

Distributed ASR Using Speech Coder Data for Efficient Feature Vector Representation

Trond Skogstad and Torbjørn Svendsen

Department of Electronics and Telecommunication, NTNU
O.S. Bragstads plass N-7491, Norway
{tronds, torbjorn}@iet.ntnu.no

Abstract

This paper proposes an alternative approach to distributed speech recognition in scenarios where both reliable feature vectors and reconstruction of the speech signal are required. By transmitting the difference between speech coded information and the desired feature vectors, this system achieves both excellent quality speech reconstruction and ASR recognition performance. Experiments show that a transparent recognition rate is achieved with as little as 0.6 kbps of additional information supplementing the AMR speech coder operating at 4.75 kbps. The total rate is comparable to the the ETSI 202 211 extended front-end standard.

1. Introduction

The transmission of speech data over noisy communication channels can lead to a degradation in the performance of automatic speech recognition (ASR) systems. This is due to two separate effects: 1) Information reducing operations done in the source coder and 2) Transmission errors.

Linear predictive coding (LPC) and related analysis-synthesis algorithms are the prevailing methods for low to medium rate compression of speech. These coders seek to model the speech signal as a time-varying all-pole filter, obtained through linear predictive analysis, and an excitation/residual signal. The specifics of the excitation signal representation differ from codec to codec, but it is invariably a coarse approximation to the unquantized residual. This, combined with quantization of the filter parameters leads to spectral distortion detrimental to recognition performance [1],[2].

The distributed speech recognition approach sidesteps the speech coder by extracting the features used for recognition in the terminal and transmitting them to a back-end recognizer over an error protected data channel. The ETSI Aurora group has developed several standards in this field, including a mel-cepstrum front-end [3] and a front-end with improved robustness to background noise. The latest developments are extensions to these standards, allowing speech reconstruction at the back-end. [4] lists potential applications such as speech recognition of "sensitive information", e.g. banking and brokerage transactions where human verification may be necessary, and mixed human/machine recognition (human assisted dictation).

An alternative approach would be to base the coding of the front end feature vectors on the speech coded data, coding and transmitting only the difference between the speech coder information and the desired front end feature vectors. This way the reconstructed speech will have the excellent quality attained by state-of-the-art speech coders while the ASR system sustain no loss in recognition rate. One set of trained models suffice, even

in scenarios where the ASR system is processing speech data that have undergone different kinds of speech coding, as the feature vectors can be made compliant with any chosen front-end extraction standard.

This work explores two different ways of predicting the feature vector from the speech coded data and shows that the prediction error can be coded at a low rate. One approach investigated is to use the autoregressive spectral estimate given by the LP analysis as an approximation to the FFT spectral estimate used in MFCC calculations, the other to use MFCCs calculated from the reconstructed speech signal to predict the MFCCs of the original signal. In our work we use the ETSI ES 201 208 front-end and the 3GPP adaptive multi-rate speech codec at two different modes, but the principles are applicable to most front-end configurations and speech codecs.

2. ETSI mel-cepstrum front-end

ETSI ES 201 208 specifies a front-end feature extraction algorithm and compression algorithms for distributed speech recognition. The features used are traditional mel-frequency cepstral coefficients and the log-energy of the signal, generated in the following way [4]: After analog-to-digital conversion, a notch filter is used to remove the DC offset. Every 10 ms the signal is framed and the log-energy is calculated. A pre-emphasis filter is applied to each analysis frame before it is multiplied with a Hamming window. For each frame, a FFT-based magnitude spectrum is computed and passed through a 23 channel filterbank. The filters are triangular-shaped and half-overlapping, with center frequencies equidistant in the mel frequency domain. The cepstral coefficients are then found as the DCT-II of the natural logarithm of the filter outputs. The final feature vector consist of the 13 first cepstral coefficients (C_0, C_1, C_{12}) and the log-energy.

Split vector quantization is employed for signal compression. The feature vector components are grouped in pairs and quantized at the rates stated in table 1, yielding a source rate of 4.4 kbps. With header fields and some channel coding in the form of CRC bits the total data rate is 4.8 kbps.

ETSI ES 202 211 extends this with pitch- and voicing class information, enabling enhanced tonal language recognition and making speech waveform reconstruction a possibility. Operating at a rate of 5.6 kbps, the reconstructed speech is reported to be of the same or better quality as the US DoD MELP coder at 2.4 kbps [5].

Table 1: *The sub-vectors used by ETSI 201 208.*

| Feature pair | Bits/frame |
|------------------|------------|
| $\ln E, C_0$ | 8 |
| C_1, C_2 | 6 |
| C_3, C_4 | 6 |
| C_5, C_6 | 6 |
| C_7, C_8 | 6 |
| C_9, C_{10} | 6 |
| C_{11}, C_{12} | 6 |

3. 3GPP Adaptive Multi-Rate speech coder

The adaptive multi-rate speech coder [6] is in the class of Algebraic Code Excited Linear Prediction (ACELP) coders. The excitation signal is represented as a gain-weighted sum of two components - an entry from the adaptive codebook, representing information relating to pitch and an entry from the algebraic codebook representing the frame innovation. The coder has 8 different modes, operating at bit-rates of 4.75-12.2 kbps.

The coder operates on speech frames of 20 ms, each divided into 4 subframes of 5 ms. In the 12.2 kbps mode, 10th order Linear prediction (LP) analysis is performed twice per frame, using asymmetric windows with weights concentrated at the second and fourth subframe. For all other modes, one 10th order LP analysis is carried out using a window with its weight concentrated at the fourth subframe. All windows are 30 ms long, with a tail into the previous frame. A 60 hz bandwidth expansion is employed in the LP analysis, broadening the bandwidths of the spectral peaks. Before quantization, the LP filter coefficients are converted to the line spectral pair (LSP) representation. Interpolation is performed in the LSP domain, yielding one set of coefficients per subframe.

4. Feature Prediction

We want to express the cepstral coefficients of the uncoded speech signal, C_i , as the sum

$$C_i = \hat{C}_i + \varepsilon_i \quad (1)$$

where \hat{C}_i are cepstral coefficients based on the speech coded data and ε_i the prediction error. If quantized information from the speech coder is used, and supposing ε_i is not quantized, C_i can be reconstructed without information loss. Two different methods of finding the cepstral estimates \hat{C}_i are presented, one (section 4.1) using parameters from the speech coder, the other (section 4.2) using reconstructed speech. As the residual signal is only used to find filter gains, the method in 4.1 relies heavily on the ability of the spectral envelope found by LP analysis to model the more volatile FFT magnitude spectrum. Reconstructing the speech signal comes at a higher computational cost, but this method may utilize spectral information from both the LP filters and the residual.

4.1. Feature prediction based on LP spectral estimates

Under the assumption of a white residual spectrum, the autoregressive magnitude spectrum given by the LP filters from the speech coder, subject to the pre-emphasis filter $H(e^{j\omega})$ from the ETSI 201 108 standard,

$$\hat{X}(\omega) = \frac{G|H(e^{j\omega})|}{|A(e^{j\omega})|} \quad (2)$$

can be used as an estimate of the magnitude spectrum of the speech signal used in MFCC calculations,

$$X(\omega) = \left| \sum_{n=0}^{N-1} s_w(n)e^{-j\omega n} \right| \quad (3)$$

where $s_w(n)$ is the preprocessed speech signal. As a first step in finding ε_i , $\hat{X}(\omega)$ and $X(\omega)$ are passed through mel filter banks. The filters compute the average spectrum around a center frequency and thereby to some extent counteract the fact that $X(\omega)$ is more rapidly varying than $\hat{X}(\omega)$. Due to the mel frequency scale, more samples are averaged at high frequencies, leading to better prediction. With f_i and \hat{f}_i denoting the output of the i -th filter bank channel,

$$d_i = \ln(\hat{f}_i) - \ln(f_i), i = 1, \dots, 23 \quad (4)$$

are calculated. The prediction errors are then found as the DCT-II of the differences:

$$\varepsilon_i = \sum_{j=1}^{23} d_j \cos\left(\frac{\pi i}{23}(j - 0.5)\right), 0 \leq i \leq 12 \quad (5)$$

As the DCT is a linear transform, this is, of course, equivalent to calculating C_i based on eq. (3), \hat{C}_i based on eq. (2) and finding the difference $\varepsilon_i = C_i - \hat{C}_i$. The gain G is chosen as the square root of the energy in the quantized residual. Since LP analysis tends to overestimate the spectrum in valleys [7], \hat{f}_i is typically an overestimate of f_i , leading to a bias in d_i . Alternatively, G could have been chosen such that

$$\int X(\omega)d\omega = \int \hat{X}(\omega)d\omega, \quad (6)$$

but that would mean transmitting additional information. However, as we are interested in the logarithm of the filter bank outputs (implying that the multiplicative gain can be separated into an additive logarithmic term), the filters are of equal area and the cosine basis vectors used in the DCT-II have zero mean for $i \neq 0$, the gain only affects ε_0 . If the effects of the overestimation is somewhat systematic between frames, a suboptimal choice of the gain parameter would mainly lead to a shift in the mean value of C_0 rather than increase the variance, which has little effect on coding.

The log energy parameter and C_0 convey more or less the same information. An alternative to reconstructing the speech signal at the encoder is thus to estimate the log energy from C_0 . Least squares linear regression on a large set of speech data yields the line

$$\ln E = 0.098C_0 - 2.25 \quad (7)$$

to be the best fit. No substantial gain is achieved by using a higher order polynomial.

Some of the differences in the signal flow of the AMR speech coder and ETSI 201 208 are not counteracted. We use the LP filter parameters from the best sub frame in the AMR coder to estimate the spectrum, but do not attempt to interpolate between sub frames to compensate for the difference in analysis window length and shape between the two standards. Since low order LP analysis is not able to model spectral zeros, the difference between the sharp high pass filters used to remove DC-components are ignored. The bandwidth expansion in the AMR coder is also disregarded. To find the effect of this mismatch, the results using the LP analysis in the AMR speech coder are compared with an idealized LP analysis using the same preprocessing as the ETSI 201 208 standard.

4.2. Feature prediction based on reconstructed speech

In this approach the estimates \hat{C}_i are calculated by applying the ETSI 201 208 standard directly on the reconstructed (encoded/decoded) speech. If the low order LP analysis used in AMR speech coding is not entirely capable of capturing the spectral shape of the speech signal, the residual signal contains significant spectral information and this approach should produce better results than the other method. In the experiments presented later, the coder is operated at both the maximum bit rate of 12.2 kbps and the minimum rate of 4.75 kbps, to determine the effects of the internal quantization in the AMR coder.

4.3. Prediction gain

Following [8], we define the prediction gain for the i -th cepstral coefficient as the ratio

$$G_{pi} = \frac{\sigma_{C_i}^2}{\sigma_{\varepsilon_i}^2} \quad (8)$$

where $\sigma_{C_i}^2$ is the variance of C_i and $\sigma_{\varepsilon_i}^2$ the variance of the prediction error ε_i . The prediction gain of the log-energy is defined similarly. If C_i and ε_i are described by PDFs of similar shape, the prediction gain can be seen to represent the SNR improvement in going from direct quantization of the cepstral coefficients to quantization of the prediction errors. Further, under the same assumption, this SNR improvement translates to a possible bit rate reduction of

$$\Delta R_i = \frac{10 \log_{10}(G_{pi})}{6} \quad (9)$$

bits/sample.

4.4. Speech data

The speech data used for gain calculations and VQ design was 109 seconds of speech from the set of phonetically rich sentences in the Norwegian SpeechDat [9]. Sentences from 6 different speakers of diverse age were used, 3 male and 3 female. The sound clips were manually edited to remove silent parts leading and trailing the sentence. While the data set used is not big, the results presented should generalize well. The gains calculated are probably conservative estimates, as increased speaker and language variation should cause a larger increase in $\sigma_{C_i}^2$ than in $\sigma_{\varepsilon_i}^2$, due to the differential nature of the latter.

Table 2 states the estimated bit rate reduction ΔR_i [bits/sample] for the different features and prediction schemes as calculated from eq. (9) on the speech data. The last row of the table express the total reduction as the sum of each column. This is only achievable if the feature prediction gains are independent. This is probably not strictly true, but since the cepstral coefficients are decorrelated by means of the DCT, it should be a good approximation.

Contrasting the second column of table 2, stating the ΔR_i s for the method described in section 4.1, with the ΔR_i s in the two last columns, representing the method in section 4.2, we see that the second method it estimated to yield a total rate reduction more than twice as high. While the prediction gains for the log energy and the first cepstral coefficients are acceptable, there is almost no gain for the high order cepstral coefficients.

The rate reductions for idealized (in the sense that the ETSI 201 208 rather than the AMR preprocessing is used) 10th and 16th order LP analysis are stated in the third and fourth column respectively. Comparing column 2 and 3 we see that the mismatch in preprocessing incurs a loss of 2 bits. By using 16th

Table 2: Estimates of bit rate reductions for different features and coding modes. AMR-LP, I-LP10 and I-LP16 denotes the predictions based on spectral estimates of the AMR speech coder and an idealized 10th and 16th order LP analysis respectively. MR475 and MR122 denotes prediction based on reconstructed speech from the AMR coder operating at 4.75 kbps and 12.2 kbps.

| Feature | AMR-LP | I-LP10 | I-LP16 | MR475 | MR122 |
|-------------------|--------|--------|--------|-------|-------|
| $\ln(E)$ | 1.59 | 1.59 | 1.59 | 2.10 | 2.66 |
| C_0 | 2.38 | 3.32 | 3.37 | 4.15 | 4.27 |
| C_1 | 2.18 | 2.47 | 2.57 | 3.12 | 3.33 |
| C_2 | 1.92 | 2.15 | 2.42 | 2.85 | 3.02 |
| C_3 | 1.41 | 1.49 | 1.91 | 2.71 | 2.80 |
| C_4 | 1.53 | 1.61 | 2.06 | 2.63 | 2.61 |
| C_5 | 0.89 | 0.95 | 1.40 | 2.40 | 2.50 |
| C_6 | 0.83 | 0.89 | 1.34 | 2.08 | 2.12 |
| C_7 | 0.70 | 0.77 | 1.15 | 2.12 | 2.14 |
| C_8 | 0.49 | 0.56 | 0.91 | 1.92 | 1.99 |
| C_9 | 0.35 | 0.49 | 0.85 | 1.93 | 1.95 |
| C_{10} | 0.32 | 0.47 | 0.81 | 2.08 | 2.06 |
| C_{11} | 0.24 | 0.36 | 0.59 | 1.75 | 1.69 |
| C_{12} | 0.20 | 0.30 | 0.45 | 1.71 | 1.70 |
| $\sum \Delta R_i$ | 15.4 | 17.4 | 21.4 | 33.5 | 34.8 |

instead of 10th order LP analysis, 4 bits in total rate reduction is possible. This significant improvement is a clear indication that low order LP analysis as used in the AMR coder is not able to represent the spectral subtleties needed in the MFCC calculations.

The last two columns show the total possible rate reductions for the two modes of the AMR coder tested, 4.75 kbps and 12.2 kbps respectively. The difference is only 1.3 bits, which is quite remarkable considering that the bit rate used by the speech coder is more than halved. Multiplying the theoretical rate reductions by a frame rate of 100 frames/sec. and comparing to the source rate of 4.4 kbps of the ETSI 201 108 front end, we see that very little additional information is needed.

The prediction gains G_{pi} corresponding to the rate reductions stated in the two last columns can also be interpreted as the SNR in each feature, where the noise is introduced by the speech coding.

5. Recognition experiments

The difference in prediction gain between the two methods tested was substantial. Only the most promising prediction method, based on reconstructed speech, was used for recognition experiments. The prediction errors ε_i were encoded with vector quantizers designed using the Linde-Buzo-Gray algorithm [10] on the training set described in sec. 4.4. Ideally the distortion measure used in the training process should reflect that different elements of the feature vector are of different importance to recognition performance as well as unequally impaired by the speech coding. Since finding such a measure is far from trivial, conventional MSE is used herein. In some experiments we use a separate VQ for $\ln E$ and ε_0 , as these features have higher variance than the other features and might unjustly dominate the distortion calculations in the VQ design process [12]. The code books used are of the same size or smaller than the code books used in ETSI 201 208.

The experiments were carried out on the Norwegian

SpeechDat [9] corpus using a HTK 3.2 speech recognizer based on an ETSI 201 208 compliant front-end. All 13 cepstral coefficients as well as first and second order deltas were used. The $\ln(E)$ parameter was included in feature vector quantization, but not used in recognition. One set of monophonic acoustic models with 32 Gaussian mixtures were trained using clean (not speech coded) features, and used in all recognition experiments. The training script was mainly based on [11]. Testing were performed with a vocabulary of 2986 words, taken from the set of Norwegian city names and the set of phonetically rich words.

Table 3: *Recognition results. Uncoded is the baseline with no speech coding. When the additional information is expressed as a sum, the first addend is the bits used for $\ln E$ and C_0 , the second the bits used for the remaining cepstral coefficients. Otherwise, only one code book is used.*

| Coding Schemes | WER |
|----------------|------|
| Uncoded | 30.7 |
| MR475, 2+4 | 30.7 |
| MR475, 8 | 31.0 |
| MR475, 2+3 | 31.0 |
| MR122 | 31.3 |
| MR475, 6 | 32.0 |
| MR475 | 32.6 |

As speech coding at 12.2 kbps gave an increase in word error rate of only 0.6% compared to the uncoded case, all other test were performed at 4.75 kbps. When 4.75 kbps are used without additional information, the increase in word error rate is 1.9%. We see that transparent (no loss) performance is achieved when six (2+4) bits of additional information are transmitted. Multiplied by the frame rate and added to the 4.75 kbps used by the speech coder this is 5.35 kbps, comparable to the source rate of ETSI 202 211 (the extended front-end) of 5.2 kbps.

The systems operating with two code books perform better than systems using only one. This underlines the need for a distortion measure related to feature importance.

6. Conclusions and future work

Two different methods for feature vector prediction were examined. Prediction from LP filter parameters gave disappointing results, as the filter parameters alone did not contain enough information about the signal spectrum for efficient coding. The results were more encouraging for the other approach, prediction from reconstructed speech. High prediction gains were achieved and the recognition experiments showed that it is possible to obtain transparent performance at a low rate. At similar bit rates, this technique yields the same recognition performance and superior quality in the reconstructed speech compared to the ETSI 202 211 standard. One reason this is possible is that the latter does not compress the feature vectors as much as feasible. This implicit redundancy can of course be used in error resilience schemes to reduce the impact of transmission errors, but it is not necessary the best way of doing channel coding.

The techniques proposed in this work are probably more suited in tandem with speech coders operating at even lower bit rates, as recognition experiments on AMR speech coded data without additional information gave surprisingly good results. For applications where recognition performance is not of utmost concern, direct recognition on AMR speech coded data is

a decent solution.

The computational demands of the proposed system are quite high. An interesting avenue for future research would be to incorporate the spectral information from the residual signal into eq. (2), eliminating the need for coder side reconstructing of the speech signal. The design of vector quantizers using distortion measures related to feature importance is another interesting topic.

7. References

- [1] Lilly, B. T. and Paliwal, K. K., "Effect of Speech Coders on Speech Recognition Performance", Proc. ICSLP 1996.
- [2] Raj, B., Migdal, J. and Singh, R., "Distributed Speech Recognition with Codec Parameters", Proc. ASRU 2001.
- [3] ETSI(2003), "Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms" European Telecommunications Standards Institute, Document ETSI ES201 208 v1.1.3 (2003-09), <http://www.etsi.org>.
- [4] ETSI(2003), "Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Extended front-end feature extraction algorithm; Front-end feature extraction algorithm; Compression algorithms" European Telecommunications Standards Institute, Document ETSI ES202 211 v1.1.1 (2003-11), <http://www.etsi.org>.
- [5] Ramabadra, T., Sorin, A., McLaughlin, M., Chazan, D., Pearce, D., and Hoory, R., "The ETSI Extended Distributed Speech Recognition (DSR) Standards: Server-Side Speech Reconstruction", Proc. ICASSP 2004.
- [6] 3GPP(2004), "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Mandatory Speech Codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec; Transcoding functions (Release 6), Document 3GPP TS 26.090 V6.0.0 (2004-12), <http://www.3gpp.org>
- [7] T. F. Quatieri, Discrete-Time Speech Signal Processing: Principles and Practice. Upper Saddle River, NJ: Prentice Hall, 2002.
- [8] N. S. Jayant and P. Noll, Digital Coding of Waveforms: Principles and Applications to Speech and Video, Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [9] Johansen, F.T. and Amdal, I., "SpeechDat, Norwegian Database for the Fixed Telephone Network", 1997, <http://www.telenor.no/fou/prosjekter/taletek/speechdat/>
- [10] Linde, Y., Buzo, A. and Gray, R. M., "An Algorithm for Vector Quantizer Design", IEEE Transactions on Communication, Vol. Com-28, No. 1, Jan 1980.
- [11] Lindberg, B., Johansen, F.T., Warakagoda, N., Lehtinen, G., Kacic, Z., Zgank, A., Elenius, K., Salvi, G., "A Noise Robust Multilingual Reference Recogniser Based on Speechdag(II), <http://www.telenor.no/fou/prosjekter/taletek/refrec/>.
- [12] Srinivasamurthy, N., Ortega, A. and Narayanan, S., "Enhanced Standard Compliant Distributed Speech Recognition (Aurora Encoder) Using Rate Allocation", Proc. ICASSP 2004.