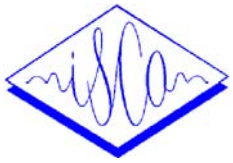

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

Sequential Bias Compensation for Robust Speech Recognition

Mohamed Afify
Department of Electrical Engineering, Cairo University, Fayoum Branch, Fayoum, Egypt

Abstract
Additive bias compensation is a simple and effective technique to overcome the performance degradation caused by acoustic mismatch in speech recognition systems. Bias is usually estimated in a batch mode, assuming that its parameters are constant for the whole utterance. This paper develops a new sequential algorithm for additive bias estimation, which can potentially track time varying mismatch effects within a test utterance. Relation to recursive Kullback-Leibler technique is pointed out, and the method is tested using computer simulations and speech recognition experiments. Significant performance improvements in the recognition rate are obtained for supervised adaptation.

10.21437/Eurospeech.1999-623

I. Introduction

Speech recognition systems suffer from performance degradation when they are operated in mismatched testing conditions. It has been demonstrated, in previous research efforts, that a simple and effective way to reduce this deterioration is additive bias compensation. In this method the mismatch is modeled as an additive bias and is estimated to maximize the likelihood of the adaptation or test data e.g. [1,2] . In estimating the bias parameters, it is usually assumed that the bias is constant for an utterance or a group of utterances. This article relaxes this assumption by considering sequential estimation of time varying additive bias. Compared to batch estimation techniques, the proposed method can potentially lead to improved convergence, track time varying mismatch, and reduce computation and memory requirements of the compensation process.

The study of sequential estimation algorithms for transformation parameters was addressed in [3,6] using a criterion based on the Kullback-Leibler (KL) information measure, originally introduced in [4]. Direct application of the KL criterion results in a computationally intensive algorithm, and some approximations are usually done. Here we give an alternative sequential algorithm based on minimizing the recursive prediction error[5]. It is also shown, that for the well known Viterbi approximation, the proposed method reduces to an approximate version of the KL recursion.

The bias compensation algorithm is formulated assuming that the underlying HMM is known. This is not true for unsupervised adaptation, where the identity of the adaptation or test data is not available. Thus, we also propose some extensions to the proposed method for the unsupervised case.

The paper is organized as follows. The problem is formulated, and sequential bias estimation is derived in Section II, Section III points out some extensions concerning the application of the proposed method to unsupervised adaptation, followed by experimental evaluation using computer simulation, and an isolated Arabic speech recognition task in Section IV. Finally conclusions are given in Section V.

II. Sequential Bias Compensation

This section presents the formulation and derivation of the sequential bias compensation algorithm. In the formulation, for simplicity, we consider scalar observations, but the extension to vector observations with independent components can be achieved by the application of the proposed method separately to each vector dimension. We also assume that each state has a single Gaussian distribution, and the extension to the Gaussian mixture case is straightforward.

Assume that speech is modeled by a hidden Markov model (λ), and the mismatch is represented by an additive bias. Following [5] we define the recursive prediction error (RPE) as

$$V_k(b) = \frac{1}{2}\sum_{t=1}^{k}\sum_{i=1}^{N}\frac{(y_t + b - \mu_i)^2}{\sigma_i^2}\gamma_{t|t}(i) \quad (1)$$

Where k is the current time instant, N is the number of HMM states, μ_i and σ_i^2 are the mean and variance of state i, and $\gamma_{t|t}(i)$ is the filtered state estimate equal to $p(s_t=i|y_1 \quad y_t,\lambda,b_1,..,b_{t-1})$. The goal of the bias estimation algorithm is to recursively minimize the RPE in (1) with respect to the bias parameter b. Sequential minimization can be achieved using a stochastic approximation algorithm as given in (2)

$$b_k = b_{k-1} - \frac{\partial l_k(b)/\partial b\big|_{b=b_{k-1}}}{\partial^2 V_k(b)/\partial b^2\big|_{b=\{b_1,\ldots,b_{k-1}\}}} \quad (2)$$

Where

$$l_k(b) = \tfrac{1}{2}\sum_{i=1}^{N}\frac{(y_k + b - \mu_i)^2}{\sigma_i^2}\gamma_{k|k}(i) \quad (3)$$

Following [5], an approximation of the second derivative in the denominator of (2) (denoted $1/f_k$) can be written as

$$\frac{1}{f_K} = \frac{\eta}{f_{K-1}} + \sum_{i=1}^{N}\frac{\gamma_{k|k}(i)}{\sigma_i^2} \quad (4)$$

Where η is a forgetting factor between 0 and 1, which helps tracking time varying mismatch, and the gradient in the numerator of (2) (denoted ϕ_k) evaluates to

$$\phi_k = \sum_{i=1}^{N}\gamma_{k|k}(i)\frac{(y_k + b - \mu_i)}{\sigma_i^2}\Big|_{b=b_{k-1}} + \tfrac{1}{2}\sum_{i=1}^{N}\frac{(y_k + b - \mu_i)^2}{\sigma_i^2}\frac{\partial\gamma_{k|k}(i)}{\partial b}\Big|_{b=b_{k-1}} \quad (5)$$

Thus, the sequential bias update equation in (2) can be rewritten as

$$b_k = b_{k-1} - f_k\phi_k \quad (6)$$

Where f_k and ϕ_k can be calculated from (4) and (5), and the derivative of the filtered state estimate in the second term of (5) can be calculated (refer to (8) and (9)) as

$$\frac{\partial\gamma_{k|k}(i)}{\partial b} = -\gamma_{k|k}(i)\frac{(y_k + b - \mu_i)}{\sigma_i^2} + \gamma_{k|k}(i)\sum_{j=1}^{N}\gamma_{k|k}(j)\frac{(y_k + b - \mu_j)}{\sigma_j^2} \quad (7)$$

To summarize, we start from an initial bias estimate and recursively apply the update equation (6) at each time instant, and using (4), (5), and (7) to calculate the required values, until we reach the end of the utterance. The filtered state estimates required in the equations are calculated using a variant of the forward algorithm of HMMs as shown below

$$\alpha_{k|k}(i) = \left[\sum_{j=1}^{N}\alpha_{k-1|k-1}(j)a_{ji}\right]\times N(y_k + b_{k-1};\mu_i,\sigma_i^2) \quad (8)$$

$$\gamma_{k|k}(i) = \frac{\alpha_{k|k}(i)}{\sum_{j=1}^{N}\alpha_{k|k}(j)} \quad (9)$$

Where a_{ji} s are the transition probabilities of the HMM, and N(.) is a Gaussian density. Note that in contrast to using the whole utterance for the smoothing of state probabilities in the conventional forward-backward algorithm, the calculations in (8) and (9) use only the current frame, and are referred to as filtered estimates, see [4] for details.

If we make the approximation that $\gamma_{k|k}(i) = 1$ for the dominant state, and is zero otherwise, it can be easily shown that the algorithm reduces to the recursions in (10) and (11), where s_k denotes the dominant state at time k. This will be referred to as the Viterbi approximation.

$$b_k = b_{k-1} - f_k\frac{(y_k + b_{k-1} - \mu_{s_k})}{\sigma_{s_k}^2} \quad (10)$$

$$\frac{1}{f_k} = \frac{\eta}{f_{k-1}} + \frac{1}{\sigma_{s_k}^2} \quad (11)$$

Interestingly, the recursions in (10) and (11) have a close relationship to the KL based algorithm [3,4]. In performing the KL recursions the smoothed (up to time k) state estimates are needed, and require intensive calculations. To reduce this computational burden, fixed lag smoothed estimates, and the filtered estimates (as above) are often used [3,4]. Using these approximations we find that the KL algorithm reduces to the update equations (10)-(11). The interested reader is referred to [5] for convergence analysis of the algorithm.

III. Extensions to Unsupervised Adaptation

We use the proposed algorithm for bias compensation of the test utterance. The estimated bias is added to the test features, and the resulting frames are used by the recognizer. For supervised adaptation (assuming the identity of the test word known for adaptation), the algorithm can be directly used. For unsupervised adaptation we tried the following variants:

1- Use the recognizer output, from an initial recognition session, as the identity of the test word. This will be referred to as RO.
2- Perform compensation separately, for each word model , and use the resulting bias to generate the new feature, i.e. use different biases for different models. This will be referred to as SEP.
3- Generate different biases from different models, and interpolate them, using weights from an initial recognition session, to calculate a final bias. This will be referred to as INT.

Results for supervised and unsupervised adaptation will be presented in the evaluation section. In all the results, based on some preliminary experiments, the forgetting factor was set to 0.9. For comparison purpose we use a bacth adaptation algorithm in the evaluation section. This algorithm is similar to conventional techniques, e.g. [1], but uses the filtered state estimates as defined in Section II.

IV. Experimental Evaluation

a)Simulation Experiments

We first performed some simulation experiments to compare the algorithm in (4)-(7), referred to as recursive prediction error (RPE) algorithm, to the simplified algorithm (10)-(11), referred to as recursive least square (RLS) algorithm. To this end, we used a 3-state left to right HMM with no skipping, the transition probabilities were set to 0.5 each, and the state variances were taken equal to one. The state means were respectively 0,5, and 10. We added different additive biases to the observations generated by the HMM, and used both the RPE, and RLS algorithms to estimate the bias value. In all cases we found that both algorithms converge to the true bias value. Figure 1 shows a sample result for b = 2.0, and a sequence of 20 observations. In these experiments, due to the stationary nature of the bias, we take the forgetting factor equal to 1. Due to the comparable performance of both algorithms we used the computationally cheaper RLS algorithm in our speech recognition experiments.

b)Speech Recognition Experiments

We test the proposed method on an isolated word Arabic speech recognition task. The vocabulary size is equal to 12. The sampling rate is 11.025KHz. Each vocabulary word is represented by a 5-state left to right HMM with no skipping, and each state has a 5 component Gaussian mixture. The feature vector consists of 12 LPC derived cepstral coefficients appended by the first order difference coefficients, resulting in a 24-dimension feature space. All models are trained using the segmental K-means algorithm [7]. The database consists of 42 speakers, each uttering the vocabulary 5 times. 28 speakers are used for training, and 14 speakers are used for testing. 3 different permutations of the training and testing speakers are used, resulting in about 2500 test utterances. The speaker independent recognition rate is 93%. We used the proposed method for adaptation as detailed in Section III. Results of different adaptation scenarios are presented and discussed below. All results are compiled in Table I.

1- Supervised adaptation yielded 96.8% recognition rate, which is about 50% error reduction compared to the uncompensated error rate. Also batch adaptation resulted in 95.2% recognition rate, which is significantly lower than the proposed algorithm.
2- Unsupervised adaptation using the recognizer output yielded 92.4%, while that using separate bias for each word model gave 89.5%, and that performing bias interpolation resulted in 91.1%. Both 3 techniques are worse than the uncompensated result, and this point requires further investigation.

V. Conclusion

We developed a sequential algorithm for additive bias compensation, in the framework of ve prediction error principle [5]. An approximation to the obtained recursion was found to have a close relationship to KL sequential bias estimation [3]. When applied in an isolated word speech recognition task, the proposed method, used in a supervised mode, resulted in significant improvement of the recognition rate, compared to the uncompensated case, and to popular batch compensation methods. Initial experiments using unsupervised adaptation gave degraded results, and requires further analysis.

Acknowledgement

The author is thankful to Eng. Mohamed Farouk Mansour for generously providing his HMM software, with which the experimental evaluation was carried out.

References

[1] M. Rahim, and B. H. Juang
removal by maximum likelihood estimation for robust telephone speech reco
Transactions on Speech and Audio Processing, Vol. 4, No. 1, pp.19-30, Jan. 1996.

[2] M. Afify, Y. Gong, and J.-P. Haton, A general joint additive and convolutive bias compensation approach applied to noisy Lombard speech recognition, IEEE Transactions on Speech and Audio Processing, Vol. 6, No. 6, pp. 524-538, Nov. 1998.
[3] L. Delphin-Poulat, C. Mokbel, and J. Idier, ynchronous stochastic matching based on the Kullback-Leibler
Proc. ICASSP-98, Vol. 1, pp. 89-92.
[4] V. Krishnamurthy, and J.B. Moore, On-line estimation of hidden Markov model based on the Kullback-Leibler information measure, IEEE Transactions on Signal Processing, vol. 41, pp. 2257-2273, Aug. 1993.
[5]
HMM state est
Signal Processing, vol. 46, No. 2, pp.475-486, Feb. 1998.
[6] N. S. Kim, Nonstationary environment compensation based on sequential estimation, IEEE Signal Processing Letters, vol. 5, no. 3, pp. 57-59, March 1998.
[7] L.R. Rabiner, B.H. Juang, Fundamentals of speech recognition, Prentice Hall Signal Processing Series, 1993.

Figure 1 Bias Value (vertical axis) Vs. frame index (horizontal axis) for the RLS and RPE algorithms. The actual bias value is 2.0.

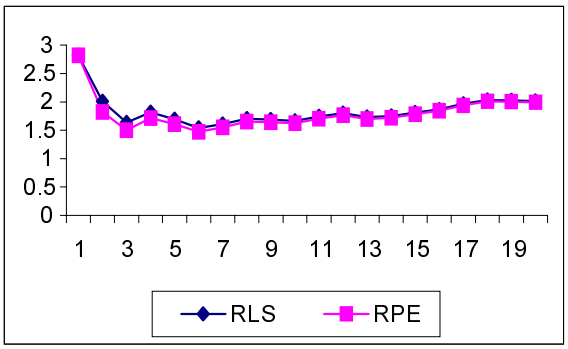

Table 1 Recognition percentage for the un-compensated case (UN) , supervised adaptation(SUP) , unsupervised adaptation (RO,SEP, and INT) as discussed in Section III, and batch adaptation (BA).

UN	SUP	RO	SEP	INT	BA
93	96.8	92.3	89.5	91.1	95.2