100) pairs of samples $(\mathbf{s}_i, \mathbf{n}_i)$ from the Gaussians, lift into the spliced log Mel power domain using the inverse total transformation (e.g., $\mathbf{s}'_i = \mathbf{s}_i\mathbf{M}^{-1}$), sum the powers dimension-wise to get the combined vector $\mathbf{a}'_{id} = \log(\exp(\mathbf{s}'_{id}) + \exp(\mathbf{n}'_{id}))$, and project back to get $\mathbf{a}_i = \mathbf{M}\mathbf{a}'_i$. The combined Gaussian is obtained by computing the sample mean and variance of the N vectors $\mathbf{a}_i$ in the 40 retained dimensions.

3.1. Sampling technique

We draw N samples (e.g., N=100) *a priori* from a 117-dimensional Gaussian distribution and normalize the set of N points to have zero mean and unit variance in each dimension (this eliminates certain first-order random effects due to the small sample size). We set the $N+1$'th point to be the same as the 1st point. Call this set of normalized random points $\mathbf{p}_i$. Then, to obtain the sets of N random vectors $\mathbf{s}_i$ and $\mathbf{n}_i$, we simply set $\mathbf{s}_i$ to $\mathbf{p}_i$ and $\mathbf{n}_i$ to $\mathbf{p}_{i+1}$, using appropriate scalings and shifts to obtain the correct means and variances in the retained dimensions.

3.2. Computation on demand

This technique is extremely expensive, so it is necessary to do this computation only for the Gaussians which we need to evaluate. Within a mixture of Gaussians, we first compute the likelihoods of all components, and then retain and noise-adapt only those components for which the log-likelihood is within a beam (e.g, 2.0) from the most likely component.

4. Noise model initialization and training

The diagonal-variance Gaussian noise model is computed per utterance. It is initialized from the sample mean and variance of all frames that are labeled as silence in an initial decoding pass. Re-estimation of the noise model relies on weak-sense auxiliary functions [11]. To maximize the likelihood of the data under the speech plus noise model, on each frame we accumulate the gradient of the likelihood with respect to the parameters of the noise Gaussian, and at the end of the utterance we compute "fake" mean and variance statistics with a count equal to the number of frames in the utterance, these statistics being computed so that the gradient of the standard auxiliary function is the same as the total gradient we computed.

4.0.1. Gradient of the likelihood w.r.t. noise model parameters

Before going into details, note that the derivative of a scalar-valued function with respect to its inputs may be computed in the same amount of time it takes to compute the function itself, provided that all intermediate values can be kept in memory. The following computation is an instance of this.

Let the data likelihood, which is our objective function, be f, and the gradient of f with respect to some vector $\mathbf{x}$ be $\frac{\partial f}{\partial}\mathbf{x}$. This is a vector having the same dimension as $\mathbf{x}$. Let the parameters of the Gaussians in the speech model be 117-dimensional means μ_j and diagonal variances $\mathbf{\Sigma}_j = \sigma^2_{jd}, 1 \leq d \leq 117$ for each Gaussian $j = 1 \dots J$ (the index runs over all states and mixture components), where default parameters $\mu_{jd} = 0$ and $\sigma^2_{jd} = 1$ are used for ($d > 40$). The noise model has parameters μ_n and $\mathbf{\Sigma}_n$.

The forward computation, in which we compute adapted parameters $\hat{\mu}_j$ and $\hat{\mathbf{\Sigma}}_j$ for a speech Gaussian, j, is

$$\begin{aligned}
s_{id} &= \mu_{jd} + p_{id}\sigma_{jd} && (2)\\
n_{id} &= \mu_{nd} + p_{i+1,d}\sigma_{nd} && (3)\\
\mathbf{s}'_i &= \mathbf{M}^{-1}\mathbf{s}_i && (4)\\
\mathbf{n}'_i &= \mathbf{M}^{-1}\mathbf{n}_i && (5)\\
a'_{id} &= \log(\exp(s'_{id}) + \exp(n'_{id})) && (6)\\
\mathbf{a}_i &= \mathbf{M}\mathbf{a}'_i && (7)\\
\hat{\mu}_j &= \tfrac{1}{N}\textstyle\sum_{i=1}^{N} \mathbf{a}_i && (8)\\
\hat{\sigma}^2_{jd} &= \left(\tfrac{1}{N}\textstyle\sum_{i=1}^{N} a^2_{id}\right) - \hat{\mu}^2_j && (9)
\end{aligned}$$

for $i = 1 \dots N$ and $d = 1 \dots 117$.

Let the occupation probability of Gaussian j at time t be $\gamma_j(t)$ and the observation vector at time t be $\mathbf{x}(t)$. Then, the auxiliary function for the likelihood (which we compute only for accepted dimensions) is

$$f = \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{d=1}^{40} -0.5\gamma_j(t)\left(\log\hat{\sigma}^2_{jd} + \frac{(\hat{\mu}_{jd} - x_d(t))^2}{\hat{\sigma}^2_{jd}}\right) \quad (10)$$

From this we compute (differentiating backwards):

$$\frac{\partial f}{\partial} \hat{\mu}_{jd} = \begin{cases} \sum_{t=1}^{T} \gamma_j(t) \frac{x_d(t) - \hat{\mu}_{jd}}{\hat{\sigma}^2_{jd}} & d \leq 40 \\ 0 & d > 40 \end{cases} \tag{11}$$

$$\tag{12}$$

$$\frac{\partial f}{\partial} \hat{\sigma}^2_{jd} = \begin{cases} 0.5 \sum_{t=1}^{T} \gamma_j(t) \left(\frac{(\hat{\mu}_{jd} - x_d(t))^2 - \hat{\sigma}^2_{jd}}{\hat{\sigma}^4_{jd}} \right) & d \leq 40 \\ 0 & d > 40 \end{cases} \tag{13}$$

$$\frac{\partial f}{\partial} a_{id} = \frac{1}{N} \left(\frac{\partial f}{\partial} \hat{\mu}_{jd} + 2 \left(a_{id} - \hat{\mu}_{jd} \right) \frac{\partial f}{\partial} \hat{\sigma}^2_{jd} \right) \tag{14}$$

$$\frac{\partial f}{\partial} \mathbf{a}'_i = \mathbf{M}^T \frac{\partial f}{\partial} \mathbf{a}_i \tag{15}$$

$$\frac{\partial f}{\partial} n'_{id} = \frac{\exp(n'_{id})}{\exp(s'_{id}) + \exp(n'_{id})} \frac{\partial f}{\partial} a'_{id} \tag{16}$$

$$\frac{\partial f}{\partial} \mathbf{n}_i = \mathbf{M}^{-1^T} \frac{\partial f}{\partial} \mathbf{n}'_i \tag{17}$$

$$\frac{\partial f}{\partial} \mu_n = \sum_{i=1}^{N} \frac{\partial f}{\partial} \mathbf{n}_i \tag{18}$$

$$\frac{\partial f}{\partial} \sigma_{nd} = \sum_{i=1}^{N} p_{id} \frac{\partial f}{\partial} n_{id} \tag{19}$$

$$\frac{\partial f}{\partial} \sigma^2_{nd} = \frac{\frac{\partial f}{\partial} \sigma_{nd}}{2 \sigma_{nd}} \tag{20}$$

If the objective function were a regular likelihood function given the noise Gaussian and observed noise data with zeroth, first and second order statistics γ_n, $\mathbf{x}_n$ and $\mathbf{S}_n$, we would have

$$\frac{\partial f}{\partial} \mu_{nd} = \frac{x_{nd} - \gamma_n \mu_{nd}}{\sigma^2_{nd}} \tag{21}$$

$$\frac{\partial f}{\partial} \sigma^2_{nd} = 0.5 \left(\frac{S_{nd} - 2 x_{nd} \mu_{nd} + \gamma_n \mu^2_{nd}}{\sigma^4_{nd}} - \frac{\gamma_n}{\sigma^2_{nd}} \right) \tag{22}$$

We use these equalities to create for the noise “fake” zeroth, first and second order noise statistics as follows:

$$\gamma_n = T \tag{23}$$

$$x_{nd} = \gamma_n \mu_{nd} + \sigma^2_{nd} \frac{\partial f}{\partial} \mu_{nd} \tag{24}$$

$$\begin{aligned} S_{nd} &= \left(2 \sigma^4_{nd} \frac{\partial f}{\partial} \sigma^2_{nd} \right) + \gamma_n \sigma^2_{nd} \\ &\quad + 2 x_{nd} \mu_{nd} - \gamma_n \mu^2_{nd} \end{aligned} \tag{25}$$

4.0.2. Updating the noise

The model update is the standard maximum-likelihood update given these fake noise statistics. The update is stable close to convergence, but is not always stable far from convergence. To remedy this problem, if the K-L divergence from the previous to the updated noise Gaussian is more than 1, or if any new variance element is negative, we increase γ_n by T, and recompute the fake statistics (Equations 24 and 25), iterating until the K-L divergence is less than 1. The update is iterative, with 8 iterations typically being sufficient. Note that this approach requires that the initial estimate of the noise have sufficient power. If it does not, the gradient becomes zero. Therefore, initializing the noise to that of the previous speaker is not a good idea.

4.1. Mean-only update

Much of the difficulty of model-space noise modeling techniques such as this one comes from the need to estimate the combined variance. Mean-only updates are much easier to make efficient as the computation can take place dimension by dimension in the spliced log Mel power domain. Therefore we were motivated to investigate how important the variance update is to our technique; if it is not important, then it becomes much easier to make more efficient versions of this scheme. The mean-only combination of the speech and noise Gaussians is obvious in this case: we just compute the mean of the combined points $\mathbf{a}_i$ and ignore their variance, keeping the variance from the original speech model. The training of the noise in the mean-only context simply involves setting the objective function gradient with respect to the combined variance to zero, i.e. Equation 13 becomes:

$$\frac{\partial f}{\partial} \hat{\sigma}^2_{jd} = 0 \tag{26}$$

4.2. Speaker Adaptive Training

Since this is a speaker adaptation technique (more correctly: utterance adaptation), it is natural to consider speaker adaptive training in the same way as is done for other techniques like VTLN, Constrained MLLR and MLLR. Model-space training is possible in this scheme in a way that is analogous to the noise update: the computation is symmetric with respect to the speech and noise so we can use the same approach, the difference being we need to accumulate statistics over all the data to update the speech model. The limiting of the K-L divergence to 1 as with noise has not been necessary in our system because none of the speech Gaussians move that far.

5. Experimental conditions

Our training data is 50 hours of English news broadcasts obtained by subsampling the 1996 and 1997 Hub4 training sets (LDC97S44 and LDC98S71 respectively). The features are as described above. We report results for a speaker-independent (SI) system with 1000 quinphone context-dependent states and 30000 mixture components and for a speaker adaptively trained i.e. CMLLR-SAT[2] system having 3000 quinphone context-dependent states and 50000 mixture components. The CMLLR-SAT system uses VTLN and CMLLR (Constrained MLLR) in training for speaker normalization. Our test set is the Dev04f test set from the DARPA EARS project, which comprises 3 hours of speech from 6 broadcasts collected between 15 November and 1 December 2003, and includes 22.6K words. The language model used for testing is a 3.3M 4-gram LM trained on a corpus of 335M words.

6. Experimental Results

Noise modeling is done per utterance after all other adaptations have been applied. We test three different systems: an unadapted SI system, an adapted SI system that uses CMLLR and MLLR in test only, and an adapted CMLLR-SAT system that uses VTLN, CMLLR and MLLR. MLLR adaptation operates only on the model means, and uses a regression tree with a minimum count of 3000 to estimate up to 16 transforms.

Table 1 shows the word error rate and per-frame likelihood on the test set with no noise adaptation, noise adaptation with the noise trained on silence and noise adaptation with the noise re-estimated for various numbers of iterations to maximize likelihood. The line “None/pruned” refers to not using any noise model, but only evaluating Gaussians within a margin of 2.0 log-likelihood from the best Gaussian within a state (to match the computation with noise). We can see that noise modeling

[2] Normally known as SAT but we are currently also publishing results with MLLR-SAT so we make this explicit

Noise model	Word Error Rate		
	SI	adapted SI	CMLLR-SAT
None	34.1%	30.8%	25.7%
None/pruned	34.1%	30.9%	25.9%
Silence	37.0%	32.4%	26.5%
1 iter	34.6%	30.4%	25.4%
2 iter	33.6%	30.1%	25.2%
4 iter	33.5%	30.2%	25.2%
8 iter	33.5%	30.3%	25.3%
16 iter	33.5%	30.3%	25.4%
	Likelihood/frame		
None	-54.24	-52.54	-52.98
None/pruned	-54.31	-52.60	-52.10
Silence	-56.71	-56.20	-56.20
1 iter	-54.48	-54.23	-54.61
2 iter	-54.47	-52.47	-53.60
4 iter	-54.04	-52.67	-52.56
8 iter	-53.94	-52.54	-52.17
16 iter	-53.93	-52.47	-52.07

Table 1: Noise adaptation and effect of noise re-estimation.

Noise model	Word Error Rate		
	SI	adapted SI	CMLLR-SAT
None/pruned	34.1%	30.9%	25.9%
Mean+Variance	33.5%	30.3%	25.3%
Mean only	33.6%	30.4%	25.3%
	Likelihood/frame		
None/pruned	-54.31	-52.60	-52.10
Mean+Variance	-53.94	-52.54	-52.17
Mean only	-53.79	-52.53	-52.05

Table 2: Noise adaptation (8 iterations of update) and effect of variance update.

gives a substantial degradation when the noise is estimated from the silence frames only. This is not surprising because the system was trained under matched conditions (there is no more noise in the test set than in the training set) so in effect we are modeling the noise twice. We get about 0.5% absolute improvement from noise modeling when we re-estimate the noise, which is surprising given that the test set is recorded under fairly clean conditions and does not have artificially added noise. The necessity of re-estimating the noise was also reported in [4].

Table 2 shows the effect of removing the variance part of the update and updating the model mean only. There is very little word error rate difference caused by not updating the variance. The likelihood seems to actually increase somewhat. The fact that variances make very little difference is good because almost all of the difficulty in these types of computations arises from the need to compute the updated variance; however, note that the effect on WER may be SNR dependent ([1], p. 95).

We also tried updating the model parameters using the same update technique used for the noise Gaussians; this involved applying our computation to all the training data (which is very slow). We did this on the unadapted system and tested with 8 iterations of noise update in both train and test; the WER improved from 33.5% to 33.4% and the likelihood changed very little, from -53.94 to -53.92. In another experiment, modeling the noise on a per-speaker rather than per-utterance level degraded WER from 30.3% to 30.6%.

Note that our system was built with MFCC features not PLP [12] because PLP features are harder to convert back into mel bin powers. A fully adapted CMLLR-SAT PLP system otherwise similar to our fully adapted CMLLR-SAT MFCC baseline gives a WER of 25.3% which is the same as our best noise adapted results; however, if we take account of a 0.2% loss from pruning away less likely Gaussians we can surmise that with a full computation we would have gained 0.2% versus PLP.

7. Conclusions and future work

We have demonstrated a proof-of-concept noise adaptation technique for use with spliced and projected MFCC features, and have shown improvements on fairly clean data (no noise added) with matched training. In our system the usefulness of these types of approaches will be limited by the difficulty of combining this approach with feature-space discriminative training (e.g., fMPE), which gives much more improvement. At this level of noise, even without discriminative training any gains are canceled out by the need to use MFCC rather than PLP features. Nevertheless, we hope that this approach may be useful in higher-noise environments and where feature-space discriminative training is not used. In particular, the observation that most of the improvement comes from mean adaptation opens the door to much more efficient implementations.

8. Acknowledgments

This work was funded by DARPA contract HR0011-06-2-0001. Thanks to John R Hershey, Hank Liao, and Steven J Rennie for comments and suggestions.

9. References

[1] Gales., M. J. F., “Model-Based Techniques for Noise Robust Speech Recognition”, PhD thesis, Cambridge University, 1996.

[2] Moreno, P.J, Raj, B. and Stern, R. M., “A Vector Taylor Series Approach for Environment Independent Speech Recognition”, Proc. ICASSP, 1996.

[3] Liao, H. and Gales, M. J. F., “Joint Uncertainty Decoding for Noise Robust Speech Recognition”, Proc. Interspeech, 2005.

[4] Liao, H., “Uncertainty Decoding for Noise Robust Speech Recognition,” PhD thesis, Cambridge University, 2008.

[5] Liao, H. and Gales, M. J. F., “Issues with Uncertainty Decoding for Noise Robust Speech Recognition,” Proc. Interspeech. Pittsburgh, USA, 2006.

[6] Deng, L., Acero, A., Plumpe, M and Juang, X. D., “Large Vocabulary Speech Recognition under Adverse Acoustic Environments,” Proc. ICSLP, 2000.

[7] Gales, M. J. F, “Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition,” Computer Speech and Language, vol. 12, Jan. 1998.

[8] Droppo, J., Acero, A. and Deng, L., “Uncertainty Decoding with SPLICE for Noise Robust Speech Recognition,” in Proc. ICASSP, 2002.

[9] Frey, B. J., Kristjansson, T., Deng, L. and Acero A., “ALGONQUIN: Iterating Laplace’s Method to Remove Multiple Types of Acoustic Distortion for Robust Speech Recognition”, Proc. Eurospeech, 2001.

[10] Kristjansson T., Hershey J., Olsen P., Rennie S., Gopinath R., “Super-Human Multi-Talker Speech Recognition: The IBM 2006 Speech Separation Challenge System”, ICSLP 2006.

[11] Povey, D., “Discriminative Training for Large Vocabulary Speech Recognition,” PhD Thesis, Cambridge University, 2003.

[12] Hermansky, H., “Perceptual linear predictive (PLP) analysis of speech”, Journal of the Acoustical Society of America, vol. 87, April 1990.