Multi-Resolution Soft Features for Channel-Robust Distributed Speech Recognition

Valentin Ion, Reinhold Haeb-Umbach

University of Paderborn
Dept. of Communications Engineering
33098 Paderborn, Germany
{ion,haeb}@nt.uni-paderborn.de

Abstract

In this paper we introduce soft features of variable resolution for robust distributed speech recognition over channels exhibiting packet losses. The underlying rationale is that lost feature vectors can never be reconstructed perfectly and therefore reconstruction is carried out at a lower resolution than the resolution of the originally sent features. By doing so, enormous reductions in computational effort can be achieved at a graceful or even no degradation in word accuracy. In experiments conducted on the Aurora II database we obtained for example a reduction of a factor of 30 in computation time for the reconstruction of the soft features without an effect on the word error rate. The proposed method is fully compatible with the ETSI DSR standard, as there are no changes involved in the front-end processing and the transmission format.

Index Terms: distributed speech recognition, error concealment, uncertainty decoding, soft feature

1. Introduction

The term Distributed Speech Recognition (DSR) stands for a client-server architecture, where feature extraction is carried out on a thin client, typically a mobile device, the features are transmitted over a communication network, and the actual recognition is carried out on a back-end server. While this approach has several advantages, such as low computational requirements on the client and ease of maintainability of the recognition engine and the application on the server side, it encompasses also the problem of sensitivity towards transmission errors.

Several methods have been proposed to overcome the degradation of the speech recognition performance due to unavoidable transmission errors in a DSR scenario [1]. One way to view at the problem is to cast it in the missing feature framework [2]. First, the location of missing features has to be detected, which is trivial for a channel characterized by packet losses, but is less trivial on channels exhibiting bit errors [3]. Second, the missing features are reconstructed. Here, a particularly powerful method is to formulate it as a Bayesian estimation problem and compute the maximum a posteriori (MAP) estimate of the sent feature, given the received ones [4]. Since the features have been quantized prior to transmission, the forward-backward algorithm can be applied for reconstruction. Finally, a measure of reliability can be computed for the reconstructed features and forwarded to the recognizer, which then implements an uncertainty decoding rule [5].

In this paper we are mainly concerned with the second issue and to some extent with the third. In the MAP-based reconstruction the sent features represent the states of a hidden Markov Model, which has as many states as there are possible values N for the sent feature. Assuming for simplicity that any state can follow any other state, the complexity of the Forward-Backward (FB) algorithm can be estimated to be on the order of $2N^2$ per frame, where the factor of 2 is due the fact that there is a forward *and* a backward recursion. In the split vector quantization scheme used in the ETSI standard [6] codebooks of up to size 8 bits are used. Therefore the number of computations for the FB algorithm is on the order of 2^{17} per frame. While the true complexity is lower, since the HMM is not fully connected, this consideration still shows, that a reduction of the computational effort is highly desirable.

In this paper we show that indeed the complexity can be drastically reduced by up to 30 times, without degrading the recognition accuracy. This is achieved by using a lower-resolution representation of the feature during error bursts, which translates to a reduced number of states to be visited by the FB algorithm. The underlying assumption is that lost features can never be reconstructed perfectly and therefore the reconstructed feature can afford a coarser resolution.

This paper is structured as follow. The next section briefly reviews speech recognition with soft features. Section 3 presents two possible source models involved in the posterior computation, one modeling only the static components of a feature vector and the other modeling also the dynamic ones. Section 4 deals with the multi-resolution quantization of feature vectors and estimation of feature posteriors. Section 5 introduces an additional method to reduce the computational load by selecting the resolution depending on the length of the error burst. Finally, the experimental results are presented in in Section 6, followed by conclusions in the last section.

2. Soft feature speech recognition

In [5] we showed how the classical Bayesian framework of automatic speech recognition (ASR) needs to be adapted for recognition with unreliable or soft features. Instead of evaluating the HMM state s_t dependent observation probability of the acoustic model $p(\mathbf{x}_t|s_t)$, in one, possibly erroneous, "point" $\mathbf{x}_t$, we integrate over all possible sent features, weighted by their posterior probability:

$$\int p(\mathbf{x}_t|s_t)\frac{p(\mathbf{x}_t|\mathbf{Y})}{p(\mathbf{x}_t)}d\mathbf{x}_t \tag{1}$$

The subscript t denotes the time-index and $\mathbf{Y}$ is the sequence of received features involved in the posterior computation.

In case of an error-free transmission the posterior $p(\mathbf{x}_t|\mathbf{Y})$

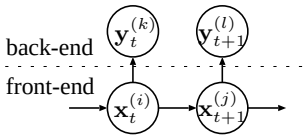

Figure 1: *Source-channel model of a DSR system*

becomes a Dirac delta impulse, and (1) reduces to the ordinary observation probability $p(\mathbf{x}_t|s_t)$ (the term $p(\mathbf{x}_t)$ can then be dropped as it is a constant). In case of a completely unreliable transmission the posterior $p(\mathbf{x}_t|\mathbf{Y})$ tends to the prior $p(\mathbf{x}_t)$, as the received feature becomes uninformative. Thus, the observation probability tends to unity, which is equivalent to a marginalization of that feature.

In [5] it was shown that (1) can be evaluated without numerical integration if the posterior pdf is Gaussian. Although this assumption is debatable, it is one way to make (1) tractable. Therefore, in this work we attempt to estimate the parameters of a Gaussian distribution which best approximates the true posterior.

3. Source-channel modeling

The source-channel model under consideration is depicted in Fig. 1. The feature vectors $\mathbf{x}_t$ computed in the front-end are assumed to be produced by a Markov source. They are quantized prior to transmission. In the ETSI standard for DSR [6] the 14 components of a feature vector are grouped into seven two-dimensional subvectors $sv_1 \cdots sv_7$, and each subvector is quantized separately. Since the feature components are uncorrelated, the resulting subvectors can be considered statistically independent and therefore generated by independent Markov sources. The quantization indices are sent over the digital channel as bit patterns. In Fig. 1 the short-hand notation $\mathbf{x}_t^{(i)}$ denotes the quantized vector at time t, where (i) indicates the identity of the transmitted bit pattern. The feature $\mathbf{y}_t^{(k)}$ observed at the channel output may be different from the sent one due to transmission errors. This can be described by the channel transition probability $p(\mathbf{y}_t^{(k)}|\mathbf{x}_t^{(i)})$. This conditional probability depends on the time varying properties of the channel and can be evaluated as shown in [3].

Given a sequence of T observations, $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_T]$, the forward-backward (FB) algorithm [3] delivers the posterior state distribution $p(\mathbf{x}_t^{(i)}|\mathbf{Y})$ at each t. From this the continuous posterior $p(\mathbf{x}_t|\mathbf{Y})$ is obtained as:

$$p(\mathbf{x}_t|\mathbf{Y}) = \sum_{i=1}^{N} p(\mathbf{x}_t|\mathbf{x}_t^{(i)}) \cdot p(\mathbf{x}_t^{(i)}|\mathbf{Y}) \tag{2}$$

where N is the number of quantization clusters and the term $p(\mathbf{x}_t|\mathbf{x}_t^{(i)})$ is the cluster conditioned probability density function of $\mathbf{x}_t$. This cluster conditioned pdf is modeled as a Gaussian: $p(\mathbf{x}_t|\mathbf{x}_t^{(i)}) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_t^{(i)}, (\sigma_t^2)^{(i)})$, where the within-cluster variance $(\sigma_t^2)^{(i)}$ is obtained from training data.

In order to simplify the evaluation of the observation probability (1), the posterior $p(\mathbf{x}_t|\mathbf{Y})$ is approximated by a Gaussian pdf: $\mathcal{N}(\mathbf{x}_t; \mu_t, \sigma_t^2)$. The parameters μ_t and σ_t^2 of this Gaussian can be obtained by finding that Gaussian which has the smallest Kullback-Leibler divergence to the original posterior $p(\mathbf{x}_t|\mathbf{Y})$ given in (2). This delivers the following estimates:

$$\mu_t = \sum_{i=1}^{N} p(\mathbf{x}_t^{(i)}|\mathbf{Y}) \cdot \mathbf{x}_t^{(i)} \tag{3}$$

$$\sigma_t^2 = \sum_{i=1}^{N} p(\mathbf{x}_t^{(i)}|\mathbf{Y}) \cdot \left[(\mathbf{x}_t^{(i)} - \mu_t)^2 + (\sigma_t^2)^{(i)}\right] \tag{4}$$

This result makes intuitively sense: The mean μ_t of the Gaussian is equal to the mean of the posterior, and the variance is the sum of the between-cluster variance, i.e. the variance of the cluster means, and the average within-cluster variance.

At high resolution, i.e. if N is large, the within-cluster variance is negligibly small, such that (4) simplifies to

$$\sigma_t^2 \approx \sum_{i=1}^{N} p(\mathbf{x}_t^{(i)}|\mathbf{Y}) \cdot (\mathbf{x}_t^{(i)} - \mu_t)^2 \tag{5}$$

In the case of absence of transmission errors, $p(\mathbf{y}_t^{(k)}|\mathbf{x}_t^{(i)})$ is a Delta pulse $\delta(k-i)$, resulting in a Delta posterior at that time. Therefore the FB algorithm needs to be performed only inside isolated erroneous regions (bursts), i.e. when $p(\mathbf{y}_t^{(k)}|\mathbf{x}_t^{(i)})$ is not a Delta pdf. Then the FB recursions are initialized using the last reliable feature before and the first reliable feature after the burst.

This source-channel model can be improved by augmenting the feature vector with its temporal derivatives. This leads to improved reconstruction performance although there are no observations of the dynamic features at the back-end, as they are not actually transmitted [5].

4. Multi-resolution quantization

The vector quantization scheme of the ETSI DSR standard uses $M = 6$ bits for $sv_1 \cdots sv_5$, 5 bits for sv_6 and 8 bits sv_7. The resulting quantization error insures a good trade-off between loss of recognition accuracy and limited channel bit rate.

The idea of this work is to assume that the source emits features quantized at a lower resolution within the burst period. The motivation for this is that, due to channel errors, the sent feature cannot be recovered at original resolution, thus a coarse representation should be sufficient.

This can be explained by the following consideration: if the channel is completely unreliable, the posterior $p(\mathbf{x}_t|\mathbf{Y})$ must equal the a priori pdf $p(\mathbf{x}_t)$ of the features, no matter which resolution was used for the quantization. Indeed, this result can be obtained from eq. (2). For this extreme case we can assume that a 1-centroid codebook was used to quantize the source ($N = 1$), i.e. there is only one term in the sum of eq. (2), and $p(\mathbf{x}_t^{(i)}|\mathbf{Y}) = 1$. The source model has only one state whose cluster conditioned probability $p(\mathbf{x}_t|\mathbf{x}_t^{(i)})$ is the prior pdf. Using this in (2) we indeed obtain $p(\mathbf{x}_t|\mathbf{Y}) = p(\mathbf{x}_t)$.

Note that the assumption of lower resolution during error bursts does not incur any modification of the standard. The error mitigation scheme is still fully standard compatible!

To get the channel transition probability of lower resolution features, $p(\mathbf{y}_t^{(k)}|\mathbf{x}_t^{(i)})$ must be projected on the lower resolution space by:

$$p(\mathbf{y}_t^{(k)}|\mathbf{x}_t^{(n)}) = \sum_{i=1}^{2^M} p(\mathbf{y}_t^{(k)}|\mathbf{x}_t^{(i)}) \cdot p(\mathbf{x}_t^{(i)}|\mathbf{x}_t^{(n)}) \tag{6}$$

where $\mathbf{x}_t^{(n)}$ denotes the n-th centroid of the lower resolution codebook, $p(\mathbf{y}_t^{(k)}|\mathbf{x}_t^{(i)})$ are the channel transition probabilities and $p(\mathbf{x}_t^{(i)}|\mathbf{x}_t^{(n)})$ represents the probability of sent feature $\mathbf{x}_t^{(i)}$ when its quantization at lower resolution $\mathbf{x}_t^{(n)}$ is known. This last term is assumed constant for those i's falling in the n-th lower resolution cluster and zero otherwise.

For each subvector we trained vector quantizers on the Aurora 2 training set, with $M-1$, $M-2$, $\cdots$, 0 bits resolution using a Generalized Lloyd Algorithm. Similarly, we trained vector quantizers for the dynamic components, delta and acceleration of each subvector.

5. Burst length dependent processing

The transmission over the public Internet is a prominent example for transmission encountering packet losses. According to [7] the packet losses are bursty and they can be reproduced by a 2-state Gilbert model. The resulting burst length distribution is exponential while the short bursts are predominant. In the experimental section of this work it can be observed that for channel conditions with low and moderate losses, i.e. short bursts, the performance obtained with low resolution is almost the same as that by using the original front-end resolution, however at a much lower computational expense. Therefore, we propose to choose the resolution setting for each burst according to its length. The appropriate mapping of burst length to resolution can be experimentally chosen as a trade-off between concealment effectiveness and the available computing resources. A simple solution is to compare the burst length with a predefined threshold and perform low resolution processing for shorter bursts and higher resolution processing for longer one. Since the threshold can be set so that only a small amount of burst exceeds it, the average computing time is reduced correspondingly.

6. Experimental results

In this section we present the recognition accuracies obtained using the proposed error concealment method as well as those obtained with the standardized scheme for DSR. Further, an approximate measure of computational complexity is used to assess the computational load.

The experimental setup consisted of the standard front-end for DSR, the packet-channel model and the back-end with error concealment and soft feature speech recognition. Each packet contained two consecutive feature vectors. The bursty packet losses were induced by a 2-state Markov chain, see [7], characterized by the conditional loss probability clp and the mean loss probability mlp. The set of investigated conditions are summarized in Table 1.

Table 1: *The conditional loss probability and mean loss probability of the four simulated network conditions.*

Condition	C1	C2	C3	C4
clp	0.147	0.33	0.5	0.6
mlp	0.006	0.09	0.286	0.385

The recognition task consisted of 4004 utterances of AURORA 2 database, clean test set. In the absence of channel errors the word accuracy (WAcc) was 99,14%.

The transition probabilities of the Markov source, which models the generation of the clean features, were estimated for each quantizer resolution on the clean training set of the same database.

6.1. Source model for static components only

Table 2 shows the WAcc's obtained with different fixed resolution settings, together with a measure of complexity per frame and subvector. The digit string in the first column indicates the number of bits used for each subvector. The first string 6666658 is the resolution of the quantization scheme used in the ETSI-DSR standard: 6 bits for first five subvectors $sv_1, \ldots, sv_5$, then 5 bits for sv_6 and 8 bits for the last subvector sv_7. For comparison, NFR denotes the nearest frame repetition approach used by ETSI-DSR, [6]. The parameters of the posterior pdf were computed using (3) and (5).

The second column denotes a measure of the computational complexity per frame. As there are as many states as there are vector quantization centroids, and assuming that each centroid may follow each, the complexity of the FB algorithm is on the order of $2N^2$ per frame, where N is the number of centroids and the factor of 2 is due the fact that there is a forward and a backward recursion. In the unequal resolution of the ETSI quantization scheme the subvector with the highest resolution, i. e. sv_7 with a resolution of 8 bits, gives a computational complexity on the order of $2N^2 = 2(2^8)^2 = 2^{17}$, which clearly dominates the contributions of all the other subvectors to the computational complexity. For the other resolutions studied in the Table 2, we counted the contribution of each subvector. Note that this is only a rough estimate of the computational complexity, since the HMM states are not fully connected in practice [3].

Aditionally, the computing time of error concealment per lost frame, measured on a workstation with 2.3 GHz Intel Xeon 5140 at 100% processor load, is given in the third column. Comparing the complexities of resolution 6666658 and 5555555 it can be seen that if all HMM states were connected, reduction of complexity by a factor of 9 would be possible without any performance loss. Practically, the computing time was shortened by the factor 6.6.

However, at resolutions lower than 4 bits the performance starts to degrade, where the degradation is strongest for the worst channel model C4.

Table 2: *Results for a source model for static components only. Posterior variance computation according to eq. (5).*

	$\mathcal{O}$	t[μs]	C1	C2	C3	C4
NFR	–		99.14	98.94	97.70	94.98
6666658	$\simeq 2^{17}$	337	99.10	99.02	98.32	96.87
6666657	$\simeq 2 \cdot 2^{15}$	190	99.10	99.02	98.27	96.81
6666656	$\simeq 6 \cdot 2^{13}$	148	99.10	99.00	98.27	96.87
5555555	$\simeq 7 \cdot 2^{11}$	51.0	99.10	99.00	98.24	96.79
4444444	$\simeq 7 \cdot 2^{9}$	17.5	99.10	98.98	98.17	96.81
3333333	$\simeq 7 \cdot 2^{7}$	6.4	99.10	98.93	98.20	96.63
2222222	$\simeq 7 \cdot 2^{5}$	3.2	99.07	98.73	97.41	95.07
1111111	$\simeq 7 \cdot 2^{3}$	1.93	99.07	97.64	92.33	86.86

Table 3 presents the word accuracies when the exact variance computation is used, i.e. when (5) is replaced by (4). As this has only an effect for very low resolution, results are only given for the coarsest three resolutions of Table 2. It can be seen that some of the performance degradation can be recovered.

Table 3: *Results for a source model for static components only. Posterior variance computation according to eq. (4).*

	$\mathcal{O}$	C1	C2	C3	C4
3333333	$\simeq 7 \cdot 2^7$	99.10	98.93	98.12	96.55
2222222	$\simeq 7 \cdot 2^5$	99.08	98.83	97.43	95.43
1111111	$\simeq 7 \cdot 2^3$	98.09	98.46	95.27	91.44

6.2. Source model for static and dynamic components

In [5] we have demonstrated that improved recognition accuracy can be achieved if a Markov model is used for both the clean static and dynamic components of the feature vector. In the second experiment we evaluated the improvement by using this augmented source model. In order to limit complexity only a coarse quantization was used: 3 bits for first-order derivatives (velocity) and 1 bit for second-order derivatives (acceleration). Still, the complexity increase is considerable. While for the quantization table of the static components only, the finest resolution, i.e. sv_7 with 8 bit quantization, resulted in a value of the complexity measure of 2^{17}, this is now increased to $2(2^{8+3+1})^2 = 2^{25}$ in the case of a source model for static and dynamic features. Note that the actual complexity is much lower since the HMM transition matrix is sparse at that resolution. While the number of bits for the dynamic features was kept fixed at 3 bit for delta and 1 bit for delta-delta, the resolution of the static components was successively decreased, as is indicated by the left column of Table 4. The results show that for resolutions down to 5 bit, dynamic features come with noticeable improvements. Unfortunately the increase in complexity is enormous and makes it questionable whether the increased word accuracy is worth the additional effort. At resolutions of 4 bits and lower, the word accuracy is limited by the resolution of the static features, i.e. the augmented source model does no longer pay off in increased word accuracy.

Table 4: *Results for a source model for static and dynamic components: 3 bit delta, 1 bit delta-delta.*

	$\mathcal{O}$	t[ms]	C1	C2	C3	C4
6666658	$\simeq 2^{25}$	64	99.10	99.07	98.47	97.46
6666657	$\simeq 2 \cdot 2^{23}$	30	99.10	99.06	98.44	97.38
6666656	$\simeq 7 \cdot 2^{21}$	22	99.10	99.07	98.47	97.40
5555555	$\simeq 7 \cdot 2^{19}$	7.3	99.10	99.03	98.39	97.29
4444444	$\simeq 7 \cdot 2^{17}$	2.3	99.10	99.02	98.27	96.83

6.3. Reduction of mean complexity by burst length dependent reconstruction

Using the model with static components only, we next illustrate the reduction of complexity based on the method proposed in Section 5. As can be seen in the Table 2, the word accuracies at C1 and C2, where the short bursts predominate, are less sensitive to low resolutions than C3 and C4 which exhibit longer bursts. Therefore processing the short burst at lower resolution than the longer one should gracefully degrade the accuracy while the mean complexity decreases correspondingly. For simplicity, we set the resolution to 3 bit if the burst length does not exceed a threshold of B frames, and 4 bit otherwise.

Table 5 shows the percentage of erroneous frames processed at 3 bit and 4 bit resolution under C4 scenario. The threshold setting is given in the first column. The last two columns present the WAcc's and the measured mean processing time per frame. The results show that setting the threshold B to 8, the accuracy decreases only marginally from 96.81 to 96.71%, although only 30% of erroneous frames are reconstructed at 4-bit resolution. At the same time the processing time is reduced from 17.5 to 10μs. This corresponds to a reduction of processing time relative to 6666658 (337μs) by a factor of about 30.

Table 5: *Percentage of frames processed at 3 and 4 bit resolution for $C4$ condition, WAcc's and processing time per frame depending on the burst length threshold.*

B	3-bit [%]	4-bit [%]	WAcc	t[μs]
1	0	100	96.81	17.5
8	70	30	96.71	10
48	100	0	96.63	6.4

7. Conclusions

In Distributed Speech Recognition, feature vectors lost during error bursts can not be reconstructed perfectly. We showed that by attempting to reconstruct a lost feature at a lower resolution than at original resolution of the sent feature, drastic complexity reduction is achieved at graceful or even no degradation in recognition accuracy. Moreover, the approach can be easily scaled to the available computing resources and does not incur changes of the standard DSR front-end.

8. Acknowledgements

This work was supported by Deutsche Forschungsgemeinschaft under contract number HA 3455/2-3.

9. References

[1] Z.-H. Tan, P. Dalsgaard, and B. Lindberg, "Automatic speech recognition over error-prone wireless networks," *Speech Communication*, vol. 47, no. 1-2, pp. 220–242, Sep.-Oct. 2005.

[2] Andrew Morris, Jon Barker, and Herv Bourlard, "From missing data to maybe useful data: soft data modelling for noise robust ASR," *Proc. WISP*, vol. 06, 2001.

[3] V. Ion and R. Haeb-Umbach, "Uncertainty decoding for distributed speech recognition over error-prone networks," *Speech Comm.*, vol. 48, pp. 1435–1446, 2006.

[4] A.M. Peinado, V. Sanchez, J.L. Perez-Cordoba, and A. de la Torre, "HMM-based channel error mitigation and its application to distributed speech recognition," *Speech Communication*, vol. 41, no. 6, pp. 549–561, Nov. 2003.

[5] Valentin Ion and Reinhold Haeb-Umbach, "Improved source modeling and predictive classification for channel robust speech recognition," in *Proc. of Interspeech, Pittsburgh*, 2006.

[6] ES.202.050, "Speech processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms," *ETSI*, Oct 2002.

[7] C. Boulis, M. Ostendorf, E:A. Riskin, and S. Otterson, "Graceful degradation of speech recognition performance over packet-erasure networks," *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 8, pp. 580–590, Nov. 2002.