COMBINING FEATURE COMPENSATION AND WEIGHTED VITERBI DECODING FOR NOISE ROBUST SPEECH RECOGNITION WITH LIMITED ADAPTATION DATA

Xiaodong Cui and Abeer Alwan

Department of Electrical Engineering
University of California, Los Angeles, CA 90095
Email: xdcui@icsl.ucla.edu, alwan@icsl.ucla.edu

ABSTRACT

Acoutic models trained with clean speech signals suffer in the presence of background noise. In some situations, only a limited amount of noisy data of the new environment is available based on which the clean models could be adapted. A feature compensation approach employing polynomial regression of the signal-to-noise ratio (SNR) is proposed in this paper. While clean acoustic models remain unchanged, a bias which is a polynomial function of utterance SNR is estimated and removed from the noisy feature. Depending on the amount of noisy data available, the algorithm could be flexibly carried out at different levels of granularity. Based on the Euclidean distance, the similarity between the residual distribution and the clean models are estimated and used as the confidence factor in a back-end Weighted Viterbi Decoding (WVD) algorithm. With limited amounts of noisy data, the feature compensation algorithm outperforms Maximum Likelihood Linear Regression (MLLR) for the Aurora2 database. Weighted Viterbi decoding further improves recognition accuracy.

1. INTRODUCTION

Speech recognition systems trained in quiet suffer from performance degradation in the presence of ambient noise. This is mainly due to the mismatch between the clean acoustic models and noisy features. Generally, there are two ways to reduce the mismatch to achieve satisfactory performance. One apporach is to denoise front-end feature vectors while keeping the clean models unchanged [1][2] or develop noise robust features [3] [4]. The other approach involves adapting the back-end acoustic models according to the noisy environments [5] [6] [7].

In [8], a set of variable parameter HMMs whose Gaussian mean vectors are polynomial functions of the environments is used to deal with noisy speech. In this paper, polynomial regression of the utterance SNR is applied to compensate noisy features by removing the bias with respect to the clean features while the clean models remain unchanged. When dealing with limited environment adaptation data, one advantage of polynomial regression is that by learning the trend of the bias as a function of SNR, the algorithm can predict the bias at unseen SNRs. After feature compensation, the residual distribution of the compensation is estimated. The similarity between the residual distribution and clean Gaussian distribution is measured by the Euclidean distance between the mean vectors which is used as the confidence factor in a weighted Viterbi decoding algorithm.

The remainder of this paper is organized as follows. In Sections 2 and 3, formulations of feature compensation based on SNR polynomial regression and weighted Viterbi decoding are provided, respectively. Experimental results are shown in Section 4, and Section 5 concludes the paper with a summary.

2. FEATURE COMPENSATION

2.1. Bias removal by polynomial regression

Under the assumption that the power of a noisy speech signal in each frame is the sum of clean speech and noise, we have:

$$\mathbf{Y} = \mathbf{X} + \mathbf{N} \tag{1}$$

where $\mathbf{Y}$, $\mathbf{X}$ and $\mathbf{N}$ denote noisy speech, clean speech and noise, respectively. In the log-power domain, Eq.1 could be rewritten as:

$$\begin{aligned} \mathbf{Y^l} &= \mathbf{X^l} + \mathbf{log}(\mathbf{1} + \frac{\mathbf{N}}{\mathbf{X}}) \\ &= \mathbf{X^l} + \mathbf{g(snr)} \end{aligned} \tag{2}$$

where $\mathbf{Y^l}$ and $\mathbf{X^l}$ represent noisy and clean speech in the log-power domain and $\mathbf{snr}$ is the signal-to-noise ratio. A similar relationship is true in the cepstral domain:

$$\mathbf{Y^c} = \mathbf{X^c} + \mathbf{f(snr)} \tag{3}$$

where $\mathbf{Y^c}$ and $\mathbf{X^c}$ are noisy and clean speech in the cepstral domain, respectively.

From Eq.2 and 3, it is clear that the bias between the clean and noisy features is a nonlinear function of $\mathbf{snr}$. In this paper, this nonlinear function is modeled by a polynomial of order P, that is:

$$\mathbf{Y^c} = \mathbf{X^c} + \sum_{\mathbf{j=0}}^{\mathbf{P}} \mathbf{c_j (snr)^j} \tag{4}$$

Assuming that the acoustic models are Gaussian mixture HMMs, the probability density of observing feature o_t from state i is computed as:

$$p(o_t | s_t = i) = \sum_k \alpha_{i,k} b_{i,k}(o_t) \tag{5}$$

where $b_{i,k}(o_t) \sim N(o_t; \mu_{i,k}, \Sigma_{i,k})$ is the kth multivariate Gaussian mixture in state i with weight $\alpha_{i,k}$, $\mu_{i,k}$ and $\Sigma_{i,k}$ are the mean vector and covariance matrix associated with it, respectively.

The feature compensation algorithm removes the estimate of the bias from the noisy feature by computing the polynomial with

respect to SNR during the mixture Gaussian probability calculation based on clean HMMs and noisy data, which is shown in Eq.6:

$$p(o_t|s_t = i) = \sum_k \alpha_{i,k} N(o_t - \sum_{j=0}^{P} c_{ikj}\eta_t^j;\ \mu_{i,k}, \Sigma_{i,k}) \quad (6)$$

where η_t is the SNR for the frame at time t. c_{ikj}'s are the coefficients of the regression polynomial of state i, mixture k and polynomial order j. Depending on the adaptation data available, these coefficients could be tied at different levels - mixture, state, phonetic class or one for all phonemes. Note that c_{ikj} is a vector which has the same dimension as the feature vector. In other words, each element in the feature vector has its own regression polynomial.

2.2. Polynomial estimation

The regression polynomial of SNR is estimated based on the EM algorithm under maximum likelihood criterion [9]. Define the EM auxiliary function we are interested in as:

$$Q_b(\lambda;\overline{\lambda}) = \sum_{r=1}^{R}\sum_{i\in\Omega_s}\sum_{k\in\Omega_m}\sum_{t=1}^{T^r}\gamma_t^r(i,k)\cdot\log b_{ik}(o_t^r) \quad (7)$$

where R is the utterance number and T^r is the frame number of the rth utterance. $\Omega_s = \{1, 2, \cdots, N\}$ and $\Omega_m = \{1, 2, \cdots, M\}$ are the state and mixture sets, respectively. $\gamma_t^r(i,k) = p(s_t^r = i, \xi_t^r = k|O^r, \overline{\lambda})$ is the probability of staying at state i mixture k at time t given the rth observation sequence.

Without loss of generality, we assume each Gaussian mixture has one set of distinct regression polynomials. For other tying strategies, the derivations follow accordingly.

Optimizing $Q_b(\lambda;\overline{\lambda})$ with respect to c_{ikl}, we obtain:

$$\begin{aligned}\frac{\partial Q_b(\lambda;\overline{\lambda})}{\partial c_{ikl}} &= \frac{\partial}{\partial c_{ikl}}\sum_{r=1}^{R}\sum_{t=1}^{T^r}\gamma_t^r(i,k)\cdot \quad (8)\\ &\quad \log N(o_t^r - \sum_{j=0}^{P} c_{ikj}(\eta_t^r)^j; \mu_{ik}, \Sigma_{ik})\\ &= \sum_{r=1}^{R}\sum_{t=1}^{T^r}\gamma_t^r(i,k)\cdot\Sigma_{ik}^{-1}\cdot \quad (9)\\ &\quad (o_t^r - \sum_{j=0}^{P} c_{ikj}(\eta_t^r)^j - \mu_{ik})\cdot(\eta_t^r)^l = 0\\ &\quad l = 0, 1, \cdots, P\end{aligned}$$

By regrouping items, Eq.9 can be rewritten as:

$$\begin{aligned}&\sum_{j=0}^{P}\left[\sum_{r=1}^{R}\sum_{t=1}^{T^r}\gamma_t^r(i,k)\cdot\Sigma_{ik}^{-1}\cdot(\eta_t^r)^{j+l}\right]c_{ikj} \quad (10)\\ &= \sum_{r=1}^{R}\sum_{t=1}^{T^r}\gamma_t^r(i,k)\cdot\Sigma_{ik}^{-1}\cdot(o_t^r - \mu_{ik})\cdot(\eta_t^r)^l\\ &\quad l = 0, 1, \cdots, P\end{aligned}$$

In a similar way as [8], define:

$$\psi(\zeta,\rho,\alpha,\beta) = \sum_{r=1}^{R}\sum_{t=1}^{T^r}\gamma_t^r(i,k)\cdot\Sigma_{ik}^{-1}\cdot\zeta^\alpha\rho^\beta \quad (11)$$

Therefore, Eq.10 could be expressed as:

$$\begin{aligned}\sum_{j=0}^{P}\psi(\eta_t^r,\eta_t^r,l,j)\cdot c_{ikj} &= \psi(\eta_t^r, o_t^r - \mu_{ik}, l, 1)\\ &\quad l = 0, 1, \cdots, P \quad (12)\end{aligned}$$

c_{ikj} can be obtained by solving the $P+1$ equations in Eq.12. If the covariance matrix Σ_{ik} in ψ is diagonal (which is usually the case), the computational load could be significantly reduced [10].

2.3. Utterance SNR estimation

The SNRs employed in the feature compensation algorithm are signal-to-noise ratio of the whole utterance. They are computed by averaging over all speech frame SNRs in the utterance where the frame SNRs are estimated based on the minima statistics tracking algorithm [11].

3. WEIGHTED VITERBI DECODING

3.1. Estimation of residual distribution

Let $\hat{\mathbf{X}}^c$ denote the estimate of the clean speech feature $\mathbf{X^c}$ after the bias removal from the noisy speech feature $\mathbf{Y^c}$:

$$\hat{\mathbf{X}}^c = \mathbf{Y^c} - \mathbf{f(snr)} \quad (13)$$

Since $\hat{\mathbf{X}}^c$ is the approximation of the clean signal, the distribution of it can convey useful information of the bias removal. Consider the residual of bias removal $\hat{\mathbf{X}}^c$ as a random variable assuming Gaussian distribution:

$$\hat{\mathbf{X}}^c \sim N(X;\hat{\mu},\hat{\Sigma}) \quad (14)$$

the maximum likelihood estimation of the residual Gaussian distribution of state i and mixture k could be readily obtained by the EM algorithm as:

$$\hat{\mu}_{ik} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T^r}\gamma_t^r(i,k)\cdot\hat{o}_t^r}{\sum_{r=1}^{R}\sum_{t=1}^{T^r}\gamma_t^r(i,k)} \quad (15)$$

$$\hat{\Sigma}_{ik} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T^r}\gamma_t^r(i,k)(\hat{o}_t^r - \hat{\mu}_{ik})(\hat{o}_t^r - \hat{\mu}_{ik})^T}{\sum_{r=1}^{R}\sum_{t=1}^{T^r}\gamma_t^r(i,k)} \quad (16)$$

where $\hat{o}_t^r = o_t^r - \sum_{j=0}^{P} c_{ikj}\eta_t^j$.

3.2. Weighted Viterbi decoding

To measure the similarity of residual distributions after the feature compensation and the distributions in the clean model, averaged Euclidean distance is applied:

$$d(f,g) = \frac{1}{L}(\mu_f - \mu_g)^T(\mu_f - \mu_g) \quad (17)$$

where f and g are two probability distributions and L is the dimension of the vector.

The Euclidean distance between the residual distribution and the clean model distribution describes in an expectation sense the effectiveness of bias removal in the feature compensation algorithm. The smaller the distances, the more accurate the feature

compensation algorithm is. Therefore, in the Viterbi decoding network, the Euclidean distance is used as a confidence factor.

Weighted Viterbi Decoding (WVD) modifies the recursive step of the Viterbi algorithm by weighting the probability of observing features o_t given the HMM state j, $b_j(o_t)$, with the confidence factor of current state after feature compensation. The confidence factor γ_j can be inserted into the Viterbi algorithm by raising the probability $b_j(o_t)$ to the power γ_j to obtain the following state update equation [12]:

$$\phi_j(t) = \max_i\{\phi_i(t-1)\cdot a_{ij}\}[b_j(o_t)]^{\gamma_j} \tag{18}$$

where $\phi_j(t)$ represents the maximum likelihood of observing speech features o_1 to o_t and being in state j at time t, a_{ij} stands for the transition probability from state i state j and $\gamma_j \in [0,1]$ is a state dependent confidence factor that maps the Euclidean distances of the state j into the interval $[0,1]$.

Let d_i denote the Euclidean distance associated with state i and d_{ik} the Euclidean distance of mixture k within state i. d_i is defined as:

$$d_i = \sum_{k=1}^{M} \alpha_{ik} d_{ik} \tag{19}$$

which is the weighted summation of the Euclidean distances of the mixtures in state i.

The mapping function from the state Euclidean distance d_i into the state confidence factor γ_i is shown in Fig.1 where the $thresh$ and τ are experimentally determined.

$$\gamma_i = \begin{cases} 1 & \text{for} \quad d_i \le thresh \\ e^{-\tau\cdot(d_i - thresh)} & \text{for} \quad d_i > thresh. \end{cases} \tag{20}$$

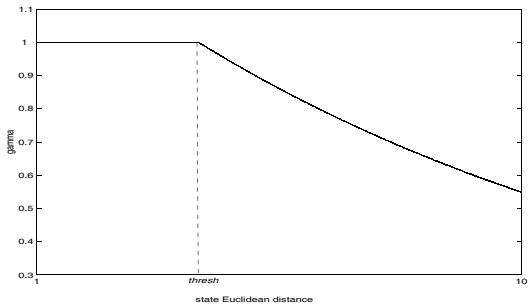

Fig. 1. Mapping curve from state Euclidean distance to confidence factor γ.

4. EXPERIMENTAL RESULTS

The algorithms of feature compensation and WVD proposed in this paper are trained and tested using connected digits in Sets A and B of the Aurora 2 database. There are eight types of background noise in this database, which are subway, babble, car and exhibition noise in Set A and restaurant, street, airport and station noise in Set B. Noisy speech data are generated by artificially adding the noise signals at a variety of SNR levels. Word-based HMMs are employed in the acoustic modeling with 16 emission states for each digit, 3 states for the silence model and 1 state for the short pause model. There are 3 mixtures in each state of digit models and 6 mixtures for silence and short pause models. Models are trained with clean speech data. Adaptation data sizes are chosen as 10, 50 and 200 utterances with SNR levels at clean, 20 dB, 15 dB, 10dB, 5 dB and 0 dB. The utterance SNRs are re-estimated based on the approach mentioned in Section 2.3.

The polynomials are different for different components of the feature vectors. For example, Fig. 2 shows the estimated SNR regression polynomials of the energy component(E) and $C1$ dimension of feature vectors for the four types of noise in Set A. The curves show that less compensation is needed when the SNR is high.

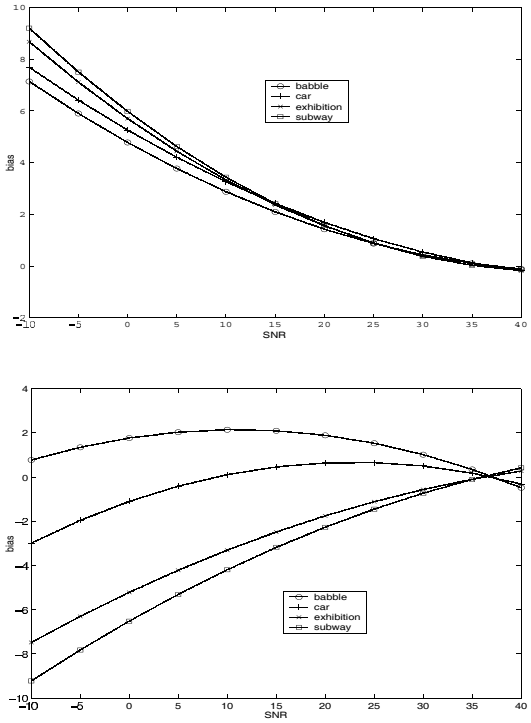

Fig. 2. SNR Regression polynomials of E(top) and $C1$(bottom) components of 4 types of background noise in Set A of the Aurora 2 database.

Comparative experiments are conducted between the popular adaptation algorithm MLLR and SNR polynomial regression-based feature compensation algorithm (referred to as FC) proposed in this paper under 10, 50 and 200 utterances adaptation conditions. In the 10 and 50 utterances cases, polynomials are shared for all word models. In the 200 utterances case, polynomials are shared within states. MLLR adaptation uses a regression class tree which is created by clustering Gaussian mixture means into 8 classes based on the Euclidean distance. Depending on the amount of adaptation data, the granularity of adaptation is dynamically chosen according to the statistics accumulated in the nodes. The transformation matrices assume three-block diagonal form with blocks accounting for the static, first and second order derivatives of the features. For the mapping function in WVD, $thresh$ is set to 0.6 and τ to 0.05. The performances are summarized in Tables

1 and 2.

Compared with the baseline, MLLR gives improvements in most cases. But for higher SNR conditions (e.g. clean and 20 dB), MLLR's performance is not satisfactory. FC obtains comparable accuracy to the baseline in the clean condition and significant improvements under all the other conditions. FC consistently overperforms MLLR under all SNR levels with all adaptation data sizes. The best FC performance over MLLR is achieved when only 10 utterances are used for environment adaptation since the regression polynomial can make a good prediction of unseen SNR levels. Note that as the number of adaptation utterances increases beyond 200, our recognition results approach those of the matched (multi-condition) case. In that case, the polynomials are mixture specific.

	baseline	10 Utt.		50 Utt.		200 Utt.	
		mllr	fc	mllr	fc	mllr	fc
Clean	99.0	94.2	98.8	94.6	99.0	97.1	99.0
20 dB	95.4	90.8	96.0	92.9	96.7	93.9	96.9
15 dB	87.3	84.1	92.1	88.1	93.0	88.4	93.4
10 dB	67.7	69.9	79.5	76.3	80.3	76.3	81.7
5 dB	39.5	48.9	58.0	58.7	61.2	57.8	65.8
0 dB	17.0	23.2	31.1	30.7	33.4	29.9	35.4

Table 1. Performance of baseline, MLLR and FC of Set A of the Aurora 2 database. WVD was not used.

	baseline	10 Utt.		50 Utt.		200 Utt.	
		mllr	fc	mllr	fc	mllr	fc
Clean	99.0	96.3	99.0	96.2	99.0	97.2	99.0
20 dB	92.8	94.0	96.2	95.0	97.4	95.2	97.4
15 dB	81.3	90.0	93.4	92.1	94.0	90.2	94.0
10 dB	59.0	79.9	80.5	83.7	81.6	79.6	86.0
5 dB	31.9	59.6	59.7	55.4	58.8	57.7	68.3
0 dB	13.7	29.8	31.4	32.2	31.5	33.0	38.5

Table 2. Performance of baseline, MLLR and FC of Set B of the Aurora 2 database. WVD was not used.

Table. 3 shows the recognition accuracy of the combination of FC and WVD algorithms on Sets A and B with different adaptation data sizes. Further performance improvement is observed.

	10 Utt.		50 Utt.		200 Utt.	
	set A	set B	set A	set B	set A	set B
Clean	98.9	99.0	99.0	99.0	99.0	99.0
20 dB	96.1	96.7	97.1	97.5	96.9	97.5
15 dB	93.0	94.0	94.1	94.3	94.6	94.6
10 dB	81.3	81.5	82.6	82.6	83.5	87.6
5 dB	62.2	62.4	65.1	64.0	69.2	70.0
0 dB	34.7	35.9	37.3	36.1	39.4	41.5

Table 3. Performance of combination of FC and WVD on Sets A and B of the Aurora 2 database.

5. SUMMARY AND CONCLUSIONS

In this paper, a polynomial regression-based feature compensation algorithm is proposed to reduce the mismatch between clean trained acoustic models and noisy speech features. The polynomials are a function of SNR and noise type. Weighted Viterbi decoding strategy is applied based on the Euclidean distance between the residual distribution of the feature compensation and the clean models. On average, the feature compensation algorithms obtains 35% word error reduction compared with the baseline and 15% over MLLR algorithm. The combination of feature compensation and weighted Viterbi Decoding algorithms can achieve further improvements of about 7%.

6. ACKNOWLEDGEMENTS

This work was supported in part by the NSF, and ST Microelectronics and the state of California through the UC Micro Program.

7. REFERENCES

[1] Steven Boll, "Suppression of acoutic noise in speech using spectral subtraction," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-27, no. 2, pp. 113–120, 1979.

[2] L. Deng, A. Acero, J. Droppo, and X. Huang, "High-performance robust speech recognition using stereo training data," *Proc. of ICASSP*, 2001.

[3] X. Cui, M. Iseli, Q. Zhu, and A. Alwan, "Evaluation of noise robust features on the aurora databases," *Proc. of ICSLP*, vol. 1, pp. 481–484, 2002.

[4] H. Hermansky and N. Morgan, "Rasta processing of speech," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 6, pp. 578–589, 1994.

[5] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density Hidden Markov Models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[6] M. Gales and S. Young, "HMM recognition in noise using parallel model combination," *Proc. of Eurospeech*, vol. 2, pp. 837–840, 1993.

[7] X. Cui, A. Bernard, and A. Alwan, "A noise-robust ASR back-end technique based on weighted Viterbi recognition," *Proc. of Eurospeech*, pp. 2169–2172, 2003.

[8] X. Cui and Y. Gong, "Variable parameter Gaussian mixture Hidden Markov Modeling for speech recognition," *Proc. of ICASSP*, vol. 1, pp. 12–15, 2003.

[9] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[10] Y. Gong, "Noise-dependent Gaussian mixture classifiers for robust rejection decision," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 2, pp. 57–64, 2002.

[11] R. Martin, "An efficient algorithm to estimate instantanous SNR of speech signals," *Proc. of Eurospeech*, pp. 1093–1096, 1993.

[12] A. Bernard and A. Alwan, "Low-bitrate distributed speech recognition for packet-based and wireless communication," *IEEE Trans. on Speech and Audio Processing*, vol. 10, pp. 570–580, 2002.