# SEQUENTIAL NOISE ESTIMATION WITH OPTIMAL FORGETTING FOR ROBUST SPEECH RECOGNITION

*Mohamed Afify*[*] *Olivier Siohan*

Multimedia Communications Research Lab
Bell Laboratories, Lucent Technologies
600 Mountain Avenue, Murray Hill, NJ 07974, USA
afify,siohan@research.bell-labs.com

## ABSTRACT

Mismatch is known to degrade the performance of speech recognition systems. In real life applications mismatch is usually non-stationary, and a general way to compensate for slowly time varying mismatch is by using sequential algorithms with forgetting. The choice of forgetting factor is usually performed empirically on some development data, and no optimality criterion is used. In this paper we introduce a framework for obtaining optimal forgetting factor. The proposed method is applied in conjunction with a sequential noise estimation algorithm, but can be extended to sequential bias or affine transformation estimation. Speech recognition experiments conducted first under a controlled scenario on the 5K Wall Street Journal task corrupted by different noise types, then under a real-life scenario on speech recorded in a noisy car environment validate the proposed method.

## 1. INTRODUCTION

In real world situations the input of speech recognition systems is often corrupted by different sources of mismatch. This significantly degrades their performance, and may limit the widespread use of speech recognition technology in practical applications. Many techniques were proposed to improve the performance of speech recognition in adverse environments. A good review of these methods can be found in [2]. Mismatch sources which are stationary are relatively easy to deal with, and non-stationarity often makes the problem more difficult. However, many real life scenarios include time varying mismatch sources, and thus dealing with non-stationarity is an important practical problem.

One approach to deal with non-stationarity is the use of sequential estimation algorithms. The ability to track slowly time varying environments is facilitated by using a forgetting factor (FF). This framework was used, for example, for the estimation of additive bias [9], frame synchronous stochastic matching [8], and additive noise estimation in the log spectral domain [4]. In all these situations the choice of the value of the FF is crucial for good performance. Usually the best value of the FF is empirically determined using a development test set, and no optimality criterion is involved.

A technique to obtain optimal step size for the recursive least square (RLS) algorithm was suggested in [6], and its convergence was analyzed in [7]. The basic idea is to recursively compute the step size to minimize the expected value of the error, and to develop an efficient recursion for the derivatives required in the minimization by differentiating the original recursion. In this paper we apply the same principle to optimizing the forgetting factor in sequential noise mean estimation as given in [4]. Extensions of the same framework to sequential bias estimation or frame synchronous stochastic matching will be considered in future work.

The approach of [4] is based on applying the sequential estimation algorithm of [10] in a vector Taylor series (VTS) [3] framework. VTS is a popular technique for noise compensation. A Taylor series expansion of a nonlinear mismatch function is used to linearize the problem. This facilitates estimating noise statistics, and clean speech features from the noisy speech.

The paper is organized as follows. In Section 2 we briefly review the VTS approach. This is followed by presenting the constant forgetting factor sequential algorithm of [4] in Section 3. Then we show how to derive optimal forgetting and how to embed it in the sequential noise estimation algorithm in Section 4. Section 5 gives experimental evaluation of the algorithm on two different tasks: the 5K Wall Street Journal (WSJ) task artificially corrupted by additive noise, a digit-string and command words recognition task in a real car environment. This is followed by conclusion in Section 6.

## 2. THE VTS APPROACH

Vector Taylor series approach [3] is used to compensate in the log spectral domain for the effect of additive noise in the linear spectral domain. Because noise is additive in the linear spectral domain the mismatch function will be nonlinear in the log spectral domain. Denote the noisy speech, clean speech, and noise in the log spectral domain by $z$, $x$, and $n$ respectively. The mismatch function in the log spectral domain can be written as

$$z \equiv f(n,x) = x + \log(1 + \exp(n - x)). \quad (1)$$

VTS is introduced to approximate this nonlinear function by a Taylor series expansion of order $i$. We call this $i^{th}$ order expansion $f^i(n,x)$. In an extension of VTS in [5], $f^i(n,x)$ is approximated by a minimum mean square error first order polynomial. We call this polynomial $g^i(n,x)$. This facilitates parameter estimation and feature compensation. The basic approach and its extension in [5] are summarized below. Details can be found in [3]-[5]. For the following the noise probability density function (pdf) $p(n)$ is assumed to be Gaussian, and the clean speech pdf $p(x)$ is a Gaussian mixture of size $M$.

1. Represent $f(n,x)$ by its $i^{th}$ order VTS, $f^i(n,x)$.
2. Approximate the VTS by a linear function $g^i(n,x) = A^i x + B^i n + C^i$ and obtain the MMSE estimates of $A^i$, $B^i$, and $C^i$. These 2 steps are required to get a linear relation between $z$, $x$, and $n$, so that it becomes possible to express the pdf of $z$ in terms of $x$ and $n$ pdfs.

[*]On leave from Faculty of Information and Computer, Cairo University, Cairo, Egypt.

3. Initialize the noise pdf $p(n)$ using the first $N$ frames of an utterance. Typically we use $N = 10$.
4. Derive $p(z)$ given $p(x)$, $p(n)$, and the linear approximation function $g^i(n, x)$.
5. For each test utterance refine the estimate of the noise mean $\mu_n$ (using sequential or batch estimation), and derive the corresponding $p(z)$.
6. Derive an MMSE estimate $\hat{x}$ given $z$, $p(z)$, and $p(x)$: $\hat{x} = E[x|z]$.
7. Map $\hat{x}$ to the cepstrum domain. This mapping is done as in the traditional parallel model combination (PMC) formulation [1]

In what follows we are interested in the noise mean estimation (step 5 above). Traditionally, maximum likelihood estimation in the framework of the EM algorithm is used. Kim [4] applied the sequential approach of [10, 11] to obtain sequential estimates of the noise mean. A constant forgetting factor is used to track slowly varying environment. We introduce sequential noise estimation with optimal forgetting based on the framework suggested in [6]. Both sequential algorithms will be presented in the following two sections.

## 3. SEQUENTIAL ESTIMATION WITH CONSTANT FORGETTING

In what follows we assume that in the log spectral domain clean speech $X$ is represented by a Gaussian mixture of size $M$ with means and variances $\{\mu_{xm}, \sigma^2_{xm} \mid 1 \leq m \leq M\}$, and the noise is Gaussian with mean $\mu_n$ and variance $\sigma^2_n$. As we assume components are independent we present algorithms in scalar form and repeat the same processing for every vector dimension. Also in the following we assume that the Taylor series order $i$ is known, and hence drop the dependence of the series coefficients on $i$. Instead, since the series coefficients depend on the mixture component index, we call these coefficients $A_m$, $B_m$, and $C_m$ to indicate dependence on $m$.

The sequential algorithm in [4] is obtained by maximizing with respect to $\mu$ the following Kullback-Leibler (KL) measure [10, 11]

$$Q_{t+1}(\mu_t, \mu) = E[\log p(Z_{t+1}, L_{t+1}|\mu)|Z_{t+1}, \mu_t], \quad (2)$$

where $Z_{t+1}$ is the sequence of observations and $L_{t+1}$ is the sequence of mixture indices. For the given Gaussian mixture model Equation (2) reduces to

$$Q_{t+1}(\mu_t, \mu) = -\frac{1}{2}\sum_{\tau=1}^{t+1}\sum_{m=1}^{M} p(m|z_{t+1})e^2_{\tau,m}(\mu)/\sigma^2_{zm}, \quad (3)$$

where

$$e_{\tau,m}(\mu) \equiv z_\tau - A_m\mu_{xm} - B_m\mu - C_m. \quad (4)$$

Applying the stochastic approximation algorithm [10] to the objective function in Equation (3) leads to the following sequential algorithm. The recursion is shown below in a slightly modified way to facilitate derivation of optimal forgetting in the next section. The convergence of this recursion to the true parameter value is guaranteed, under some suitable regularity conditions, as discussed in [12].

$$\mu_{t+1} = \mu_t + \epsilon s_{t+1}K^{-1}_{t+1} \quad (5)$$

$$K_{t+1} = (1-\epsilon)K_t + \epsilon r_{t+1} \quad (6)$$

$$s_{t+1} = \sum_{m=1}^{M} p(m|z_{t+1})e_{t+1,m}(\mu_t)B_m/\sigma^2_{zm} \quad (7)$$

$$r_{t+1} = \sum_{m=1}^{M} p(m|z_{t+1})B^2_m/\sigma^2_{zm} \quad (8)$$

where the noise mean at time frame $t$ is denoted $\mu_t$, $\mu_{xm}$ and $\sigma^2_{zm}$ are mean and variance of mixture component $m$ of the clean speech,and the noisy speech respectively. $z_t$ is the noisy observation at frame $t$, and the forgetting factor $\rho = 1 - \epsilon$, where $\epsilon$ is a non-negative constant with value less than 1.

## 4. SEQUENTIAL ESTIMATION WITH OPTIMAL FORGETTING

A reasonable way to obtain optimal forgetting is to try to maximize the measure in Equation (3) with respect to the forgetting factor or alternatively with respect to $\epsilon$. Benveniste et al. [6] suggested such an approach by recursively minimizing the expected error in a recursive least square (RLS) setting. They used the original recursion to get an expression for the required derivatives.

Here, recursive maximization of Equation (3) with respect to $\epsilon$ can be done using the following gradient algorithm

$$\epsilon_{t+1} = [\epsilon_t + \alpha s_{t+1}V_t]^{\epsilon^+}_{\epsilon^-}, \quad (9)$$

where $s_{t+1}$ is defined in Equation (7), $V_t \equiv \partial\mu_t/\partial\epsilon$, $\alpha$ is the learning rate, and $\epsilon^+$ and $\epsilon^-$ are upper and lower bounds to be discussed below.

Note that $\mu_t$ is an unconventional function of $\epsilon$, and also its derivative. However, by differentiating Equations (5) and (6) , with respect to $\epsilon$, we can obtain recursions to calculate the required derivatives. The required recursions are shown below together with recursions for $\mu$ and $K$ modified to include time dependent $\epsilon$. The algorithm consists of Equations (10)- (13) in addition to Equation (9) for updating the forgetting factor.

$$\mu_{t+1} = \mu_t + \epsilon_t s_{t+1}K^{-1}_{t+1} \quad (10)$$

$$K_{t+1} = (1-\epsilon_t)K_t + \epsilon_t r_{t+1} \quad (11)$$

$$W_{t+1} = (1-\epsilon_t)W_t + (r_{t+1} - K_t) \quad (12)$$

$$\begin{aligned} V_{t+1} &= V_t(1 - \epsilon_t K^{-1}_{t+1}r_{t+1}) + \\ &\quad s_{t+1}K^{-1}_{t+1}(1 - \epsilon_t K^{-1}_{t+1}W_{t+1}) \end{aligned} \quad (13)$$

In the update Equation (9), $\epsilon^-$ and $\epsilon^+$ are upper and lower bounds to prevent the forgetting factor from taking too small or too large values. Usually there is no problem in setting $\epsilon^-$ very close to zero but $\epsilon^+$ requires some adjustment for obtaining best performance. Also the learning rate $\alpha$ can be set to a very small positive number or a decreasing sequence . In our implementation we found that taking $\alpha_t = K_t$ gives very good performance and in addition does not require manual adjustment of the learning rate. Accordingly, Equation (9) is modified to

$$\epsilon_{t+1} = [\epsilon_t + K_{t+1}s_{t+1}V_t]^{\epsilon^+}_{\epsilon^-}. \quad (14)$$

Thus the final algorithm, as used in our experiments, consists of Equations (10)-(14).

| Estimation | SNR 10 | Variable |
|---|---|---|
| No Comp. | 46.6 | 51.6 |
| Seq 1.0 | 15.6 | 27.2 |
| Seq 0.95 | 17.4 | 20.3 |
| Seq Opt | 15.8 | 20.3 |
| Batch | 15.9 | 26.4 |

Table 1: Word error rates (%) for the baseline system (No Comp.) and different noise mean estimation algorithms.

Convergence of the algorithm in [6] was proved, under suitable conditions, in [7]. The proposed algorithm, however, differs from that in [6, 7]. For example, the weighting by the posteriors $p(m|z_t)$, which are functions of the noise mean and the forgetting factor, complicates the convergence analysis. Theoretical convergence analysis of the proposed recursion is outside the scope of this paper, and experimental results will be used to assess its performance.

## 5. EXPERIMENTAL RESULTS

The proposed algorithm has been evaluated on two different databases. The first one corresponds to a controlled evaluation on the 5K Wall Street Journal task corrupted by additive noise. The objective of this evaluation is to illustrate and study the behavior of the proposed noise compensation technique under an exact additive noise assumption. The second evaluation is carried out using a database recorded in a moving car, and involve real background noise conditions as well as various sources of distortions in the car environment.

### 5.1. Experiments on artificial noise

In this section, the proposed algorithm is tested on the 5K Wall Street Journal task, corrupted by additive noise. Two types of noise are used; the first is white Gaussian noise at 10 dB SNR, and the second is obtained by mixing two white Gaussian noises at 10 and 5 dB SNR respectively. The mixing coefficient varies linearly from zero to one within the utterance. The SI-84 training set (WSJ0) is used to construct tree clustered triphone HMMs using the algorithm in [13]. A 39 dimension feature vector consisting of 12 MFCC, log energy[1], and their first and second order derivatives is used by the system. A 5K lexicon and the standard trigram model provided by NIST are used in all experiments. Decoding is performed using a dynamic one pass decoder [14]. The word error rate for clean speech is 4.7%. For both types of noise the error rate increases to 46.6% and 51.6% respectively. We perform noise compensation using the VTS method outlined in Section 2, based on a 1st order approximation. The clean speech model $p(x)$ used for noise compensation is obtained by estimating a 64-Gaussian mixture model in the cepstral domain on a subset of the WSJ0 training data and mapping it into the log-spectral domain. The batch, sequential with constant forgetting, and sequential with optimal forgetting are tested for noise mean estimation. For the algorithm with optimal forgetting , $\epsilon^-$ was set to 0.001, while $\epsilon^+$ was varied. The results are summarized in Table 1. Seq $\rho$ stands for sequential estimation with forgetting factor $\rho$ , and Seq Opt stands for optimal forgetting with $\epsilon^+ = 0.05$.

[1] In order to map a feature vector back and forth between the cepstrum and log-spectrum domain, C0 (the first cepstral coefficient) is used as energy coefficient (rather than the traditional short-term energy)

| $\epsilon^+$ | SNR 10 | Variable |
|---|---|---|
| 0.05 | 15.8 | 20.3 |
| 0.10 | 16.5 | 19.6 |
| 0.15 | 17.1 | 20.5 |

Table 2: Word error rates (%) for optimal sequential estimation algorithm for different forgetting constant $\epsilon^+$.

As expected for the constant noise a forgetting factor of 1.0 (no forgetting) works best. Also a large increase in the word error rate is observed if the forgetting factor is changed to 0.95. For the variable noise a forgetting factor of 0.95 works substantially better than no forgetting. In both cases the optimal algorithm leads to similar performance to the best manually tuned forgetting factor. This is important in practical situations where environment rapidly changes from an utterance to another, or when no sufficient development data, for hand tuning of the forgetting factor, is available. The results of the batch algorithm, also shown in the table, show that it is highly desirable to use sequential estimation, especially for varying noise conditions.

To study the sensitivity of the optimal algorithm to the value of $\epsilon^+$, we have run experiments for $\epsilon^+ = 0.1$ and $0.15$. Note that given value of $\epsilon^+$ indicates a minimum value of the forgetting factor $\rho_{min} = 1 - \epsilon^+$. Results are shown in Table 2. We can observe that the performance of the algorithm is slightly sensitive to the value of $\epsilon^+$. However, this sensitivity is not as prominent compared to that of the constant forgetting algorithm to the forgetting factor. For example, for the variable noise, the error rate of the constant forgetting algorithm increases to 27.2% when the forgetting value moves from 0.95 to 1.0. On the other hand the error rate increases only to 20.5% when $\epsilon^+$ changes from 0.1 to 0.15.

### 5.2. Experiments on real noise

Our noise compensation algorithm is also evaluated on an in-house hands-free database (CARVUI database) recorded inside a moving car. The data was collected in Bell Labs area, under various driving conditions (highway/city roads) and noise environments (with and without radio/music in the background). About 2/3rd of the recordings contain music or babble noise in the background. Simultaneous recordings were made using a close-talking microphone and a 16-channel array of 1st order hypercardioid microphones mounted on the visor. A total of 56 speakers participated in the data collection, including many non-native speakers of American English. The recorded text is made of various materials, including phonetically balanced TIMIT sentences, some digits strings with 1 to 7 digits, and about 85 short command words, like “window up”, “turn radio off”, etc. The speech material from 50 speakers is used for training, and the data from the 6 remaining speakers is used for test, leading to a total of 4417 utterances available for training and 993 utterances for test. The data is recorded at 24kHz sampling rate and is down-sampled to 8kHz and followed by a MFCC feature extraction step for our speech recognition experiments. The recognition task consists of command words and digits string of unspecified length, modelled by a finite state grammar. In our experiments, data from 2 channels only are used. The first one is the close-talking microphone (CT), the second one is a single channel from the microphone array, referred to as Hands-Free data (HF) henceforward. The average SNR is about 21dB for the CT channel and 8dB for the HF channel. We should also point out that the test utterances are usually very short, ranging from 0.6 to 5.8 seconds with an average duration of 1.8 seconds.

| Test Data | Corr | Sub | Del | Ins | Err | S.Err |
|---|---|---|---|---|---|---|
| CT | 97.9 | 1.4 | 0.7 | 0.4 | 2.5 | 6.3 |
| HF | 74.0 | 10.4 | 15.5 | 1.1 | 27.1 | 44.8 |

Table 3: Recognition results (in %) of the close-talking (CT) microphone data and Hands-Free (HF) data using CT model.

| Test Data | Corr | Sub | Del | Ins | Err | S.Err |
|---|---|---|---|---|---|---|
| CT - Batch | 97.8 | 1.5 | 0.6 | 0.4 | 2.6 | 6.3 |
| CT - Seq. 0.95 | 97.9 | 1.5 | 0.6 | 0.4 | 2.5 | 6.2 |
| CT - Seq. Opt | 98.0 | 1.4 | 0.6 | 0.4 | 2.4 | 6.0 |
| HF - Batch | 92.6 | 5.5 | 2.0 | 1.8 | 9.2 | 21.7 |
| HF - Seq. 0.95 | 92.0 | 5.3 | 2.7 | 2.2 | 10.2 | 23.1 |
| HF - Seq. Opt | 92.2 | 5.4 | 2.5 | 2.1 | 9.9 | 23.4 |

Table 4: Recognition results (in %) on compensated CT and HF test data using CT model.

A set of triphone models is built on the CT training data using a decision tree state tying algorithm [13] with aggressive tying given the limited amount of training data. For the noise compensation, a 128-Gaussian mixtures model is trained in the cepstral domain on the same CT data and then mapped to the log-spectral domain. The decoder is tuned on the CT test data, leading to a word error rate of 2.5% and a string error rate of 6.3%. When recognizing the HF data, the error rates degrades significantly, mainly due to a large increase in Deletions, as indicated on Table 3.

The objective of our evaluation is to study whether the proposed noise compensation algorithm can improve the recognition of the HF data, without having to modify the recognition system, without unduly degrading recognition performance on the clean CT data. Recognition results are given in Table 4 after batch, sequential compensation with fixed forgetting factor and sequential compensation with optimal forgetting factor on both the CT and HF test data. This table illustrates that the proposed compensation algorithm does not affect the clean speech (CT) recognition, and no statistical difference is observed between the original results on uncompensated data and any of the compensated data. This is worth noticing since many noise compensation approaches tend to degrade performance on clean speech data. On the other hand, for the noisy channel (HF), the noise compensation approach leads to a significant error reduction compared to the baseline system, where more than 60% reduction of the word error rate is observed, from 27.1% down to about 10%. Again, no statistical differences are observed between the different estimation mode for the noise mean. This can be explained since the test utterances are on average very short (1.8 sec.), making it difficult to track noise fluctuations, if any. However, it illustrates the effectiveness of the proposed approach as a noise compensation algorithm. We should also emphasize that since the noise can be sequentially estimated, frame after frame, the proposed noise compensation approach is well suited for real time applications, is computationally efficient and does not require an explicit speech/non-speech detection.

## 6. CONCLUSION

We have presented a method for optimizing the forgetting factor in sequential estimation algorithms. Traditionally the forgetting factor is chosen based on a development set and no optimality criterion is involved. We applied the proposed method to sequential noise mean estimation in a VTS setting. The proposed algorithm was tested on a large vocabulary noisy speech recognition task with artificial noise and a small vocabulary car database with real life noise. The obtained performance is comparable to the best manually set forgetting factor. Due to its sequential nature, the algorithm is well suited to real-time applications. Future work includes applying the same principle to other sequential schemes, as sequential bias or affine transformation estimation.

## 7. REFERENCES


[1] M. J. F. Gales and S. J. Young. Cepstral parameter compensation for HMM recognition in noise. *Speech Communication*, 12:231–239, 1993.

[2] Y.Gong, "Speech recognition in noisy environments: A survey," Speech Communication, Vol.16, pp.261-291, April 1995.

[3] P.J.Moreno, B.Raj, and R.M.Stern, "A vector Taylor series approach for environment-independent speech recognition," in Proc. ICASSP, Atlanta, GA, May 1996, pp.733-736.

[4] N.S.Kim, "Nonstationary environment compensation based on sequential estimation," IEEE Signal Processing Letters, Vol.5, No.3, pp.57-59, March 1998.

[5] N.S.Kim, "Statistical linear approximation for environment compensation," IEEE Signal Processing Letters, Vol.5, No.1, pp.8-10, January 1998.

[6] A. Benveniste, M. Metivier, and P. Priouret, Adaptive algorithms and stochastic approximation. New York, Berlin: Springer-Verlag, 1990.

[7] H.J.Kushner, J.Yang, "Analysis of adaptive step-size SA algorithms for parameter tracking," IEEE Transactions on Automatic Control, Vol.40, No.8, pp.1403-1410, August 1995.

[8] L.Delphin-Poulat, C.Mokbel, and J. Idier, "Frame-Synchronous stochastic matching based on Kullback-Leibler information," in Proc. ICASSP, Seattle, WA, April 1998.

[9] M.Afify, "Sequential bias compensation for robust speech recognition," in Proc. EUROSPEECH, Budapest, Hungary, September 1999.

[10] V.Krishnamurthy, and J.B.Moore, "On line estimation of hidden Markov model parameters based on the Kullback Leibler information measure," IEEE Trans. Signal Processing, Vol.41, pp.2557-2573, Aug 1993.

[11] E.Weinstein, M. Feder, and A.V.Oppenheim, "Sequential algorithms for parameter estimation based on Kullback Leibler information measure," IEEE Trans. Signal Processing, Vol.38, pp.1652-1654, Sept. 1990.

[12] D.M. Titterington, "Recursive parameter estimation using incomplete data," J. Royal Statistical Society, vol. 46(B), pp.257-267, 1984.

[13] W. Reichl, and W. Chou, "A unified approach to incorporating general features in decision tree based acoustic modeling," Proc. ICASSP99, Phoenix, Arizona, March 1999.

[14] Q. Zhou, and W. Chou, "An approach to continuous speech recognition based on layered self-adjusting decoding graph," Proc. ICASSP97, Munich, Germany, April 1997.