INCREASED ROBUSTNESS AGAINST BIT ERRORS FOR DISTRIBUTED SPEECH RECOGNITION IN WIRELESS ENVIRONMENTS

Brian Delaney

Georgia Institute of Technology
School of Electrical and Computer Engineering
Multimedia Communications Lab
Atlanta, GA 30332

delaney@ece.gatech.edu

ABSTRACT

In distributed speech recognition, the speech features are computed on a mobile device, compressed, and sent over a network to a speech recognition server, where the Viterbi search and hidden Markov modeling takes place. In this work, we examine some error concealment methods for distributed speech recognition over burst error channels. We consider interpolation and interleaving, and we present a novel use of the stochastic weighted Viterbi recognition algorithm to increase robustness against interpolated features. We examine interleaving at both the frame level and codebook index level. Channel errors are simulated using a Gilbert model, and the performance of our algorithm is compared with other techniques, including the ETSI DSR standard, on a digits task and a large vocabulary task. Coupled with interleaving and interpolation, our algorithm can provide accuracy as high as 96.7% on a digit recognition task during an average bit error probability of $\frac{1}{20}$. On the more difficult WSJ task, the accuracy without bit errors is 85.7%. Using our algorithm, we can achieve 82.9% accuracy with an average bit error probability of $\frac{1}{30}$.

1. INTRODUCTION

The demand for tetherless access to data is driving the industry toward smaller but more capable wireless devices. The applications include high-quality wireless web browsing, multimedia e-mail and messaging services, digital music playback, as well as personal data management applications, such as calendar and contact databases. These pocket-sized devices have small screens and tiny keypads, so appropriate use of speech recognition technology can allow users to interact with the system in a natural manner. However, these devices are limited in computation, memory, and battery energy. Complex speech recognition tasks are difficult to perform on the device due to these resource limitations. A typical speech recognition system consists of a signal processing front-end or feature extraction step, followed by a search across acoustic and language models for the most likely sentence hypothesis. The signal processing front-end is a small portion of the overall computation and storage required. The acoustic and language models typically use on the order of tens of megabytes each of storage with significant computation required for large vocabulary search. Therefore, distributing the speech recognition across the network is an attractive alternative for these mobile wireless devices.

In distributed speech recognition (DSR), the speech features, typically mel-frequency cepstral coefficients (MFCC), are calculated at the client and sent over the wireless network to a server. Figure 1 shows a block diagram of this system. By only sending the speech data required for machine recognition, we can obtain better accuracy at lower bit rates than traditional human perception-based speech coders. The back-end speech recognition search including hidden Markov model (HMM) state output evaluation and Viterbi search is performed at the server. In order to minimize the bit rate, the MFCCs are first compressed using some quantization scheme. The result is a three-step process on the mobile client involving computation, quantization, and communication. The resulting text from the speech recognition process can be sent back to the mobile client or handled at the server depending on the nature of the application.

The presence of bit errors in the quantized speech feature stream can cause a significant decrease in accuracy. It is essential that bit errors be detected and concealed when possible. A study of frame erasure and errors with respect to accuracy was demonstrated in [1], which emphasizes the need for effective error detection, correction, or concealment techniques. In this work, we attempt to alleviate the effects of bit errors on the DSR bitstream through the use of interleaving, concealment through interpolation, and a novel use of the stochastic weighted Viterbi algorithm.

In Section 2, we discuss some related work. We overview the interpolation and interleaving methods briefly in Section 3, including a characterization of the interpolation error. This interpolation error is fed into a stochastic weighted Viterbi algorithm that is explained in Section 4. Recognition results are presented in Section 5, followed by conclusions and observations in Section 6.

2. RELATED WORK

In [2], the performance of DSR over IP networks is investigated. Packet losses, containing a single frame of speech, are simulated using random losses, a Gilbert-Elliot model, and a network bottleneck simulation. Repetition based error concealment is shown to be adequate for random isolated losses, but it breaks down under lengthy burst-like packet loss. Interleaving is used in [3] to distribute the speech features across multiple packets. This reduces the probability of the loss of a complete frame at the cost of increased delay. Interleaving, coupled with linear interpolation, minimized reductions in accuracy in moderate packet loss conditions. In [4], an interpolation technique in the log-filterbank domain is shown to be more robust to consecutive frame losses since log filterbank features exhibit a much higher temporal correlation.

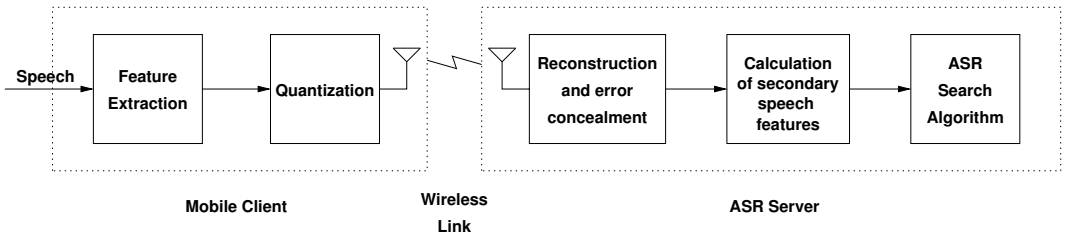

Fig. 1. A typical distributed speech recognition system.

In [1], errors were concealed by repetition, and the HMM output probability was weighted exponentially by the square root of the auto-correlation lag of the repeated feature. The result was that longer bursts of repeated features counted less toward the Viterbi search update. A stochastic version of the weighted Viterbi algorithm was presented in [5] in the context of speaker verification in noise. It is based on the expected value of the HMM output probability for noisy speech vectors.

The ETSI DSR standard provides a framework for MFCC calculation, quantization, framing and error protection for DSR applications [6]. We use the split vector quantization codebooks supplied with the ETSI DSR system, which include 7 codebooks of varying size. One potential problem with the ETSI DSR system is the use of cyclic redundancy check (CRC) error detection bits on consecutive frame pairs. It was shown in [7] that this use of error protection coupled with repetition can cause a single feature vector to be used for many consecutive frames in the presence of bit errors. We consider CRC protection bits applied only to individual frames as presented in [7]. The burst error channel is simulated using a two-state Gilbert-Elliot channel model at the bit level [8]. After CRC error detection, our algorithm sees bit errors as missing frames in the MFCC feature stream.

3. INTERLEAVING AND INTERPOLATION

Interpolation is used to conceal errors in the output feature vectors by filling the gap with data based on the good frames at either side of the gap. A frame error is defined as a failure of the 4-bit CRC in the received 48-bit frame. All codebook indices are considered corrupted and concealment is required. In [4], it was shown that interpolation in the log-filterbank domain works better as there is little temporal correlation in the cepstrum. This requires some zero-padding, followed by an inverse discrete cosine transform (DCT), interpolation, and finally a DCT to return to the cepstral domain. We performed cubic spline interpolation in the log-filterbank domain. Cubic spline interpolation results in a better representation of the missing data by requiring that the first and second derivatives be smooth and continuous, respectively.

In the presence of burst-like errors, many consecutive frames of speech can be corrupted. Interpolation techniques break down over longer burst lengths and can produce significant errors in the output. Interleaving is a common technique to reduce the chances of consecutive missing frames of data. By scrambling the order of the data frames, a burst-like error pattern will spread the loss across non-consecutive frames, allowing interpolation to be performed across a smaller gap. An $M \times N$ block interleaver rearranges the order of data frames for MN total frames. The de-interleaving operation is the inverse of the interleaver, and the input frames are restored to their original order.

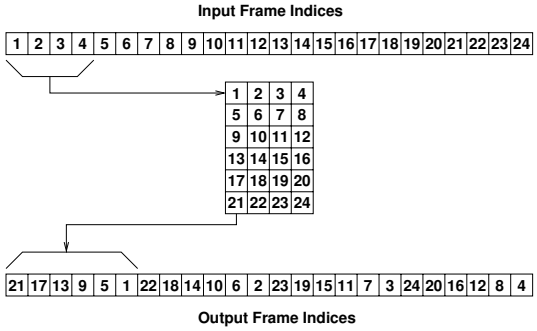

Fig. 2. A 6×4 frame-based block interleaver. Input frames are numbered sequentially.

In this work, we use a 6×4 frame-based interleaver and a 14×12 sub-frame interleaver, both of which preserve the 24-frame block size in the ETSI standard. The frame-based interleaver is shown in Figure 2. The input sequence consists of sequentially numbered frames, each consisting of 7 codebook indices. The output sequence is obtained by reading up each column. Each output frame is protected by a 4-bit CRC.

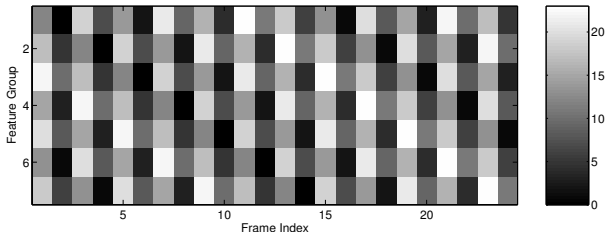

Fig. 3. The output of a 14×12 sub-frame interleaver. Input frame numbers are indicated by color, and each column represents a CRC protected block consisting of 7 codebook indices from various input frames.

In the sub-frame interleaver, the entire set of 168 codebook indices from the 24 frames of speech are interleaved using a 14×12 interleaver shown in Figure 3. These dimensions are chosen to ensure that each sequential group of 7 codebook indices at the output of the interleaver contains one value from each codebook. Each set of 7 codebook indices is then protected with a 4-bit CRC. Both interleavers have a delay of 24 frames, but each CRC protected frame at the output of the sub-frame interleaver contains codebook indices from 7 different speech frames. Therefore the loss of a single frame is spread across multiple output frames. Interpolation

can still be performed in the log-filterbank domain, but it must be performed for the cepstral coefficients associated with each codebook separately and the results combined.

3.1. Modeling the Interpolation Error

In this section, we model the error due to interpolation of bad or missing feature vectors. While the interpolation is reasonably accurate for gaps of a few frames, it breaks down with longer burst length. We consider various burst lengths in our analysis.

The interpolation error for a given feature can be modeled as zero-mean white Gaussian noise. For a given feature, n, at time, t, we have:

$$\hat{O}_{t,n} = O_{t,n} + v_{t,n} \tag{1}$$

where $\hat{O}_{t,n}$ is the interpolated feature, $O_{t,n}$ is the original speech feature (unknown at the receiver), and $v_{t,n}$ is the white Gaussian noise process. We calculate the sample mean and variance using test data, where errors of known burst length are systematically inserted and interpolation is performed. The result is a set of estimates for the mean and variance of the interpolation error for each feature at varying burst lengths. The mean was found to be very close to zero for all features and for all burst lengths tested. Figure 4 shows the distribution of the interpolation error for the 11th MFCC coefficient during a burst of length one.

4. WEIGHTED VITERBI ALGORITHM

Given a model of the additive noise for interpolated speech vectors, we can pass this uncertainty information to the Gaussian mixture density evaluation. The weighted Viterbi algorithm replaces the output probability calculation with its expected value. In the normal HMM output probability computation with diagonal covariance matrices, the output probability for HMM state k is calculated as follows:

$$b_k(\mathbf{O}_t) = \sum_{m=0}^{M-1} C_m \prod_{n=1}^{N-1} \frac{1}{\sqrt{(2\pi)\sigma^2_{m,k,n}}} \exp\left[-\frac{1}{2}\frac{(O_{t,n}-\mu_{m,k,n})^2}{\sigma^2_{m,k,n}}\right] \tag{2}$$

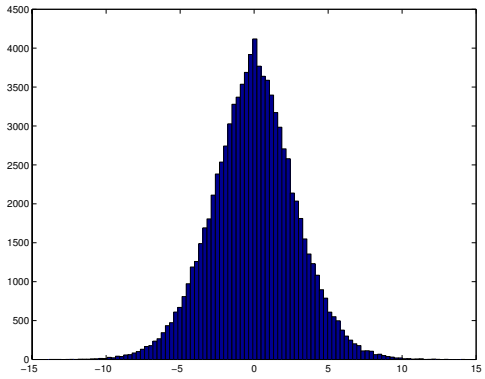

Fig. 4. The distribution of the interpolation error for $c(11)$ with a burst length of one frame.

where M is the number of Gaussian mixture densities, N is the number of features, and $\mu_{m,k,n}$ and $\mathbf{\Sigma}_{m,k,n}$ are the means and covariance matrix for state k, mixture m, and feature n. The expected value of the interpolated feature vector can be written as [5]:

$$E[b_k(\hat{\mathbf{O}}_t)] = \sum_{m=0}^{M-1} C_m \prod_{n=1}^{N-1} \frac{1}{\sqrt{(2\pi)\hat{\sigma}^2_{m,k,n}}} \exp\left[-\frac{1}{2}\frac{(E[\hat{O}_{t,n}]-\mu_{m,k,n})^2}{\hat{\sigma}^2_{m,k,n}}\right] \tag{3}$$

where $\hat{\sigma}^2_{m,k,n}$ and $E[\hat{O}_{t,n}]$ are the total variance and expected value of the corrupted feature in HMM state k, $\hat{O}_{t,n}$. Under the assumption of zero mean independent Gaussian noise, the variance, $\hat{\sigma}^2_{m,k,n}$, is:

$$\hat{\sigma}^2_{m,k,n} = \sigma^2_{m,k,n} + \tilde{\sigma}^2_{n,l} \tag{4}$$

where $\tilde{\sigma}^2_{n,l}$ is the interpolation error variance for feature n, at burst position l that was found in Section 3.1. The value of $E[\hat{O}_{t,n}]$ is:

$$E[\hat{O}_{t,n}] = E[O_{t,n}] + E[v_{t,n}] = O_{t,n} \tag{5}$$

since $E[v_{t,n}] = 0$ and $O_{t,n}$ is a constant for time instant t.

The algorithm works as follows. In the absence of any bit errors, no interpolation is performed, and the interpolation error variance, $\tilde{\sigma}^2_{n,l}$, is set to zero. In this case, the output probability is identical to (2). In the presence of bit errors, the burst length, in frames, is determined by delaying until an un-errored frame is received. Interpolation is performed, and the appropriate set of variances, $\tilde{\sigma}^2_{n,l}$, for the burst length and position within the burst for each feature are passed to the Gaussian evaluation. For large vocabulary tasks, we found that the adaptation worked best with a scaling factor of 0.3 applied to the variance, $\tilde{\sigma}^2_{n,l}$. When the variance of the interpolation error is high, the output probabilities of all states and all mixtures tend toward *zero*, and the discriminative ability of the model is decreased (i.e. larger ranges of input produce similar output probabilities.)

This differs from the algorithm presented in [1], where the output probability was weighted exponentially:

$$b_k(\mathbf{O_t}) = \prod_{k=1}^{N} [b(\mathbf{O}_{k,t})]^{\gamma_{k,t}} \tag{6}$$

The output probability of each individual feature, $b(\mathbf{O}_{k,t})$, is weighted according to the square root of the time-autocorrelation of the feature. Therfore $\gamma_{k,t} = \sqrt{\rho_k(t-t_c)}$, where $\rho_k(t-t_c)$ is the autocorrelation of feature k at lag index t_c. Secondary features were weighted according to a binary value (0 or 1) based on the length of the burst. We found that this weighting does not work well with interpolation as it does not capture the statistics of the interpolation error. In addition, the output probabilities with the exponential weighting tend toward *one* with increasing error burst length. When coupled with a language model and the associated language model weight, the acoustic model is given more emphasis in the Viterbi update equations during error bursts. The stochastic weighted Viterbi algorithm has the opposite effect, where the acoustic model is given less weight in relation to the language model during error bursts. This results in better performance for large vocabulary speech recognition.

5. RESULTS

We tested various error concealment schemes, including the weighted Viterbi recognition, using the TIDIGITS and WSJ speech corpus and the ISIP speech recognizer from Mississippi State. For the digits task, word models were trained using 941 utterances whose MFCC vectors were calculated using the ETSI DSR front-end. The test set consisted of 336 digit utterances of varying length and across different speakers. The WSJ journal system was trained on 1792 clean speech utterances and tested with 166 utterances and a bigram language model with a 5,000 word vocabulary. The burst-like error conditions were simulated using the two state Gilbert-Elliot model. The bit error patterns for a particular speech utterance were computed in advance and stored on disk. In this way, each concealment method was tested against the same bit errors for a given set of channel model parameters. We varied both the mean burst length in bits, $\overline{T_b}$, and the mean time between bursts, $\overline{T_g}$. We also show the average bit error rate (BER) for the burst channel conditions. The probability of bit error in the good state is fixed at 10^{-6}, and the probability of bit error in the bad state is fixed at 10^{-1}. The following concealment scenarios were tested:

A The ETSI DSR standard with no modifications

B Expontentially weighted Viterbi recognition from [1] with a a 6×4 frame-based interleaver

C Stochatistic weighted Viterbi recognition with a 6×4 frame-based interleaver

D Stochatistic weighted Viterbi recognition with a 14×12 sub-frame interleaver

Each system adds an increasing amount of error concealment. The bit rate of **A** is 4.8kbps, while the bit rate of **B**, **C**, and **D** is 5.0 kbps due to the single frame based CRC error detection.

The results are shown in Table 1. Word accuracy (including substitutions, deletions, and insertions) is reported. The baseline accuracy without bit errors is 99.5% for the digits task and 85.7% for the WSJ task. From the table, we can see that the ETSI system (column **A**) does not provide adequate performance in the more severe error conditions that we have simulated. Accuracy quickly drops below 90% in the digits task and barely reaches 50% in the WSJ task. Interleaving offers the biggest improvement in accuracy. Stochastic weighted Viterbi recognition (column **C**) is able to outperform the exponential version (column **B**) in all channel conditions tested. The improvement in the WSJ task is more pronounced due to the interaction between language model probabilities and acoustic model probabilities with the respective weighting techniques. Finally, the addition of sub-frame interleaving (column **D**) offers additional improvement under certain channel conditions. The risk with sub-frame interleaving is that a single bit error can be spread across 7 frames of speech, while frame-based interleaving isolates the error to a single frame. This may explain why sub-frame interleaving performs slightly worse in some conditions and better in others.

Table 1. Results of DSR simulations for TIDIGITS and WSJ Tasks.

TIDIGITS Task					
$\overline{T_g}/\overline{T_b}$	Avg. BER	A	B	C	D
500/200	2.86×10^{-2}	91.00	98.90	99.20	98.90
200/100	3.33×10^{-2}	89.70	98.40	98.80	98.60
200/200	5.00×10^{-2}	72.50	95.90	96.80	97.60
500/500	5.00×10^{-2}	71.00	94.20	94.60	95.20
WSJ Task					
$\overline{T_g}/\overline{T_b}$	Avg. BER	A	B	C	D
500/200	2.86×10^{-2}	50.50	81.20	82.70	82.90
200/100	3.33×10^{-2}	41.10	78.90	82.20	82.90
200/200	5.00×10^{-2}	16.80	68.00	70.30	73.00
500/500	5.00×10^{-2}	18.20	66.50	68.10	68.00

6. CONCLUSION

In this paper, we investigated the effects of several well-known error concealment algorithms for DSR traffic over a burst error channel. We also presented a novel use of the stochastic weighted Viterbi algorithm and sub-frame interleaving to increase robustness in the presence of bit errors. This algorithm can provide accuracy as high as 96.7% in burst error conditions with an average BER of $\frac{1}{20}$ on a digit recognition task. In a more difficult WSJ task, an accuracy of almost 83% can be maintained with an average BER of $\frac{1}{30}$. These algorithms are able to outperform both the ETSI standard [6] and the exponential weighted Viterbi recognition technique presented in [1].

7. REFERENCES

[1] Alexis Bernard and Abeer Alwan, "Low-bitrate distributed speech recognition for packet-based wireless communication," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 8, pp. 570–579, November 2002.

[2] D. Quercia, L. Docio-Fernandez, C. Garcio-Mateo, L. Farinetti, and J.C. De Martin, "Performance analysis of distributed speech recognition over ip networks on the aurora database," in *ICASSP 2002*, 2002.

[3] Pedro Mayorga, Richard Lamy, and Laurent Besacier, "Recovering of packet loss for distributed speech recognition," in *EUSIPCO 2002*, 2002.

[4] Ben Milner, "Robust speech recognition in burst-like packet loss," in *ICASSP 2001*, 2001.

[5] N.B. Yoma and M. Villar, "Speaker verification in noise using a stochastic version of the weighted viterbi algorithm," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 3, pp. 158–166, 2002.

[6] Various, "Speech processing, transmission and quality aspects (stq); distributed speech recognition; front-end feature extraction algorithm; compression algorithms," ETSI Standard: ETSI ES 201 108 v1.1.2, 2000, `http://www.etsi.org`.

[7] Zheng-Hua Tan and Paul Dalsgaard, "Channel error protection scheme for distributed speech recognition," in *ICLSP '02*, 2002.

[8] E. N. Gilbert, "Capacity of a burst noise channel," *Bell Syst. Tech. Journal*, pp. 1253–1265, 1960.