SOFT DECODING OF TEMPORAL DERIVATIVES FOR ROBUST DISTRIBUTED SPEECH RECOGNITION IN PACKET LOSS

Alastair James and Ben Milner

School of Computing Sciences, University of East Anglia, Norwich, U.K.
{a.james, b.milner}@uea.ac.uk

ABSTRACT

The aim of this work is to improve distributed speech recognition accuracy in packet loss by considering the effect of loss on the temporal derivatives of the feature vector. Analysis of temporal derivatives reveals they suffer severe distortion when static vectors are lost in times of packet loss. The application of missing feature theory and soft-decoding techniques are considered for compensating against packet loss at the decoding stage of recognition. An extension to these methods is developed which considers the static, velocity and acceleration components separately. A series of confidence measures for the temporal derivatives is devised and applied within the soft-decoding framework. Experimental results on both a connected digit task and a large vocabulary task demonstrate significant increases in recognition accuracy under a range of packet loss conditions.

1. INTRODUCTION

The growth of mobile and handheld devices for speech communication has resulted in distributed speech recognition (DSR) systems being developed. The European Telecommunication Standards Institute (ETSI) Aurora DSR standard [1] offers good robustness to noise by replacing the low bit-rate speech codec on the terminal device with the static MFCC feature extraction component of the speech recogniser.

DSR is poised to become the principle technology for accessing speech enabled services on 3G mobile networks. These are often best effort packet-switched networks that do not guarantee reliable delivery. Packets may become corrupt due to low signal strength or dropped due to network congestion, resulting in portions of the feature vector stream becoming lost.

Work on packet loss compensation for DSR can be divided into three broad groups. The first group attempts to increase the probability of correctly receiving the feature vectors through source-coding techniques [1][2]. However, these methods require additional operations on the client device, which may not be possible as the Aurora DSR standard defines the functionality of the client and the payload format. The second set of techniques concentrate solely on the server side of the DSR system which is not defined by the standard. These schemes typically attempt to reconstruct the feature vector stream prior to recognition, using methods such as repetition [1], interpolation and statistical methods [3] and work reasonably well for short duration bursts of loss but degrade as burst lengths increase. The third category concentrates on compensating for lost vectors inside the recogniser itself using 'soft-decoding' [5,7] or its subset missing feature theory [4,6].

In the ETSI Aurora DSR standard, static feature vectors are received across the network and missing vectors are reconstructed using 'nearest-neighbour repetition' [1]. Temporal derivatives are then calculated from the reconstructed static vector stream using regression [8]. This means the loss of a single static feature vector will affect several temporal components. Therefore it is clear that the treatment of velocity and acceleration components should depend not only on the status of the current static vector, but also on those that surround it. However, most previous work on soft-decoding has treated the temporal components of the entire feature vector as a whole.

The aim of this work is to improve the accuracy of distributed speech recognition in the presence of packet loss through more effective compensation of the temporal derivatives. Section 2 examines the effect that packet loss has on the temporal derivatives of the feature vector stream. Soft decoding methods are reviewed in section 3 and an extension proposed for treating temporal derivative separately from the static component. Experimental results for the proposed methods are presented in section 4 and a conclusion made in section 5.

2. THE EFFECT OF PACKET LOSS ON TEMPORAL DERIVATIVES

When compensating for missing static vectors it is important to consider their effect on both the velocity and acceleration derivatives which will subsequently be included in the feature vector at the back-end. Temporal derivatives are computed using the regression formulae given in equations 1 and 2, where W_v and W_a are the number of vectors either side of the current frame used to calculate the velocity and acceleration components. In this work $W_v=3$ and $W_a=2$ and $\mathbf{x}_t^S$, $\mathbf{x}_t^V$ and $\mathbf{x}_t^A$ represent the static, velocity and acceleration derivatives at time t,

$$\mathbf{x}_t^V = \frac{\sum_{\phi=1}^{Wv} \phi(\mathbf{x}_{t+\phi}^S - \mathbf{x}_{t-\phi}^S)}{2\sum_{\phi=1}^{Wv} \phi^2} \quad (1)$$

$$\mathbf{x}_t^A = \frac{\sum_{\phi=1}^{Wa} \phi(\mathbf{x}_{t+\phi}^V - \mathbf{x}_{t-\phi}^V)}{2\sum_{\phi=1}^{Wa} \phi^2} \quad (2)$$

For illustration, figure 1 shows the static, velocity and acceleration values for MFCC(1) over a period of 50 frames.

The authors gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC) - GR/R88243/01

Two bursts of packet loss have been introduced; a single vector loss at frame 11 and an 8 vector loss starting at frame 31. The solid line shows the original loss-free coefficients while the dashed line shows the same coefficients but with repetition used to estimate the missing static vectors.

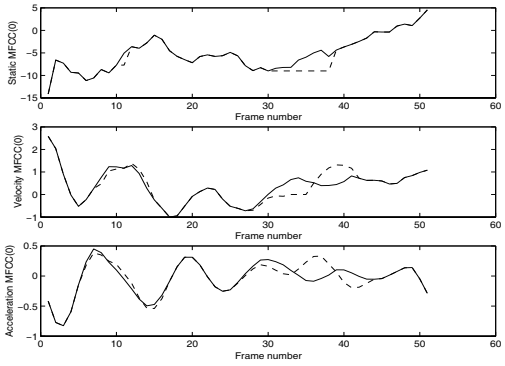

Figure 1: a)-static, b)-velocity and c)-acceleration of MFCC(1)

The figure clearly shows how distortion from the static features propagates into the velocity and acceleration derivatives and becomes worse as the burst length increases. In extreme cases, when the burst length exceeds 1 frame less than the window width used to the compute the derivate, the derivative will take a zero-valued result. This can be seen for the velocity derivatives for frames 33, 34 and 35. This distortion of the velocity component will also propagate to the acceleration component. In fact a burst of loss of b frames will affect $2W_v+b$ velocity components and $2(W_v+W_a)+b$ acceleration components. These results suggest that, as channel condition worsens, the temporal derivatives will become distorted more quickly than the static vector stream and have little, or even a negative effect, on recognition results.

3. SOFT DECODING FOR PACKET LOSS COMPENSATION

Missing feature theory [4,6], or 'hard decoding', is a technique whereby the Viterbi decoding stage of recognition is altered to account for missing vectors in the feature vector stream. The probability of observing the t^{th} feature vector, $\mathbf{x}_t$, in the j^{th} state of the HMM is given by $b_j(\mathbf{x}_t)$. In missing feature theory this is changed to,

$$b'_j(\mathbf{x}_t) = b_j(\mathbf{x}_t)^{\rho_t} \tag{3}$$

where $\rho_t = 1$ if the t^{th} feature vector is received, or $\rho_t = 0$ if it is not. Therefore, if $\rho_t = 0$ the value of $b'_j(\mathbf{x}_t)=1$ for all j, and the observation will have no influence on the path chosen by the Viterbi decoding algorithm. Instead the path will depend wholly on the prior information represented by the HMM state-transition matrix.

The above method assumes that if a vector is lost, no knowledge can be inferred about its value. However, due the temporal correlation of the feature vector stream this is not the case. Indeed, lost vectors can be replaced with estimates based on the surrounding received vectors with good results [3].

So called 'soft-decoding' [5,7] methods extend the Missing Feature Theory framework by using a two stage approach. First, the missing static vectors are reconstructed and the temporal derivatives calculated. Secondly, recognition is performed using the altered observation probability, $b'_j(\mathbf{x}_t)$, but allowing the parameter ρ_t to take on an arbitrary value ($0 \le \rho_t \le 1$). Here, ρ_t can be thought of as a measure of confidence for the estimate of the t^{th} feature vector, such that $\rho_t=1$ if the t^{th} vector is not affected by packet loss or $0 \le \rho_t < 1$ otherwise.

As shown in figure 1, if a static feature vector is received close to a burst of packet loss, it may be the case that the temporal derivatives associated with that vector will be affected by losses in neighbouring frames. Thus, the confidence measures for the static, velocity and acceleration components for each vector must be calculated separately. Therefore, assuming diagonal covariance, the observation probability can be divided into its separate components for static, velocity and acceleration.

$$b'_j(\mathbf{x}_t) = b_j^S(\mathbf{x}_t^S)^{\rho_t^S} \cdot b_j^V(\mathbf{x}_t^V)^{\rho_t^V} \cdot b_j^A(\mathbf{x}_t^A)^{\rho_t^A} \tag{4}$$

where $b_j^S(\mathbf{x}_t^S)$, $b_j^V(\mathbf{x}_t^V)$ and $b_j^A(\mathbf{x}_t^A)$ represent the static, velocity and acceleration components of the observation calculation. These are scaled by ρ_t^S, ρ_t^V and ρ_t^A which represent the confidence of the individual temporal components. The remainder of this section deals with how to best calculate these values.

3.1 Static component confidence

This section considers the calculation of confidence measures for the static feature vectors. Due to the correlation within the feature vector stream, vectors estimated close to the edge of a burst are more accurate than those estimated towards the middle. Thus, the value of ρ_t should vary, starting at the highest point (closest to 1) at the edges of the burst and dropping to a minimum at the mid-point of the burst. Various methods for varying ρ_t can be found in the literature [5]. Varying the confidence parameter linearly leads to the following calculation,

$$\rho_t^S = 1 - n\gamma_{linear} \tag{5}$$

where n is the number of frame indexes between the t^{th} vector and the closest edge of the burst of loss, i.e.,

$$n = \min(t - N_{before}, N_{after} - t) \tag{6}$$

where N_{before} and N_{after} are the frame indexes of the first correctly received vector before and after the burst of loss. The parameter γ_{linear} represents the rate that the confidence of the estimation falls off. A floor is applied to ensure that $\rho_t \ge 0$. An alternative approach is to vary the value of ρ_t exponentially,

$$\rho_t^S = \gamma_{exponential}^n \tag{7}$$

where $\gamma_{exponential}$ is a parameter representing the rate at which the confidence of the estimation decreases. The parameters γ_{linear} and $\gamma_{exponential}$ can be derived experimentally. Research by Cardenal-Lopez et. al. [5] has shown that optimum values are in the region of $\gamma_{linear}=0.1$ and $\gamma_{exponential}$=0.7, which is confirmed in this work.

3.2 Temporal derivative component confidence

As stated above, the confidence measures for the temporal derivatives should be calculated separately from those of the static vectors. As each temporal derivative component is calculated from a sliding-window on the next lower-order

temporal derivative, it is expected that the confidence measure for any component should be related to those within its sliding window. Therefore, the following methodology is proposed. First, the static confidence measures are calculated using those methods outlined in section 3.1. Secondly, the confidence measures for the velocity components are calculated as a function of the static confidence measures within the window,

$$\rho_t^V = f\left([\rho_{t-Wv}^S, \rho_{t-Wv+1}^S, \cdots, \rho_{t+Wv-1}^S, \rho_{t+Wv}^S]\right) \quad (8)$$

Similarly, the confidence measures for the acceleration components are computed from those of the velocity,

$$\rho_t^A = f\left([\rho_{t-Wa}^V, \rho_{t-Wa+1}^V, \cdots, \rho_{t+Wa-1}^V, \rho_{t+Wa}^V]\right) \quad (9)$$

The remainder of this section proposes three schemes to calculate the confidence of the temporal derivatives based on an exponential confidence measure applied to the static component.

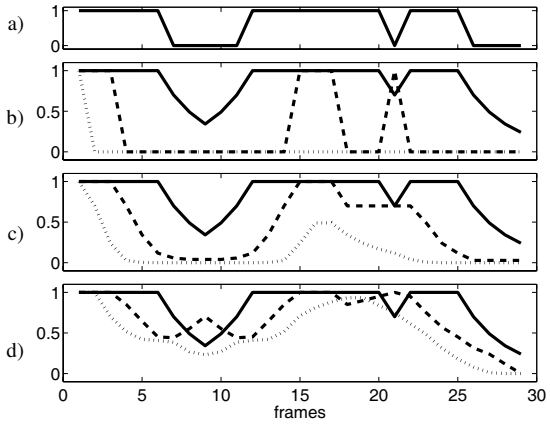

Figure 2: Temporal component confidence measures for a) hard decoding, b) product of measures and c) regression method.

3.2.1 Hard decoding of temporal derivatives

The hard decoding of temporal derivatives is the most severe method of computing the confidence measures. In the computation of the temporal derivatives, if any elements have been affected by packet loss then the confidence measure is set to zero. To illustrate this, figure 2a shows a simulated packet loss profile where 0 indicates loss and 1 no loss. Figure 2b shows the resulting confidence measure for the static (solid line), velocity (dashed line) and acceleration (dotted line) components using the hard decoding method. The figure shows that W_v velocity components either side of the loss are ignored in the decoding process. Similarly, W_v+W_a acceleration components are ignored either side of the loss. As the figure shows this scheme results in large numbers of temporal derivatives being removed from the decoding process. It is interesting to note that the confidence measure of the velocity component corresponding to an isolated static vector loss is 1. This is because the velocity measurement at time, *t*, does not consider the static value at time *t*.

3.2.2 Product of confidence measures method

This scheme calculates the confidence of the velocity derivative as the product of the static confidences under the velocity window. The confidence of the acceleration derivative is then computed as the product of the velocity confidences under the acceleration window, i.e.

$$\rho_t^V = \prod_{\phi=-Wv}^{Wv} \rho_{t+\phi}^S \qquad \rho_t^A = \prod_{\phi=-Wa}^{Wa} \rho_{t+\phi}^V \quad (10)$$

Figure 2c shows the resulting confidence profiles for the velocity and acceleration derivatives. These are not as severe as those with hard decoding and allow the temporal derivatives to have more impact in recognition.

3.2.3 Regression based method

In this method the confidence of the velocity component is obtained by applying the static component confidence measures to the regression equation used to compute velocity (eq. 1). Similarly the confidence of the acceleration is obtained by applying the confidence of the velocity components to the acceleration regression equation. The confidences are given as,

$$\rho_t^V = \frac{\sum_{\phi=1}^{Wv} \phi(\rho_{t+\phi}^S \rho_{t-\phi}^S)}{\sum_{\phi=1}^{Wv} \phi} \qquad \rho_t^A = \frac{\sum_{\phi=1}^{Wa} \phi(\rho_{t+\phi}^V \rho_{t-\phi}^V)}{\sum_{\phi=1}^{Wa} \phi} \quad (11)$$

Figure 2d shows the regression-based velocity and acceleration confidence measures.

4. EXPERIMENTAL RESULTS

This section examines the effect of applying the soft-decoding methods on two recognition tasks, a small vocabulary (connected digit) task and a large vocabulary task. The static feature vectors comprise MFCC(1-12) and a log-energy term resulting in a 13-dimensional vector. As per the ETSI Aurora standard, each packet carries two feature vectors (a single frame pair). If a vector is lost it is reconstructed using nearest neighbour repetition [1].

As shown in [3], two metrics are considered for characterising network channel conditions; the *packet loss rate*, α, and the *average burst length*, β. Four simulated channel conditions are used in these tests as shown in table 1. Although the higher values chosen for these parameters might be seen as excessively large, research has shown that packet loss conditions can be this severe, although only for short periods of time.

	Packet loss rate, α	*Av. burst length,* β
Channel A	10%	4 packets
Channel B	10%	20 packets
Channel C	50%	4 packets
Channel D	50%	20 packets

Table 1: Simulated channel conditions

4.1 Results on the ETSI Aurora connected digit database

Experiments are performed on the Aurora connected digit database [1]. Digits are modelled using 16-state, 3-mode HMMs, trained from the set of clean digits. The test set comprises 1001 noise-free digits strings.

4.1.1 Recognition with static components only

This section compares methods of estimating the confidence for the static component only, as described in section 3.1. In order to focus on the static components, these methods were compared using a recogniser that uses only static feature vectors and has a baseline accuracy of 96.6%. Results for the confidence measures described in section 3.1 over the four channel conditions are

shown in table 2. As a comparison, results using repetition only and hard decoding are shown.

The results show that the soft-decoding methods improve performance over that of hard-decoding by a significant margin. Although results using the soft-decoding methods are similar, exponential weighting generally outperforms linear weighting.

	A	*B*	*C*	*D*
NN Repetition only	94.2	89.7	82.6	62.5
Hard decoding	94.7	90.7	83.4	64.8
Linear soft-decoding	95.2	91.6	86.5	67.8
Exponential soft-decoding	95.3	91.4	86.8	68.2

Table 2: Static only soft-decoding using Aurora

4.1.2 Recognition with temporal derivatives

This section now uses the exponential variation of the static confidence measures of the previous section in combination with the three methods of computing confidences for the temporal derivative described in section 3.2. In the event of packet loss the missing static feature vectors are estimated by repetition before temporal derivatives are computed. Table 3 shows results for repetition only and then combined with the three temporal derivative confidence measures. Baseline accuracy for the 39-D feature vector with no packet loss is 99%.

	A	*B*	*C*	*D*
NN Repetition only	96.7	91.6	83.0	60.6
TD Hard decoding	97.6	93.6	88.3	70.7
TD Product	97.6	93.9	89.8	71.1
TD Regression	97.9	93.9	91.0	71.8

Table 3: Soft decoding with temporal derivatives using Aurora

Applying the confidence measures gives a substantial improvement in accuracy over the simple repetition-only compensation. Of the three temporal confidence measures the hard decoding gives the smallest increase in performance, suggesting that it removes too much information from the feature vector stream. The regression based calculation gives best performance which indicates the importance of considering the relative contribution of each element in the temporal derivative calculation.

4.2 Results on the WSJCAM0 large vocabulary database

This section broadens the evaluation of the packet loss compensation techniques to large vocabulary continuous speech recognition in the form of the 5000 word WSJCAM0 task. In this experiment a standard 5000 word closed bigram language model was used together with a set of 3-state, 20-mode monophone HMMs. Testing used a set of 100 utterances from the development set which gave a baseline word accuracy of 81% using temporal derivatives (39-dimensions) and 54.6% with static only (13-dimensions). Table 4 shows results for three configurations. *NN Repetition only* represents the case of no soft-decoding and recognition using temporal derivatives. *Static only soft-decode* shows results for recognition using only the static vectors and soft-decoding using exponential variation. *TD Regression* extends the soft-decoding method to include the temporal derivatives with their confidence measures calculated using regression – section 3.2.3.

	A	*B*	*C*	*D*
NN Repetition only	70.4	69.3	34.3	25.9
Static only soft-decode	49.5	47.7	28.1	22.0
TD Regression	75.7	72.7	47.3	38.9

Table 4: Soft decoding using WSJCAM0

These results follow a similar pattern to those presented for the Aurora digit database. Removing the temporal derivatives in the static-only configuration results in more severe degradation of performance than with the Aurora system, suggesting that the temporal derivatives are more important for the large vocabulary task than the smaller connected digit task. Using soft-decoding with temporal derivatives results in significantly improved accuracy in poor channel conditions.

5. CONCLUSIONS

This work has shown that packet loss has a wider effect on the temporal derivative components of the feature vector stream than on the static components. In severe packet loss this leads a to substantial distortion of the temporal derivatives. A novel extension to traditional soft-decoding techniques has been proposed whereby the temporal derivatives are decoded using confidence measures that take into account this widening effect. Three possible methods for calculating the confidence measures for the temporal derivative components have been considered. Of these methods the regression based method was shown to offer the greatest increase in accuracy. This method is based on the regression equation used to calculate the temporal derivatives and takes into account the varying contributions of the lower order temporal derivatives within the calculation window.

6. REFERENCES

[1] ESTI document - ES 202 050 – STQ: DSR – Extended advanced front-end feature extraction algorithm, 2003

[2] C.B. Boulis, M. Ostendorf, E.A. Riskin and S. Otterson, “Graceful degradation of speech recognition performance over packet-erasure networks”, IEEE Trans. On Speech and Audio Processing, vol. 10, no. 8, pp. 580-590, 2002.

[3] A.B. James, A. Gomez and B.P. Milner, “A Comparison of Packet Loss Compensation Methods and Interleaving for Speech Recognition in Burst-Like Packet Loss”. Proc. ICSLP, 2004.

[4] T. Endo, S. Kuroiwa and S. Nakamura, “Missing feature theory applied to robust speech recognition on IP networks”, Proc. Eurospeech, 2003.

[5] A. Cardenal-Lopez, L. Docio-Fernadez and C.Garcia-Mateo, “Soft Decoding Strategies for Distributed Speech Recognition over IP Networks”, Proc. ICASSP, 2004.

[6] M. Cooke, P.Green, L. Josifovski, A.Vizinho, “Robust Automatic Speech Recognition with Missing and Unreliable Acoustic Data”, Speech Communication, Volume 34, Issue 3, June 2001.

[7] J. Barker, L.Josifovski, M.Cooke, P.Green, “Soft Decisions in Missing Data Techniques for Robust Automatic Speech Recognition”, Proc ICSLP 2000

[8] B.A. Hanson and T.H. Applebaum, "Robust speaker-independent word features using static, dynamic and acceleration features", Proc. ICASSP, pp. 857-860, 1990.