An Integrated Solution for Error Concealment in DSR Systems over Wireless Channels

Antonio M. Peinado, Ángel M. Gómez, Victoria Sánchez,
José L. Pérez-Córdoba, Antonio J. Rubio

Department of Signal Theory, Networking and Communications
Universidad of Granada, Granada, Spain

amp@ugr.es

Abstract

Distributed Speech Recognition involves the development of techniques to conceal the degradations that the transmission channel introduces in the speech features. This work proposes a low-complexity high-accuracy error concealment technique compatible with the DSR ETSI standards. This is achieved by combining three different techniques: fast MMSE estimation, Viterbi decoding with soft-data and subvector-based error detection. We also propose a method to extend this Viterbi decoding to dynamic features. The experimental results show the effectiveness of our proposal.

Index Terms: distributed speech recognition, error concealment, minimum mean square error estimation, soft-data Viterbi decoding.

1. Introduction

During the last years, distributed speech recognition (DSR) has aroused a big interest among researchers, developers and manufacturers, since it provides an efficient translation of Automatic Speech Recognition (ASR) technologies to mobile and IP network applications. DSR has a client/server architecture, where feature extraction and encoding is carried out by a local front-end and the speech recognition engine is placed in a remote back-end. In comparison with other solutions in which the whole ASR system is embedded in user devices, DSR has clear advantages such as the use of a thin client, which does not require maintenance by the user, or language portability. The interest in this new ASR paradigm has been reflected in the development of four ETSI standards (by the ETSI Aurora working group) and two RFC documents. The ETSI standards contain suitable procedures for feature extraction and compression (to be implemented in the client) and the corresponding feature decoding algorithms. They also describe a suitable format for implementation over mobile networks. Payload formats for implementation over IP networks are described in the RFC documents.

This work is focussed on the treatment of the errors introduced in the DSR bitstream during a wireless transmission by means of *error concealment* (EC), which is carried out in the back-end. The goal of EC in DSR is to provide robust and ubiquitous speech recognition over both circuit- and packet-switched networks. In this last case we assume that errors in the payload are allowed. The application of EC to DSR has been recently analyzed in [1], where several techniques are compared using the GSM EP error patterns to simulate channel degradation. For the case in which compatibility with the ETSI-DSR standards is required, the highest accuracy was provided by the forward-backward MMSE (FBMMSE) estimation that we proposed in [2] (MMSE stands for minimum mean square error). In particular, for the most degraded condition (EP3 pattern), FBMMSE achieves 98.83% of word accuracy, which is close to the performance in clean conditions (99.04%), while the Aurora EC algorithm only achieves 93.40% [3]. The main drawback of the FBMMSE technique is its high computational cost.

The goal of this work is to propose an EC scheme with the following specifications: compatibility with the ETSI-DSR standards, low complexity and high accuracy. In order to achieve these objectives, we propose the combination of the following techniques:

- Step-by-step MMSE (SSMMSE) estimation [3]. This is a suboptimal version of the FBMMSE estimation which only requires $6N-2$ floating point operations plus an VQ quantization (N is the codebook size) per estimate (FBMMSE requires $5N^2+5N-2$). It introduces a small performance reduction that will be compensated with the introduction of the two following techniques.
- Use of soft-data in the Viterbi decoding [4]. The Viterbi decoder used for recognition can be easily modified to consider the uncertainty inherent in the SSMMSE estimation. We only have to add the variance of every MMSE estimate (generated along with the estimate) to the variances of the HMM gaussians during the Viterbi decoding. The estimate variances can be considered measures of the estimate reliability and are added to the HMM variances to account for the unreliability of the estimated features. This missing-data technique has been successfully applied to static features in the aforementioned reference. In our work, we propose an effective extension of this technique to dynamic features. The problem of this extension is that these features are not transmitted, so that the MMSE-based EC algorithm cannot directly provide any reliability measures for them. We will introduce a method to obtain the needed reliability measures for dynamic features from those of the static features.
- Subvector-based error detection [1]. The Aurora error detection algorithm detects errors in frame pairs. By means of subvector error detection, we can detect errors in feature pairs. Therefore, the limits of an error burst are more precisely detected, so that we avoid that the SSMMSE-based

Work supported by MEC/FEDER project TEC2004-03829/TCM.

EC procedure unnecessarily degrades features not affected by the burst.

The rest of the paper is organized as follows. First, we briefly describe the experimental framework employed in this work. In section 3 we review how the MMSE estimation can be applied to obtain the estimates of the received (static) features and their reliabilities, and how these data can be used by the Viterbi decoder used for recognition. Then, we propose a possible solution for the computation of the reliabilities associated to the dynamic features. Finally, in section 4 we show the experimental results obtained over the GSM EP patterns when we apply the aforementioned techniques plus subvector-based error detection.

2. Experimental framework

2.1. Aurora framework

As we have previously pointed out, one of our premises is to develop an EC scheme fully compatible with the ETSI-DSR front-ends. Since these standards share the same compression scheme and we are considering neither acoustic noise nor recognition of tonal languages, we will only consider the first and simplest standard [5] (it does not include noise reduction). This front-end provides a 14-dimension feature vector containing 13 MFCCs (including the 0th order one) plus log-Energy. These features are grouped into pairs and quantized by means of seven Split Vector Quantizers (SVQ). All codebooks have a 64-center size (6 bits), except the one for MFCC-0 and log-Energy, which has 256 centers (8 bits). The bitstream is generated by grouping frames into pairs (88 bits) that are protected by a 4-bit CRC. At the back-end, error bursts are detected by means of a CRC checking and a consistency test. The Aurora mitigation algorithm can be summarized as follows: once a burst, containing 2B frames, is detected, the first B frames are substituted by the last correct frame before the burst and the last B ones by the first correct frame after the burst. In the case of a burst at the beginning of the utterance, the first correct frame after the burst is repeated in the degraded frames. A similar solution is applied for corrupted data at the end of the utterance. The recognizer used along this paper is the one provided by Aurora and uses eleven 16-state continuous HMM word models, (plus silence and pause, that have 3 and 1 states, respectively), with 3 gaussians per state. The training and testing data are extracted from the Aurora-2 database. Training is performed with 8440 clean sentences and test is carried out over set *A* (4004 clean sentences distributed into 4 subsets).

2.2. Transmission scheme

The transmission scheme employed in this work is depicted in Figure 1. After SVQ quantization, each feature pair is represented by a vector $\mathbf{c}$ ($\mathbf{c} \in \{\mathbf{c}^{(i)}; i = 0, \ldots, 2^M - 1\}$) ($M$=6,8 in this work) that, after bit mapping, is represented by a bit sequence $\mathbf{x} = (x(0), x(1), \ldots\ldots, x(M-1))$ ($\mathbf{x} \in \{\mathbf{x}^{(i)}; i = 0, \ldots, 2^M - 1\}$), where each bit is assumed to be bipolar ($x(k) \in [-1, +1]$). This sequence is transmitted, after channel encoding, through a digital channel. Every bit of the received bit sequence $\hat{\mathbf{x}}$ is determined by hard-decision as $\hat{x}(k) = sign[y(k)]$. Additionally, when an error burst is detected, MMSE estimation is applied to mitigate the possible bit errors contained in $\hat{\mathbf{x}}$.

As indicated above, the Aurora mitigation algorithm is suitable for bursty channels, and is tested in [6] under GSM error patterns EP1, EP2 and EP3. In this work, we present the final experimental results on these patterns, but, in order to experiment over a wider range of channel conditions, we also consider a simplified bursty channel model whose details can be found in [2].

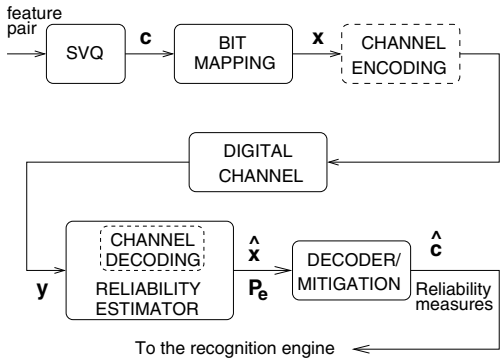

Figure 1: *Transmission scheme for a feature pair.*

3. MMSE estimation

The FBMMSE estimate [2] of a parameter vector at time t given the observation sequence $\hat{X}_0^T = (\hat{\mathbf{x}}_0, \ldots, \hat{\mathbf{x}}_T)$, where $\hat{\mathbf{x}}_0$ and $\hat{\mathbf{x}}_T$ are the last and first correctly received observations before and after an error burst, respectively, can be obtained as,

$$\hat{\mathbf{c}}_t = E[\mathbf{c}_t|\hat{X}_0^T] = \textstyle\sum_{i=0}^{N-1} \mathbf{c}^{(i)} \gamma_t(i) \qquad (0 < t < T) \qquad (1)$$

$$\gamma_t(i) \equiv P(\mathbf{x}_t^{(i)}|\hat{X}_0^T) = \alpha_t(i)\beta_t(i)/K \qquad (2)$$

where we have introduced the notation $\mathbf{x}_t^{(i)}$ for $\mathbf{x}_t = \mathbf{x}^{(i)}$, and where K is a normalization factor and $\alpha_t(i) \equiv P(\mathbf{x}_t^{(i)}|\hat{X}_0^t)$ and $\beta_t(i) \equiv P(\hat{X}_{t+1}^T|\mathbf{x}_t^{(i)})$ are the forward and backward conditional probabilities, respectively. Probabilities $\alpha_t(i)$ and $\beta_t(i)$ can be obtained by means of forward and backward recursions. In order to do this, we can model the signal source by an ergodic hidden Markov model (HMM). In our Aurora-based DSR system, each HMM models the generation of a given feature pair. Each state s_i of the model represents an SVQ codebook center $\mathbf{c}^{(i)}$ (or, equivalently, a codeword $\mathbf{x}^{(i)}$). The transition probabilities between states $a_{ij} \equiv P(\mathbf{x}_t^{(j)}|\mathbf{x}_{t-1}^{(i)})$ can be obtained from a simple analysis of the training data. The observation probabilities, defined as $b_i(\hat{\mathbf{x}}) \equiv P(\hat{\mathbf{x}}|\mathbf{x}^{(i)})$, can easily be computed from the Hamming distance between $\hat{\mathbf{x}}$ and $\mathbf{x}^{(i)}$ and considering an estimate of the average bit error probability p_e of the channel (see reference [2] for details).

The SSMMSE estimation is an approximation of FBMMSE that can be summarized as follows [3]:

1. The considered error burst of length $2B = T - 1$ is split in two halves of length B.
2. Processing of the first half: at every time step ($t = 1, \ldots, B$), the following estimate is computed,

$$\hat{\mathbf{c}}_t = E[\mathbf{c}_t|\hat{\mathbf{c}}_{t-1}, \hat{\mathbf{x}}_t] = \textstyle\sum_{i=0}^{N-1} \mathbf{c}^{(i)} \tilde{\alpha}_t(i) \qquad (3)$$

$$\tilde{\alpha}_t(i) = P(\mathbf{x}_t^{(i)}|\hat{\mathbf{c}}_{t-1}, \hat{\mathbf{x}}_t) \approx C a_{\phi(\hat{\mathbf{c}}_{t-1}),i} b_i(\hat{\mathbf{x}}_t) \qquad (4)$$

where C is a normalization constant, $\phi(\mathbf{c})$ is the index corresponding to the nearest SVQ center to vector $\mathbf{c}$ and $a_{\phi(\hat{\mathbf{c}}_{t-1}),i}$ is the transition probability from state $s_{\phi(\hat{\mathbf{c}}_{t-1})}$ to state s_i.

3. Processing of the second half: the same as for the first half but backwards ($t = 2B, \ldots, B+1$) instead of forwards.

As equation (3) indicates, the SSMMSE estimation uses the estimate of the previous step as if it was fully reliable. Thus, it avoids the high computational burden involved by the forward and backward recursions required for FBMMSE.

4. Soft-data Viterbi decoding

Although the MMSE estimation techniques previously presented are a powerful tool for EC, we must take into account that the resulting estimates are not fully reliable. We can consider that the estimate has an *evidence pdf* associated, that is, we are dealing with soft-data rather than the usual deterministic data. It has been shown that the Viterbi decoding carried out for recognition can be modified to account for this unreliability [7]. If the gaussian mixtures employed by the recognition HMMs have diagonal covariance matrices and the evidence pdf is gaussian, then the modification of the Viterbi decoder is especially simple: for a given feature x, the corresponding variance of every HMM gaussian must be increased by the variance $\sigma^2_{x,t}$ of the evidence pdf associated to the feature MMSE estimate x_t. The remaining problem is to obtain these evidence variances. We have seen in the previous section that, in general, the MMSE estimation provides feature estimates which are computed as expected values,

$$\hat{x}_t = E[x_t|\text{available data}] \tag{5}$$

Then, its corresponding evidence variance can be computed as,

$$\sigma^2_{x,t} = E[(x_t - \hat{x}_t)^2|\text{av. data}] = E[x_t^2|\text{av. data}] - \hat{x}_t^2 \tag{6}$$

This soft-data approach has an important drawback. It can directly be applied to static features. However, since the dynamic features are not transmitted, we do not have a direct form to compute their evidence variances. Then, they must be obtained from the only reliability information available at the receiver, that is, that of the static features. Therefore, if a dynamic feature Δx_t is computed as a weighted sum,

$$\Delta x_t = \sum_{k=-M}^{M} w_k x_{t+k} \tag{7}$$

then, its corresponding evidence variance can be obtained as,

$$\sigma^2_{\Delta x,t} = VAR\left[\sum_{k=-M}^{M} w_k x_{t+k}\right] \tag{8}$$

$$= \sum_{k=-M}^{M}\sum_{j=-M}^{M} w_k w_j COV[x_{t+k}, x_{t+j}] \tag{9}$$

We see that this result does not fit our goal of low complexity due to the huge amount of computation that involved by the cross-covariances contained in the previous expression. In order to solve this problem, we can assume that the random variables x_{t+k} ($-M \leq k \leq M$) involved in equation (9) are independent. Then, all the terms with $k \neq j$ become zero, so eqn. (9) can be written as,

$$\sigma^2_{\Delta x,t} \approx \sum_{k=-M}^{M} w_k^2 VAR[x_{t+k}] = \sum_{k=-M}^{M} w_k^2 \sigma^2_{x,t+k} \tag{10}$$

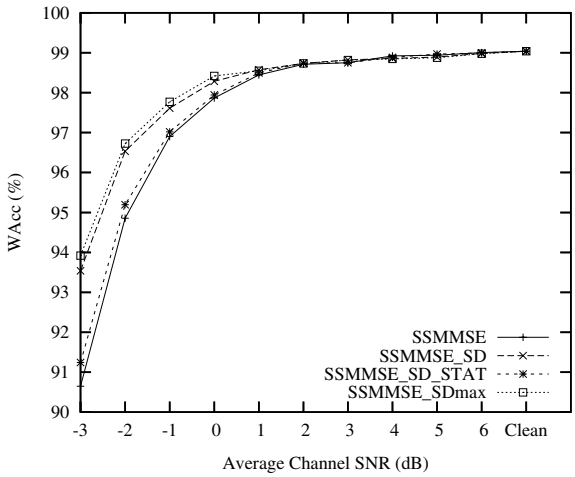

Figure 2: *DSR performance over a bursty channel.*

which only requires the evidence variances of the static features computed in equation (6).

The assumption of statistical independence for static features is obviously not true. However, this does not mean that the approximation of equation (10) is not useful. In figure 2, we compare the performance of the original SSMMSE method, SSMMSE plus soft-data Viterbi decoding (SSMMSE_SD) applying equation (10) and SSMMSE plus soft-data Viterbi decoding only applied to the static features (SSMMSE_SD_STAT). In this last case, the evidence variances of the dynamic features are set to zero. These experiments have been performed over a bursty channel with different average SNRs. We can see that the soft-data approach applied to all features (experiment SSMMSE_SD) clearly outperforms SSMMSE and SSMMSE_SD_STAT.

As previously shown, the SSMMSE method is an approximation of FBMMSE where at every time step we consider the estimate obtained in the previous step as fully reliable. Since this is not true, we can expect that the reliability of the SSMMSE estimates decreases towards the center of the error burst. Equivalently, we can say that the evidence variances increase towards the center of the burst. We can take into account this fact by performing the following postprocessing to the static evidence variances,

$$\sigma^2_{x,t} \leftarrow \max\left(\sigma^2_{x,t}, \sigma^2_{x,t-1}\right) \tag{11}$$

This ensures that the evidence variance increases monotonically towards the center of the burst. The performance of this variance postprocessing (plus the computation of $\sigma^2_{\Delta x,t}$ proposed in equation (10)) corresponds to experiment SSMMSE_SDmax in figure 2. It is shown that SSMMSE_SDmax provides the best results.

In figure 3 we present a comparison of the original Aurora EC algorithm with the original FBMMSE and SSMMSE techniques and with FBMMSE and SSMMSE plus soft-data Viterbi decoding (experiments FBMMSE_SD and SSMMSE_SDmax, respectively). It is shown that soft-data Viterbi decoding is useful for both FBMMSE and SSMMSE, although the improvement is much more noticeable for SSMMSE. We will see in the next section how we can further improve the SSMMSE-based methods in order to approximate the performance of those based on FBMMSE.

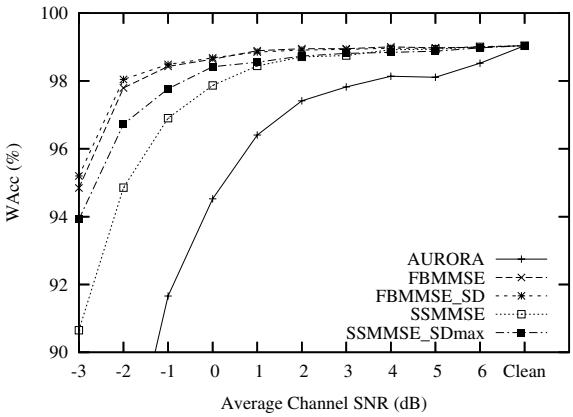

Figure 3: *DSR performance over a bursty channel.*

5. Experimental results over the GSM EP patterns

The three GSM EPx error patterns (x=1,2,3) have become quite standard to measure the robustness of DSR systems over wireless channels. They are bit error masks that are directly applied to the DSR bitstream by means of a XOR operation, and represent three different channel conditions of the GSM traffic channel (BER=0.00%,1.76%,3.48%). The results of the Aurora, FBMMSE and SSMMSE error concealment algorithms with hard- and soft-data Viterbi decoding are presented in table 1. We will pay more attention to the results over EP3, where we find more diffences. The most noticeable result is that SSMMSE with soft-data Viterbi decoding achieves almost 98% of word accuracy.

In order to obtain a further improvement of this result, we have also introduced the subvector-based error detection proposed in [1]. The Aurora standard can only discriminate whether a given frame pair is correct or not. However, once a frame pair is detected as erroneous, we can refine the localization of the errors by means of the same consistency test used in the Aurora standard. The Aurora consistency test determines the degree of continuity between the frames contained in a frame pair. This is carried out in two steps: 1) for every two consecutive feature pair subvectors (at times t and $t+1$) we check whether the two features of that feature pair do not sharply change from time t to $t+1$, and 2) a voting algorithm is applied to the 6 feature pairs to finally decide whether the frame pair is consistent or not. The subvector-based error detection avoids this second step and declares as erroneous only those feature pairs which are not consistent.

Table 2 presents the same techniques as table 1 but introducing subvector-based error detection. We can extract three interesting conclusions. First, the performance of the Aurora EC algorithm is clearly increased. Second, the performance of the FBMMSE technique is damaged by the subvector-based error detection. A possible explanation for this behavior is that, unlike Aurora or SSMMSE, the FBMMSE technique provides very good results even for long error bursts. Then, it is preferable to mitigate long but well detected bursts than shorter but erroneously detected bursts. Finally, the most important result is that SSMMSE approximates the performance of FBMMSE.

EC	Hard Data			Soft Data		
Method	EP1	EP2	EP3	EP1	EP2	EP3
AURORA	99.04	98.94	93.40	-	-	-
FBMMSE	99.04	99.02	98.83	99.04	99.02	98.80
SSMMSE	99.04	99.00	97.55	99.04	99.00	97.98

Table 1: *Word accuracies achieved by Aurora, FBMMSE and SSMMSE with hard- and soft-data Viterbi decoding over the GSM EP errors patterns.*

EC	Hard Data			Soft Data		
Method	EP1	EP2	EP3	EP1	EP2	EP3
AURORA	99.04	99.00	97.74	-	-	-
FBMMSE	99.04	99.02	98.48	99.04	99.02	98.71
SSMMSE	99.04	99.01	98.11	99.04	99.02	98.52

Table 2: *Word accuracies achieved by Aurora, FBMMSE and SSMMSE with hard- and soft-data Viterbi decoding over the GSM EP errors patterns using subvector error detection.*

6. Conclusions

In this work we have searched for an error concealment algorithm for DSR with three constraints: compatibility with the ETSI DSR standards, low complexity and high accuracy. This objective has been achieved by combining three different and complementary techniques: step-by-step MMSE estimation, Viterbi decoding with soft-data and subvector-based error detection. In particular, we have proposed a procedure to extend the Viterbi decoding with soft-data to dynamic features with a very low computational cost. We have shown that this extension is very effective when combined with step-by-step MMSE estimation.

7. References

[1] Z.H. Tan, P. Dalsgaard, and B. Lindberg, "Automatic speech recognition over error-prone wireless networks," *Speech Communication*, vol. 47, pp. 220–242, May 2005.

[2] A.M. Peinado, V. Sanchez, J.L. Perez, and A. de la Torre, "HMM-based channel error mitigation and its application to distributed speech recognition," *Speech Communication*, vol. 41, pp. 549–561, 2003.

[3] A.M. Peinado, V. Sanchez, J.L. Perez, and A.J. Rubio, "Efficient MMSE-based channel error mitigation techniques. Application to distributed speech recognition over wireless channels," *IEEE Trans. on Wireless Communications*, vol. 4, no. 1, pp. 14–19, January 2005.

[4] R. Haeb-Umbach and V. Ion, "Soft features for improved distributed speech recognition over wireless networks," in *Proc. of ICSLP'2004*, October 2004, Jeju Island, Korea.

[5] ETSI, "ETSI ES 201 108 v1.1.3. Distributed Speech Recognition; Front-end Feature Extraction Algorithm; Compression Algorithms," Tech. Rep. ETSI ES 201 108, ETSI, April 2003.

[6] D. Pearce, "Enabling New Speech Driven Services for Mobile Devices: An overview of the ETSI standards activities for Distributed Speech Recognition Front-ends," in *AVIOS 2000: The Speech Applications Conference*, May 2000, USA.

[7] A. C. Morris, J. Barker, and H. Bourlard, "From missing data to maybe useful data: Soft data modelling for noise robust ASR," in *Proc. WISP'01*, Stratford-upon-Avon, UK, 2001.