Online Minimum Mean Square Error Filtering of noisy cepstral coefficients using a sequential EM algorithm

Tor André Myrvoll* and Satoshi Nakamura†

*Department of Telecommunications, NTNU, Trondheim, Norway
†ATR Spoken Language Translation Research Laboratory, Kyoto, Japan
myrvoll@tele.ntnu.no, nakamura@slt.atr.co.jp

Abstract

In this work we propose an online filtering algorithm that aims to alleviate the decrease we see in ASR performance when the speech is corrupted by additive noise. Using an initial estimate of the noise distribution, the algorithm updates the noise model on a frame synchronous basis. The minimum mean square error (MMSE) filtering is also performed at a frame per frame basis, using the most current noise model estimate at all times. The algorithm is compared to a batch version which uses several iterations of the EM-algorithm over the complete utterance to estimate the noise model, and it is demonstrated that the performance is as good or better at a fraction of the computational complexity when the noise is non-stationary.

1. Introduction

It is well known that mismatches between the training and testing conditions of an ASR system leads to a significant performance drop. Such mismatches include varying acoustic channels, speaker variation, additive noise or a combination of the previous conditions. In this work we will be concerned with the problem of additive noise, which is of particular interest when deploying an ASR system in a noisy environment.

In general there are two approaches available to us in the case of mismatch between the ASR system and the working environment: One can either change the acoustic model, usually a hidden Markov model (HMM), to reflect the new conditions, or one can compensate for the mismatch by transforming the acoustic feature vectors to better match the ASR system. The former approach, which is superior due to the data processing theorem[1], covers the well known MLLR[2] and MAP adaptation[3]. Another model adaptation approach that works directly under the assumption of the speech being corrupted by additive noise, is the parallel model combination(PMC)[4].

On the other hand, although suboptimal in the theoretical sense, the feature adaptation approach has been used successfully in various algorithms. In this work we try to recover the original, clean speech cepstral features using a non-linear, minimum mean square error filter, and so our approach is consistent with the feature adaptation approach.

As will be demonstrated in section 2, the linear mixing of noise and speech in the time/spectral domain, results in a highly non-linear combination of the speech and noise cepstral coefficients. Previous work like the vector Taylor series (VTS) approach[5] and the Jacobian approach[6], used linear approximations to circumvent the nonlinearities. An exact formulation of the noise estimation and cepstral filtering problem was presented in [7], where numerical integration routines were utilized to solve the estimation and filtering equations. An effective approximation to the integrals that both lowered the computational complexity and improved numerical stability was presented in [8].

In general there are two important deficiencies that has to be addressed with respect to the approach presented in [7, 8]. Further speed increase is still needed, and the important problem of non-stationary noise should be investigated. A single approach that addresses both of these issues is the use of online estimation instead of batch estimation as it is used now. Online estimation has been investigated earlier in the VTS setting with good some results[9].

This paper is structured as follows: In section 2 we give a brief overview of the theoretical basis of the nonlinear filtering approach that we base this work on. In the section 3 an online version of this filter is developed. In section 4 we present some experiments that demonstrate the validity of our approach, followed by some concluding comments.

2. Noise Parameter Estimation

When speech is corrupted by additive noise in the time or spectral domain, the effect in the log-spectral domain is a non-linear mixing of the noise and the speech,

\[
z_t = x_t + \log \left(1 + e^{n_t - x_t}\right),
\]

where \( x \) is the speech, \( n \) is the noise and \( z \) is the corrupted speech, all in the log-spectral domain and all indexed by the time \( t \). We follow common practice and assume that the noise \( n \) is Gaussian with unknown mean, \( \mu_n \), and variance, \( \sigma_n \), while the speech is modeled as a mixture distribution with known parameters. One way to alleviate the effect of the noise is to find the minimum mean-square-error estimate of the clean speech. The optimal mean-square-error (MSE) estimator is given by

\[
\hat{x} = E_{X|z}[x],
\]

where \( E_{X|z} \) is the conditional expectation operator. In order to perform this filtering we need to estimate the unknown parameters of the noise distribution.

2.1. Exact EM formulation

We have previously shown that the noise parameter estimation problem can be cast as a missing data problem, where the corrupted speech \( \{z_t\} \) is the incomplete data, and \( \{z_t, x_t\} \) are the complete data. This motivates the use of the EM-algorithm [10]}
to find the noise parameter estimates. We form the auxiliary function
\[ Q(\Lambda', \Lambda^{(i)}) = \mathbb{E}_{X \mid Z} \left[ \log p_{X \mid Z} \left( \{x_t, z_t\}_{t=1}^T \mid \Lambda' \right) \right] \]
where \( \Lambda = \{\mu_n, \Sigma_n\} \) are the parameters of the noise model. In [7] it is shown that the maximum of (3) with respect to the noise parameters is obtained using,
\[
\hat{\mu}_n = \frac{1}{T} \sum_{t=1}^T \int_{-\infty}^{x_t} n(z_t, x_t) p_{X \mid Z}(x_t \mid z_t, \Lambda^{(i)}) \, dx_t
\]
\[
\hat{\sigma}_n^2 = \frac{1}{T} \sum_{t=1}^T \int_{-\infty}^{x_t} (n(z_t, x_t) - \hat{\mu}_n)^2 p_{X \mid Z}(x_t \mid z_t, \Lambda^{(i)}) \, dx_t,
\]
where
\[
p_{X \mid Z}(x_t \mid z_t, \Lambda) = \int_{-\infty}^{x_t} n(z_t, x_t) p_{X \mid Z}(x_t \mid z_t, \Lambda^{(i)}) \, dx_t.
\]
The expression can be simplified even further using the fact that \( \frac{1}{e^x} = e^{-\log(e^x)} \). This enables us to include the denominator \( t \) in the exponential, which in turn can be written,
\[
e^{-\frac{1}{2} \left( \log(1-e^{t-\mu_n}) + \sigma_n^2 \right)^2} e^{-\mu_n + \frac{\sigma_n^2}{2}}.
\]
The same reasoning is used to include the term \( \frac{1}{\sqrt{\pi \sigma_n^2}} \) in every mixture component in \( p_{X}(1-t) + z_t \).

For every mixture component of \( p_{X}(1-t) + z_t \), we now have a product of two Gaussians, both functions in \( t \) and \( \{y_n\} \). The key step now is to do piecewise linear approximations of these two functions. Both functions are smooth over the majority of the \([0, 1]\) domain, with the exceptions of \( t = 0, 1 \), where \( \log(t) \) and \( \log(1-t) \) goes to minus infinity, but in these regions asymptotically exact approximations can be used. Exchanging \( \log(t) \) and \( \log(1-t) \) by approximations of the form \( at + b \) gives us a product of two functions that are Gaussians in the variable \( t \). The product of two Gaussians is well known to be Gaussian, and so the integral in every interval can be approximated by the integral of a Gaussian, for which an efficient functional form exists[11].

In this section we have shown that the noise distribution can be estimated using the EM algorithm, and we have also shown that efficient approaches to solving the numerical integrals exist. In the next section we will present an online version of this estimation algorithm based on the principle of sequential EM algorithms.

3. Sequential EM formulation

The following material is based on the general sequential algorithm using incomplete data that was presented in [12]. For a more detailed description of the principles described here, as well as other applications, the readers should consult the original article[12].

3.1. Theoretical motivation

Let \( y_1, \ldots, y_n \), be an ergodic sequence of observations whose joint parametric distribution is represented by the parameter vector \( \Lambda \). It is then well known that the expected log-likelihood of the observation \( y_{n+1} \), given by
\[
J(\Lambda) = \mathbb{E}_{y_{n+1}}[\log f_Y(y_{n+1} \mid \Lambda)],
\]

is maximized for \( \Lambda = \Lambda_0 \), that is the parameter vector corresponding to the true distribution. There is no way to maximize equation (11) directly, though, as \( \Lambda_0 \) is unknown. The key step is then to use the ergodicity of \( \{y_n\} \) to approximate the unavailable ensemble average by the time average. Thus, the stochastic approximation approach can be used to formulate the following recursion,
\[
\Lambda^{(n+1)} = \Lambda^{(n)} + \beta_n \nabla_{\Lambda} \log f_Y(y_{n+1} \mid \Lambda)
\]
which converges in the mean square error sense given the needed regularity conditions and the condition that,
\[
\lim_{n \to 0} \beta_n = 0, \quad \sum_{n=0}^{\infty} \beta_n = \infty, \quad \sum_{n=0}^{\infty} \beta_n^2 < M < \infty
\]

In the case of incomplete data the distribution function \( f_{Y}(y \mid \Lambda) \) is either unavailable or too involved to be used directly. Because of this we follow the same reasoning as for the EM algorithm, and define the auxiliary function
\[
Q_n(\Lambda, \Lambda') = \mathbb{E}_{Y \mid X} \left[ \log f_X(x_n \mid \Lambda) \right],
\]

The expression can be simplified even further using the fact that \( \frac{1}{e^x} = e^{-\log(e^x)} \). This enables us to include the denominator \( t \) in the exponential, which in turn can be written,
\[
e^{-\frac{1}{2} \left( \log(1-e^{t-\mu_n}) + \sigma_n^2 \right)^2} e^{-\mu_n + \frac{\sigma_n^2}{2}}.
\]
where \( x_n \) represents the complete data. One can define an object function for the incomplete problem corresponding to equation (11),
\[
J(\Lambda) = E_{\Lambda n}[Q_{\Lambda}(\Lambda, \Lambda')],
\]
where the expectation this time is over \( y_n \). Clearly, to maximize \( J(\Lambda) \) is to maximize the expected value of the auxiliary function, which in turn is known to have the same maximizer as \( \log f_{\Lambda}(y_n) \).

We now use the same reasoning that led us to equation (12), that is, the ensemble average implied by equation (15) is equal to the time average under the assumption of an ergodic observation sequence. Mathematically this is expressed using the recursion,
\[
\Lambda^{(n+1)} = \arg\max_{\Lambda} Q_{n+1}(\Lambda, \Lambda^{(n)}).
\]
(16)

The reliance on the ergodicity of the source may seem inconsistent with the one of the stated goals of this work, which was to handle non-stationary noise. A modification of the update algorithm of equation (16) is given by
\[
\Lambda^{(n+1)} = \arg\max_{\Lambda} \Psi_{n+1}(\Lambda)
\]
where
\[
\Psi_{n+1}(\Lambda) = \gamma_n \Psi_n(\Lambda) + Q_{n+1}(\Lambda, \Lambda^{(n)}).
\]
(18)
In the case that \( 0 < \gamma_n < 1 \) we obtain an exponential weighting that forgets older observations as the estimates are updated. This enables us to track changing noise characteristics at the price of less robust estimates.

3.2. The online estimation algorithm

The theory summed up in the previous section is directly applicable to our estimation problem. The auxiliary function \( Q_{t+1}(\Lambda, \Lambda^{(t)}) \) can be derived directly from equation (3) by looking at a single frame at the time, i.e.
\[
Q_{t+1}(\Lambda, \Lambda^{(t)}) = E_{X|Z} \left[ \log p_{X, Z}(x_{t+1}, z_{t+1}|\Lambda) | \Lambda^{(t)}, z_{t+1} \right]
\]
\[= \int \log p_{X, Z}(x_{t+1}, z_{t+1}|\Lambda) dP_{X|Z}(x_{t+1}|z_{t+1}, \Lambda^{(t)}), \]
(19)
Note that the sequence of estimates are indexed by the time \( t \) here. Setting \( \gamma_0 = 1 \) the recursion in equation (18) can be rewritten as
\[
\Psi_{t+1}(\Lambda) = Q_{t+1}(\Lambda, \Lambda^{(t)}) + \sum_{l=0}^{t} \gamma_l Q_{t-l+1}(\Lambda, \Lambda^{(t-l)}),
\]
(20)
which, using the simplification \( \gamma_t = \gamma \), can be reduced to
\[
\Psi_{t+1}(\Lambda) = \sum_{l=0}^{t} \gamma^l Q_{t-l+1}(\Lambda, \Lambda^{(t-l)}).
\]
(21)
For every \( \Psi_{t+1}(\Lambda) \) there are optimal values for the noise distribution mean, \( \mu_n^{(t+1)} \), and variance, \( \sigma_n^{2(t+1)} \). The following discussion will be on the mean estimate exclusively as the derivation of the standard deviation follows in the same manner.

Calculating the gradient of equation (21) and then solving for \( \mu_n^{(t+1)} \) gives us the following expression,
\[
\mu_n^{(t+1)} = \frac{\sum_{l=0}^{t} \gamma^l \hat{n}_{t-l}}{\sum_{l=0}^{t} \gamma^l},
\]
(22)
where
\[
\hat{n}_t = \int_{-\infty}^{t} n(z_t, x_t) p_{X|Z}(x_t|z_t, \Lambda^{(t-1)}) dx_t.
\]
(23)
We can rewrite this in terms of the previous mean estimate and the new expected mean based on the observations at time \( t + 1 \),
\[
\mu_n^{(t+1)} = \frac{\hat{n}_{t+1} + \gamma_{t+1} \mu_n^{(t)}}{1 - \gamma_{t+1}}.
\]
(24)
Assuming that the algorithm is stable at \( t = 0 \) we can rewrite the asymptotic update recursion as,
\[
\mu_n^{(t+1)} = (1 - \gamma) \hat{n}_{t+1} + \gamma \mu_n^{(t)}
\]
(25)
In the same way we can write the variance update recursion as
\[
\sigma_n^{2(t+1)} = (1 - \gamma) \hat{s}_{t+1} + \gamma \sigma_n^{2(t)}.
\]
(26)
Finally, using the current estimate of the noise model at time \( t \), we can find the filtered cepstral coefficients according to equation (2).

In the next section we will do a series of experiments where we compare the approach describes in this section with the standard batch approach from [8] on a subset of the Aurora 2 task, as well speech that had very non-stationary noise artificially added.

4. Experiments

We will perform two sets of experiments. First we will use a subset of the Aurora 2 task to show that the online filtering algorithm indeed succeeds in improving upon the baseline recognizer trained on clean speech. Next, we will use speech data where various types of non-stationary noise has been artificially added to show that the algorithm does improve upon the batch algorithm based on the assumption of stationary noise.

In all the experiments the recognizer is based on the standard Aurora 2 hidden Markov model training scripts. The features used are the first 13 cepstral coefficients with both velocity and acceleration parameters, which makes it 39 features all in all. Note that we use the 0th cepstral coefficient instead of log-energy. The experiments are all performed by denoising the noisy speech using either the online or the batch filter, and then performing ordinary recognition using the clean speech HMM.

The first task is the speech data with added subway noise from the Aurora 2 database. The noise is added at different signal to noise ratios (SNR), from 20 dB down to -5 dB in steps of 5 dB. We compare the online filter with the baseline and the batch filter in table 1. Clearly, the batch filter outperforms the online approach in this case, although the online filter still improves upon the baseline. This indicates that for short utterances like digit strings, the noise estimates obtained using a sequential EM algorithms are significantly inferior to estimates obtained using several EM iterations. The results also indicate that the subway noise can be considered stationary over the utterance, that is, there are no clear transients that would make the average noise estimate unsuitable for the filtering process.

The next set of experiments consists of the same clean utterances that we used in the previous experiments. We artificially
added Gaussian white noise to the utterances with time-varying SNRs over the length of the utterance. Three sets of noisy utterances were created. In all three cases we estimated the energy in the utterance as to control the SNR, and the added noise was in the form of Gaussian noise. In the first set, which we will call “step”, the first half of the utterance is clean, while the second half has an SNR of 0 dB. The second set, called “linear”, has noise that increases from no noise at the start of the utterance to 0 dB SNR at the end. The noise increases linearly in the standard deviation. The final set is called “saw”, and adds noise whose energy follows a sawtooth shape with period one second, the peak value corresponding to a 0dB SNR. Recognition results are summarized in table 2.

Table 2: Recognition results on speech corrupted by subway noise at different SNRs. The parameter $\gamma$ is the same as in the update equations (25) and (26). Three EM iterations was used to estimate the noise using the batch filter.

<table>
<thead>
<tr>
<th>SNR</th>
<th>Baseline</th>
<th>Online ($\gamma = 0.01$)</th>
<th>Online ($\gamma = 0.1$)</th>
<th>Batch</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5dB</td>
<td>10.72</td>
<td>13.72</td>
<td>12.68</td>
<td>20.08</td>
</tr>
<tr>
<td>0dB</td>
<td>20.94</td>
<td>26.53</td>
<td>23.15</td>
<td>43.78</td>
</tr>
<tr>
<td>5dB</td>
<td>45.26</td>
<td>53.95</td>
<td>45.72</td>
<td>72.34</td>
</tr>
<tr>
<td>10dB</td>
<td>73.87</td>
<td>80.10</td>
<td>74.49</td>
<td>88.49</td>
</tr>
<tr>
<td>15dB</td>
<td>92.08</td>
<td>91.71</td>
<td>89.10</td>
<td>94.14</td>
</tr>
<tr>
<td>&gt;0dB</td>
<td>96.90</td>
<td>95.73</td>
<td>94.75</td>
<td>96.53</td>
</tr>
<tr>
<td>Average</td>
<td>62.70</td>
<td>65.75</td>
<td>62.65</td>
<td>73.41</td>
</tr>
</tbody>
</table>

Table 2: Recognition results on speech corrupted by subway noise at different SNRs. The parameter $\gamma$ is the same as in the update equations (25) and (26). Three EM iterations was used to estimate the noise using the batch filter.

The experiments on non-stationary noise, however artificial, clearly shows that although the batch estimation algorithm can yield good results using an estimate of the “average” noise, the online approach achieves comparable results at a lower complexity and with no time-delay. The results also show that there is bound to be a trade-off between the ability to track fast changes and stability – a trade-off that is controlled by the exponential forgetting parameter $\gamma$. For slowly changing noise, as in the “lin” case, a small $\gamma$ tracks the noise fast enough, while the long memory gives stable estimates. For the “step” and the “saw”, smaller values of $\gamma$ are necessary to track the rapidly changing conditions.

5. Conclusions

In this work we have presented an online formulation of the noisy cepstral coefficient filtering problem. The approach is based on earlier work in which the noise distribution was estimated without any of the approximations associated with e.g. vector Taylor series. Under non-stationary noise conditions the online approach performed mostly as or a little bit worse than the batch version, but doing so at a reduction in computational complexity and elimination of the time-delay. Further work should include further studies of the performance under realistic noise conditions, as well as the choice of $\gamma$, the exponential forgetting factor. The rate of convergence under various conditions is also of great interest.

6. Acknowledgments

This research was supported in part by the National Institute of Information and Communications Technology. Dr. Myrvoll was also partly supported by the Department of Electronics and Telecommunications, Norwegian University of Science and Technology, and the Norwegian Research Council through the BRAGE program (http://www.tele.ntnu.no/projects/brage/index.php).

7. References