SIGNAL BIAS REMOVAL USING THE MULTI-PATH STOCHASTIC EQUALIZATION TECHNIQUE

Lionel Delphin-Poulat and Chafic Mokbel
FT.CNET/DIH/RCP
2 av. Pierre Marzin, 22307 Lannion cedex, France.
Tel. +33 2 96 05 13 47 FAX: +33 2 96 05 35 30 e-mail : delphinp@lannion.cnet.fr

ABSTRACT

We propose using Hidden Markov Models (HMMs) associated with the cepstrum coefficients as a speech signal model in order to perform equalization or noise removal. The MUlti-path Stochastic Equalization (MUSE) framework allows one to process data at the frame level: it is an on-line adaptation of the model. More precisely, we apply this technique to perform bias removal in the cepstral domain in order to increase the robustness of automatic speech recognizers. Recognition experiments on two databases recorded on both PSN and GSM networks show the efficiency of the proposed method.

1 INTRODUCTION

Numerous studies have been carried out in order to increase the robustness of automatic speech recognizers to disturbances (ambient noise, channel distortion, Lombard effect…). These studies include, at the preprocessing stage, the design of robust features and associated distances and the use of spectral, cepstral subtraction, etc…The increase of robustness can also be performed at the pattern matching stage, using methods such as parallel model combination [2]. In general, the speech signal is rough or not taken into account .

For years, Hidden Markov Models (HMMs) have been used to model various speech units; thus, they can be viewed as a complex representation of the speech signal [1] [6]. In [1], the HMM models both the speech signal and an additive stationary noise. The noise is removed by a Wiener filter that depends on the state in the HMM, but the signal is represented in the time domain, which is one of the main limits of this method. In [6], the equalization is performed in the feature space, but the algorithm is not frame-synchronous. An interesting approach for removing bias, which is frame-synchronous, is presented in [5]: the stochastic model in this case is very simple.

To overcome these limitations, the MUlti-path Stochastic Equalization (MUSE) technique is introduced in [4]. In this framework the speech signal is viewed in the cepstrum domain since the cepstrum coefficients are efficient to extract information from the signal in time or frequency domain. In the cepstrum domain the signal is modeled by an HMM to reflect the time and frequency variabilities of the signal. The main problem with HMMs is the existence of unobserved data: the states sequence. In [6], this problem is alleviated using the well-known Expectation-Maximization algorithm, in which frames can not be processed separately. To circumvent this difficulty, MUSE associates an equalization function to each possible path. The equalization function parameters are estimated, given the path, using a *Maximum Likelihood* or a *Maximum a Posteriori* criterion. But at time t, there are, for a fully connected model, N^t possible paths (N is the number of states in the HMM). Therefore, it is necessary to prune or to merge paths.

In the following section, we review the theoretical framework of MUSE: how to find the parameters of the equalization or filtering function and how to prune the paths. Then section 3 presents a detailed application of the method to bias removal; two formulas to compute the joint likelihood of each path are reported. Experimental results are given in section 4. Finally the conclusion stresses the main advantages of the MUSE technique to perform bias removal. And from the results of the experiments, further works and issues on that method are mentioned.

2 THEORETICAL FRAMEWORK

Let $Y = y_1, \dots, y_t, \dots, y_T$ be a sequence of noisy observations. The equalization function is defined by $x_t = f_\theta(y_t)$ where the sequence $X = x_1, \dots, x_t, \dots, x_T$ is distributed according to an HMM $\lambda = \{A, B, \Pi\}$ and θ is a vector containing the function's parameters. Let $S_t = s_{t_0}, \dots, s_t$ denote a partial state sequence and $Y_t = y_{t_0}, \dots, y_t$ the partial observation sequence from time t_0 up to time t.

$$\hat{\theta}(S_t) = \arg\max_{\theta} p(y_{t_0}, \dots, y_t | \theta, S_t, \lambda) \tag{1}$$

The state dependent distributions are assumed to be Gaussian with mean μ_i and covariance matrix Σ_i in state i (extension of the proposed method to HMMs with Gaussian mixtures is straightforward). Then $\hat{\theta}(S_t)$ is the solution of the following equation:

$$\sum_{\tau=t_0}^{t} \{ \left(\frac{\partial f_\theta}{\partial \theta}(y_\tau) \right)^T \Sigma_{s_\tau}^{-1} (f_\theta(y_\tau) - \mu_{s_\tau}) - \frac{\partial}{\partial \theta} \log(|\det(J_\theta(y_\tau))|) \} = 0 \tag{2}$$

where $J_\theta(y_t) = \frac{\partial f_\theta}{\partial y_t}(y_t)$ is the Jacobian matrix. Eq. 2 gives in fact a set of p scalar equations where p is the dimension of the vector θ. Thanks to the index t_0, the formulas show that the value of the parameters can be tracked. Once the parameters are estimated, we need to find:

$$\begin{aligned}\hat{S}_t &= \arg\max_{S_t} p(Y, S_t|\hat{\theta}(S_t), \lambda) \\ &= \arg\max_{S_t} P(S_t|\lambda) \prod_{\tau=t_0}^{T} p(y_t|\hat{\theta}(S_t), s_\tau, \lambda) \quad (3)\end{aligned}$$

To reduce the complexity, Eq. 3 is replaced by :

$$\begin{aligned}\hat{S}_t &= \arg\max_{S_t} p(Y, S_t|\hat{\theta}(S_{t_0}), \ldots, \hat{\theta}(S_t), \lambda) \\ &= \arg\max_{S_t} P(S_t|\lambda) \prod_{\tau=t_0}^{T} p(y_\tau|\hat{\theta}(S_\tau), s_\tau, \lambda) \quad (4)\end{aligned}$$

The two methods are equivalent if after a certain time, the parameter converges; the optimal path is the same for the two estimations.

If we use this method in a filtering scheme, the frame can be equalized according the most likely path at time t:

$$\hat{x}_t = f_{\hat{\theta}(\hat{S}_t)}(y_t) \quad (5)$$

In both cases we have to prune the paths. As in the Viterbi algorithm, we chose to keep, for each state, the equalization function corresponding to the most likely path leading to this state. The number of equalization functions is thus reduced to N. On the contrary to the Viterbi algorithm, this algorithm is no longer optimal since the pruning and the estimation of the parameters are nested; however, iteratively estimating the most likely paths and the parameters is one of the main interesting points of this method compared to stochastic matching. We could also have chosen to keep the M most likely paths, enabling to keep more than one path leading to a state.

3 APPLICATION TO BIAS REMOVAL

We now examine the case of a simple equalization function that performs a bias removal on the cepstrum coefficients. This function is interesting from a theoretical point of view, since the formulas can be expressed without making too many assumptions. In this case given the path, the *Maximum-Likelihood* criterion is equivalent to the *Minimum Mean Square Error* criterion. Removing a bias corresponds to an adaptation of the model means, the adaptation is global but estimated separately along each path. It is also useful from a practical point of view, since it has been shown that a bias removal on the cepstrum coefficients eliminates the channel effect (see *e.g.* [3]), if the channel effect can be represented by a linear time invariant filter. Moreover, in [7], cepstral subtraction is interpreted as Wiener filtering. All these reasons justify the cepstral subtraction.

More precisely, θ now represents bias b; the mismatch function is thus $f(y_t) = x_t - b$, we have then $\frac{\partial f_b^T}{\partial b}(y_t) = I_p$ where I_p is the identity matrix of order p and $\det(J_\theta(y_\tau)) = 1$. Therefore, the resolution of Eq. 2 leads to:

$$\hat{b}(S_t) = (\sum_{\tau=t_0}^{t} \Sigma_{s_\tau}^{-1})^{-1}(\sum_{\tau=t_0}^{t} \Sigma_{s_\tau}^{-1}(y_\tau - \mu_{s_\tau})) \quad (6)$$

One should notice that the same result would have been obtained by estimating b according to a *MMSE* criterion:

$$\hat{b}(S_t) = \arg\min_b E[(X_t - \hat{X}_t)^T(X_t - \hat{X}_t)|S_t, Y_t] \quad (7)$$

where $\hat{X}_t = \hat{x}_{t_0}, \ldots, \hat{x}_t$ with $\hat{x}_t = y_t - b$.

If we define $\alpha_t = -\frac{nt\log(2\pi)}{2} + \log(P(S_t))$ where n is the the feature vector size, the log-likelihood:

$$\begin{gathered}\log(p(Y_t, S_t|b, \lambda)) = \\ \alpha_t - \tfrac{1}{2}\textstyle\sum_{\tau=t_0}^{t}\{\log(\det(\Sigma_{s_\tau})) \\ +(y_\tau - b - \mu_{s_\tau})^T \Sigma_{s_\tau}^{-1}(y_\tau - b - \mu_{s_\tau})\} \quad (8)\end{gathered}$$

Let $X_1(S_t) = \sum_{\tau=t_0}^{t}(y_\tau - \mu_{s_\tau})^T \Sigma_{s_\tau}^{-1}$ and $X_2(S_t) = \sum_{\tau=t_0}^{t} \Sigma_{s_\tau}^{-1}$. Since those quantities can be computed recursively, we can find the optimal path at time according to Eq. 3. Formula 8 becomes:

$$\begin{gathered}\log(p(Y_t, S_t|\hat{b}(S_t), \lambda)) = \\ \alpha_t - \tfrac{1}{2}\textstyle\sum_{\tau=t_0}^{t}\{\log(\det(\Sigma_{s_\tau})) \\ +(y_\tau - \mu_{s_\tau})^T \Sigma_{s_\tau}^{-1}(y_\tau - \mu_{s_\tau})\} \\ + \left[\textstyle\sum_{\tau=t_0}^{t}(y_\tau - \mu_{s_\tau})^T \Sigma_{s_\tau}^{-1}\right] \hat{b}(S_t) \\ -\tfrac{1}{2}\hat{b}(S_t)^T \left[\textstyle\sum_{\tau=t_0}^{t} \Sigma_{s_\tau}^{-1}\right] \hat{b}(S_t) \quad (9)\end{gathered}$$

But we can also find optimal paths according to the assumption made in Eq. 4; in this case, the log-likelihood can be computed in the following way:

$$\begin{gathered}\log(p(Y_t, S_t|\hat{b}(S_{t_0}), \ldots, \hat{b}(S_t), \lambda)) = \\ \alpha_t - \tfrac{1}{2}\textstyle\sum_{\tau=t_0}^{t}\{\log(\det(\Sigma_{s_\tau})) \\ +(y_\tau - \mu_{s_\tau})^T \Sigma_{s_\tau}^{-1}(y_\tau - \mu_{s_\tau})\} \\ +\textstyle\sum_{\tau=t_0}^{t}(y_\tau - \mu_{s_\tau})^T \Sigma_{s_\tau}^{-1}\hat{b}(S_\tau) \\ -\tfrac{1}{2}\textstyle\sum_{\tau=t_0}^{t} \hat{b}(S_\tau)^T \Sigma_{s_\tau}^{-1}\hat{b}(S_\tau) \quad (10)\end{gathered}$$

If we use formula 8, the only assumption made is that, during the Viterbi decoding, the most likely path leading to any state i at time t (*i.e.* knowing $\hat{b}(S_t)$ with $s_t = i$) is the same as the path which would have been selected knowing $\hat{b}(S_{t'})$ with $S_{t'}$ containing state i at time t, for all $t' > t$. In other words, we assume that the selection done by the Viterbi Algorithm is not modified by the future value of the bias. Moreover, the approximated formula 10 may be more adequate in the case of bias varying with time. The tracking can also be done in a smooth fashion; since bias is expressed as a function of two sums, we can introduce a forgetting factor λ_{ff} in $X_1(S_t)$ and $X_2(S_t)$:

$$\begin{aligned}X_1(S_t) &= (y_\tau - \mu_{s_\tau})^T \Sigma_{s_\tau}^{-1} + \lambda_{ff} X_1(S_{t-1}) \quad (11) \\ X_2(S_t) &= \Sigma_{s_\tau}^{-1} + \lambda_{ff} X_2(S_{t-1}) \quad (12)\end{aligned}$$

where $0 \leq \lambda_{ff} \leq 1$. The value of λ_{ff} must be chosen to reflect the time constant of b. For example, if b models the channel effect, λ_{ff} will be related to the amount of time for which the channel can be considered as time invariant. In this case, we also assume that the channel evolves in a smooth fashion. Of course, this forgetting factor must also be compatible with time of convergence of the proposed algorithm.

4 EXPERIMENTS

Given the theoretical background, we can discuss experiments which were carried out on two databases recorded over both the PSN and GSM networks. Each speaker utters several words in one call. The first database consists of French digits and the second one of a 50 word vocabulary. Each utterance of a word will be referred as one recording. The features used to perform the recognition task are the first 8 Mel Frequency Cepstral Coefficients (MFCC), the energy and their first and second order derivatives. The bias subtraction is done only on the energy and the MFCC. The recognition system works in a speaker-independent mode. Each word is modeled by a 30-state HMM with Gaussian distributions.

We first show how the bias converges along one call: in Fig. 1, the bias of the most likely path for frame t is plotted versus the frame number in the call. To keep memory of the estimation of the bias from one utterance to another, for a given utterance, we initialized the variables $X_1(S_1^w)$ and $X_2(S_1^w)$ with $\lambda_{fu} X_1(\hat{S}^{w-1}_{t_{w-1}})$ and $\lambda_{fu} X_2(\hat{S}^{w-1}_{t_{w-1}})$ where S_1^w is one of the initial paths for utterance w, $\hat{S}^{w-1}_{t_{w-1}}$ is the optimal path for the previous utterance $(w-1)$ and λ_{fu} is a forgetting factor from one utterance to another ($0 \leq \lambda_{fu} \leq 1$). To obtain the results shown on Fig. 1, we set $\lambda_{fu} = 1$. We can see that the bias converges along one recording, we can consider that it is characteristic for one call. One utterance is represented by about 100 frames and is just enough to have the convergence, which is why setting $\lambda_{fu} = 0$ gives poor results.

We can see that the MUSE technique can be implemented in different ways depending on the choice of the formula to compute the logarithm of the joint likelihood and the choice of the forgetting factors λ_{ff} and λ_{fu}. Thus, we have experimented with different versions of the MUSE technique applied to bias removal on the digit database. We present here three versions (which are among the best):

- MUSE (1): the logarithm of the joint likelihood is computed according to formula 10 and $\lambda_{ff} = \lambda_{fu} = 1.0$.
- MUSE (2): the logarithm of the joint likelihood is computed according to formula 9, $\lambda_{ff} = 1.0$ and $\lambda_{fu} = 0.7$.
- MUSE (3): the logarithm of the joint likelihood is computed according to formula 10, $\lambda_{ff} = 0.99$ and $\lambda_{fu} = 1.0$.

These versions are compared to the baseline one in which f_θ is the identity function. We will detail the results obtained on GSM for three different conditions:

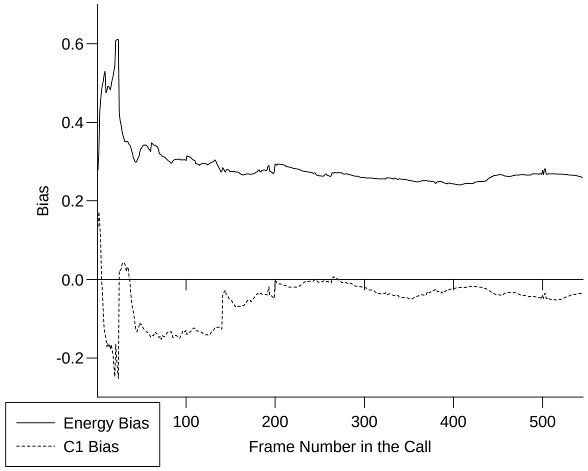

Figure 1: Bias Convergence

- GSM1: GSM indoors and stopped car
- GSM2: GSM running car
- GSM3: GSM outdoors

	PSN	GSM1	GSM2	GSM3
Baseline	0.91%	3.01%	3.67%	10.45%
MUSE (1)	0.81%	2.56%	2.4%3	6.73%
MUSE (2)	0.75%	2.50%	2.59%	6.80%
MUSE (3)	0.74%	2.53%	2.39%	6.69%

Table 1: Error rates on the digit vocabulary with a PSN-trained model

	PSN	GSM1	GSM2	GSM3
Baseline	1.78%	1.63%	1.35%	3.58%
MUSE (1)	1.52%	1.62%	1.39%	3.04%
MUSE (2)	1.58%	1.71%	1.39%	3.25%
MUSE (3)	1.52%	1.63%	1.47%	3.04%

Table 2: Error rates on the digit vocabulary with a GSM-trained model

Results on the digit database in tables 1, 2 and 3 allow us to draw initial conclusions. First the method gives poor results when training the model on the GSM-recorded data. Secondly, under matched conditions, there is no or little improvement since training and testing conditions are the same. In this case, when there is improvement, it can be explained by the fact that the MUSE technique adapts a general model trained with various data to a specific data. Thus, we observe that the method is efficient under mismatched conditions and when the speech signal model can be considered as clean.

	PSN	GSM1	GSM2	GSM3
Baseline	1.23%	2.01%	2.01%	4.26%
MUSE (1)	0.99%	1.94%	1.51%	3.52%
MUSE (2)	1.07%	1.94%	1.51%	3.25%
MUSE (3)	0.98%	1.94%	1.43%	3.25%

Table 3: Error rates on the digit vocabulary with a PSN-GSM-trained model

Other experiments showed that the two formulas for computing the logarithm of the joint likelihood give barely the same results. Introducing a forgetting factor seems to slightly improve the results. We then trained a model with PSN-data in which Cepstral Mean Normalization (CMN) was performed. It should be noted that CMN is not performed on testing data. Error rates for this model are reported in table 4. The Error rate on PSN data is thus reduced but on GSM data it remains barely constant: in this case, the signal in the MFCC domain is modeled more precisely and it is useful for PSN data but not for GSM data. For GSM data, the PSN-trained model was precise enough to perform bias removal. Here we can clearly see the tradeoff between the model and the mismatch function. Unfortunately, learning and testing under the same conditions still gives the best results. The results are limited by the form of the equalization function, which is too simple.

	PSN	GSM1	GSM2	GSM3
Baseline	0.91%	3.01%	3.67%	10.45%
MUSE (3)	0.78%	2.38%	2.69%	6.69%

Table 4: Error rates on the digit vocabulary with a model trained on PSN normalized data

These results were confirmed by carrying out tests on the 50-word database, with the version MUSE (3). On this base, tests with a GSM-trained model were not performed, since they did not give good results on the digit database and such a model can not be considered as representative of the speech process. On table 5, we can see that the reductions of the error rate are barely the same.

PSN-trained model				
	PSN	GSM1	GSM2	GSM3
Baseline	1.46%	4.10%	4.88%	9.19%
MUSE	1.33%	3.53%	3.54%	6.30%
PSN/GSM-trained model				
Baseline	1.64%	2.14%	2.07%	3.82%
MUSE	1.43%	1.92%	1.74%	3.79%

Table 5: Error rates on the 50-word database

5 CONCLUSION

In this paper, we stressed the need of using a speech signal model to perform equalization and filtering. We reviewed the MUSE framework which is one way to perform this at the frame level. A parametric function is associated to each path in an HMM. In the case of bias, we solved the equation that gives the parameters and we presented two formulas for computing the joint likelihood and two ways for tracking variation of the bias. The MUSE technique is tested on a digit database and on a 50-word database recorded on both PSN and GSM networks: the method can efficiently reduce the mismatch between testing and learning conditions (it is equivalent to an on-line adaptation of the model means). These encouraging results suggest the application of the proposed method to other kinds of equalization functions (*e.g.* to perform spectral subtraction). A linear mismatch function is currently being tested, this function allows one to modify both the mean and the variance of the model. Furthermore different strategies for pruning or merging paths should be examined.

References

[1] Y. Ephraim, “Statistical-Model-Based Speech Enhancement Systems”, Proc. of the IEEE, vol. 80, no. 10, Oct. 1992.

[2] M.J.F. Gales, S.J. Young, “Robust Continuous Speech Recognition Using Parallel Model Combination”, *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 5, Sept. 1996.

[3] C. Mokbel, J. Monné, D. Jouvet, *“On-line Adaptation of a Speech Recognizer to Variations in Telephone Line Conditions”*, Proc. EUROSPEECH’93, pp. 1023-1026, Berlin, 1993.

[4] C. Mokbel, *“MUSE : MUlti-Path Stochastic Equalization A theoretical framework to combine equalization and stochastic modeling”*, Proc. ESCA workshop on Robust Speech Recognition, pp. 211-214, Pont-à-Mousson, France, 1997.

[5] M.G. Rahim, B.-H. Juang, “Signal Bias Removal by Maximum Likelihood Estimation for Robust Telephone Speech Recognition”, *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 1, Jan. 1996.

[6] A. Sankar, C.-H. Lee, “A Maximum-Likelihood Approach to Stochastic Matching for Robust Speech Recognition”, *IEEE Trans. on Speech and Audio Processing*, vol. 3, no. 4, May 1996.

[7] S.V. Vaseghi, B.P. Miller, *“Noise-Adaptative Hidden Markov Models Based on Wiener Filters”* Proc. of EUROSPEECH’93, pp. 1023-1026, Berlin, 1993.