Proceedings of 20th International Congress on Acoustics, ICA 2010

23-27 August 2010, Sydney, Australia

An HMM-based segment quantizer and its application to low bit rate speech coding

Motoyuki Suzuki(1), Masashi Adachi(2), Minoru Kohata(3), Akinori Ito(4), Shozo Makino(5) and Fuji Ren(1)

(1) Institute of Science and Technology, The University of Tokushima, Tokushima, Japan
(2) Graduate School of Advanced Technology and Science, The University of Tokushima, Tokushima, Japan
(3) Faculty of Information and Computer Science, Chiba Institute of Technology, Chiba, Japan
(4) Graduate School of Engineering, Tohoku University, Sendai, Japan
(5) Tohoku Bunka Gakuen University, Sendai, Japan

PACS: 43.72.Gy

ABSTRACT

Several speech coding systems employ a segment quantizer instead of a vector quantizer. One of the most important problems is how to construct a segment codebook. In this paper, a new speech coder based on the ML-BEATS is proposed. The ML-BEATS is one of the HMM-based segment quantizer. First, it splits a vector sequence into several sub-sequences, and then these sub-sequences are clustered in order to construct a codebook. Each cluster center is represented by a left-to-right HMM. In the encoding process, input speech is matched with HMMs in the codebook, and then HMM index and duration information are sent to the decoder. In the decoding process, a decoded sequence is generated from HMM parameters by applying the HMM-based speech synthesis method. From the experimental results, the HMM-based speech coder gave 1.13 dB spectral distortion with 5.83 bit/frame. It is 0.11 dB higher spectral distortion than that given by G.729 coder, but bit rate decreased only 32%. In order to consider a shifting problem of LSP dimensions, we also propose a new codebook construction method. Many training vectors are extracted from training samples by shifting dimensions, and all vectors are used for constructing a universal codebook. The universal codebook can deal with any shifted vectors because all possibilities are included in the training data. From the experimental results, the shifted vector method encoded an input speech with very low bit rate, but it gave higher spectral distortions.

INTRODUCTION

Many speech coding systems employ a vector quantizer in order to encode a sequence of speech feature vectors. Increasing a size of codebook brings lower quantization distortion, but it also brings higher bit rate. In order to decrease bit rate without high distortion, several speech coder[1-3] employ a segment quantizer instead of a vector quantizer. It can encode a sequence of feature vectors efficiently by using temporal correlation between vectors.

One of the most important problems is how to split a sequence into segments. Splitting a sequence by fixed length is the simplest method, but it does not give high performance because a definition of segment is not decided by quantization efficiency point of view. If a sequence consists of a combination of a few kinds of segment, high performance (low spectral distortion with low bit rate) segment quantizer can be constructed because only a few kinds of segment should be registered into a codebook. In other words, each segment in a codebook should be corresponded to something chunk which is repeatedly appeared in a sequence. These chunks do not have a fixed length.

The LZSQ method[4] can automatically acquire variable length segments which are frequently appeared in training samples. However, this method cannot acquire optimum segmentation because it determines segment boundaries one after another. In this paper, a new segment quantizer is proposed. It represents a sequence of speech feature vectors by many HMMs, and a codebook consists of the HMMs. Both the HMMs and boundaries of sequences are appropriately determined by using a maximum likelihood criterion.

ML-BEATS

The ML-BEATS (Maximum Likelihood Boundary Estimation Algorithm for Time Sequences) has been proposed[5, 6] to find a new acoustic unit for speech recognition. Traditional speech recognition systems employ “phoneme” as a unit of acoustic models. However, nobody knows whether a phoneme-based acoustic model is optimum for speech recognizer, or not. The ML-BEATS can find an appropriate unit by using maximum likelihood criterion, and it showed higher performance than a traditional phoneme-based speech recognizer[5, 6].

The ML-BEATS can be used as a clustering algorithm for sequences. It splits a sequence into sub-sequences appropriately, and sub-sequences are clustered into several clusters. Each sub-sequence has various length, and a cluster center is represented by an HMM. The ML-BEATS carries out following two steps repeatedly, and segmentation of input sequences and clustering can be optimized simultaneously.

- ✓ Split input sequences into sub-sequences by using "current" cluster center (HMMs). All HMMs are concatenated in parallel, and transitions from a last state in any HMMs to a start state in any HMMs are added. After that, the Viterbi algorithm is carried out in order to find a correspondence between each frame and state. Finally, input sequences are split into sub-sequences each of which corresponds to an HMM.
- ✓ Split a cluster into two clusters and update HMM parameters by using SSS-free algorithm[9]. The SSS-free find the state with the widest output distribution (the largest variance), and split the state into two states. It means a cluster center is split into two clusters. After that, parameters in HMMs are updated by using the Baum-Welch algorithm. In this step, the "current" sub-sequences are used as training samples.

A NEW SPEECH CODER BASED ON ML-BEATS

In this paper, a new speech coder based on the ML-BEATS is proposed. It uses the HMM-based segment quantizer. The codebook consists of HMMs, and both HMM indexes and state indexes are sent to the decoder. At a decoding process, a decoded sequence is generated from HMM parameters by using the HMM-based speech synthesis method[7].

There are three phases to construct a speech coding system based on ML-BEATS.

Constructing an HMM-based segment codebook by using the ML-BEATS

Sequences of LSP coefficients are calculated from speech data, and first order differential coefficients (ΔLSP) are calculated as a dynamic feature for each dimension. Both LSP and ΔLSP coefficients are combined into a vector, and it is used as a training sample. ΔLSP is used for generating a decoded sequence from HMM parameters.

An HMM-based segment codebook is constructed by ML-BEATS. A total number of states in HMMs is given by hand before constructing a codebook. This parameter can control a size of the codebook. After constructing the codebook, all training samples are encoded, and statistics are calculated for Huffman coding.

Encoding an input speech

An input speech is encoded by using the HMM-based codebook. All HMMs in the codebook are concatenated in parallel, and transitions from a last state in any HMMs to a start state in any HMMs are added. It means any HMMs can be followed by any HMMs. After that, the Viterbi algorithm is carried out in order to make a correspondence between states in the HMMs and frames in the input speech.

Finally, HMM index and duration information are sent to decoder. The duration information means that how many frames are assigned into the state in the HMM. Figure 1 shows an example of encoding process. In this figure, vector sequence are aligned into several HMMs (red, blue and red again). The index of the red HMM is "0", and blue HMM is "2". The red HMM has four states, and each state is assigned to 3 frames, 4 frames, 1 frame and 6 frames at beginning of the vector sequence. Then, the encoded sequence is "0 2 3 0 5". First number is the HMM index, and other numbers are a number of self-loop transitions (it is same as a number of frames assigned to the state minus one). Finally, encoded sequence is translated by Huffman code, and sent to the decoder.

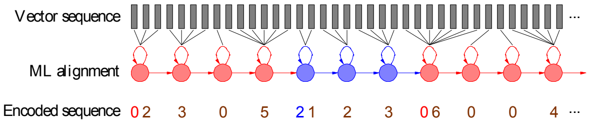

Figure 1: An example of encoding

A number of frames corresponding to an HMM is depends on both input vectors and HMMs in the codebook. If a large number of frames correspond to an HMM, encoded sequence becomes shorter. But we cannot control such a correspondence because it is determined automatically by using maximum likelihood criterion. It is a future work to propose a new controlling method to decreasing a length of encoded sequences.

Decoding a speech by using HMM-based speech synthesis method

At a decoder, an LSP coefficient sequence is generated by using both the encoded sequence and HMM parameters in the codebook. The HMM-based speech synthesis method[7] can be used for it. It can calculate the sequence which gives the highest likelihood for the HMM sequence sent by encoder. Of course, a vector which gives the highest likelihood for a state is the mean vector of the normal distribution assigned to the state. It means that the vector sequence which gives the highest likelihood for an HMM is the sequence of mean vectors. However, the HMM-based speech synthesis method can consider both LSP and ΔLSP coefficients in a vector. ΔLSP is used as a constraint of the LSP sequence in order to smooth it.

EXPERIMENTS

In order to investigate an effectiveness of the proposed method, coding experiments for LSP sequence were carried out.

Experimental conditions

LSP coefficients were calculated by the ITU-T G.729 encoder[8]. The number of dimensions of the LSP was set to 10, and it was separated into 3, 3, and 4 dimensional vectors in order to avoid "curse of dimensionality". Three codebooks were constructed independently for each separated vectors, and four codebook sizes were tested. Totally, $4^3 = 64$ experiments were carried out.

In the calculation of bit rate, Huffman coding method was applied. Other experimental conditions are shown in Table 1.

Table 1: Experimental conditions

Speech data	Japanese read speech corpus
Training	1,500 sentences uttered by 15 males and 15 females
Testing	600 sentences uttered by 6 males and 6 females
Acoustical analysis	8 kHz sampling 10 ms frame shift

Results

Figure 2 shows spectral distortion of decoded LSP sequences given by both the HMM-based coder and G.729. The red cross in this figure denotes a result given by the G.729, and each green cross denotes a result given by a combination of three codebooks. The results show that a larger size of codebook gave lower spectral distortion and higher bit rate. The lowest distortion in this experiment was 1.13 dB with 5.83 bit/frame. On the other hand, G.729 coder gave 1.02 dB with 18 bit/frame. The HMM-based coder could encode a test data only using 32% bit rate compared with G.729, even though spectral distortion increased about 0.11 dB. From this figure, increasing 1 bit brings decreasing 0.1 dB. If larger size of codebook will be constructed, the spectral distortion may reach to the result of G.729 (1.02 dB) with lower bit rate (about 7 or 8 bit/frame).

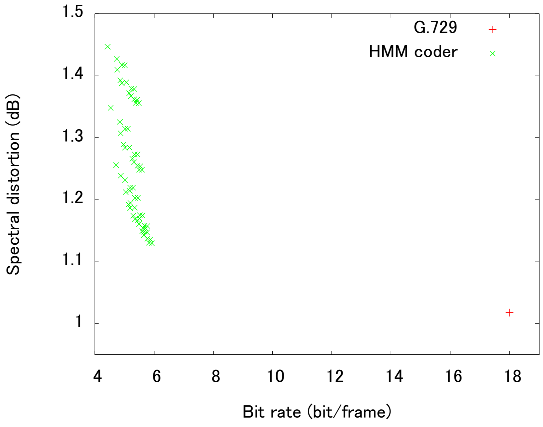

Figure 2: Spectral distortion given by the proposed method and G.729

Figure 3 shows outlier rates for the G.729 and the HMM-based coder. In this experiment, outlier of spectral distortion was defined as over 2 dB. From this figure, the HMM-based coder showed higher outlier rate than that of the G.729. The lowest rate was 7.4%, and G.729 gave only 3.1%. It means that the decoded speech given by the HMM-based coder had many “wrong” frames. It decreased a quality of speech even though an average spectral distortion was not so high. We have to investigate the reason of this problem in the future.

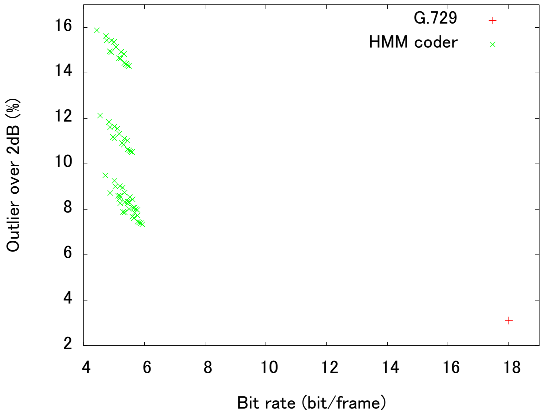

Figure 3: Outlier rate for both methods

CONSIDERING SHIFTING LSP DIMENSIONS

In both constructing codebooks and encoding steps, speech data is analyzed and 10 LSP coefficients are output for a frame. These 10 coefficients are split into three vectors, 1st-3rd, 4th-6th, and 7th-10th dimensions. However, dimensions of LSP coefficients are determined by order of coefficients. It means that if a small coefficient is inserted, larger coefficients are shifted to larger dimensions. For example, the 3rd dimension is shifted to the 4th dimension by inserting a new LSP coefficient into less than the 3rd dimension. This sometimes happens in the calculation of LSP coefficients.

It causes a mismatch between codebook and input speech. In order to solve this problem, we also propose a new codebook construction algorithm.

Construction of the shifted vector codebooks

In order to consider shifted LSP coefficients, training samples are converted into “pseudo” shifted vectors. The algorithm is as follows:

1. ***Define a length of vectors***
   A number of dimensions in a vector is defined by hand. Minimum number is 1 (it means scholar), and maximum number is 10 (order of LSP analysis).

2. ***Make a “pseudo” shifted LSP vectors from training data***
   All LSP vectors in training data are split into several vectors which have the pre-defined dimensions. In this step, dimensions are shifted, and then vectors are extracted with overlapping. For example, a number of dimensions is set to three. In this case, three dimensional vectors are extracted from a 10 dimensional vector. At first, 1st-3rd dimensions are extracted into a vector. And then, 2nd-4th dimensions are extracted. A part of this vector is overlapped with the first vector, and it simulates a dimension shifting by inserting a new LSP coefficient into 1st dimension. As a same way, totally 8 vectors are extracted from a 10 dimensional vector.

3. ***Construct the universal codebook***
   Using all shifted vectors, one codebook is constructed by ML-BEATS. In this method, the codebook is used for all vectors because it has to deal with any shifted vectors.

4. ***Construct Huffman codes for each dimension***
   Non-shifted vectors (for example, 1st-3rd, 4th-6th, 7th-9th, and 8th-10th) are extracted from training data, and encode these using the universal codebook. After that, Huffman codes are constructed. The codebook is used for all vectors, but Huffman code is constructed for each vector set. For example, a Huffman code is constructed by only using vectors consisting of 1st-3rd dimensions, and other Huffman codes are also constructed by using vectors consisting of 4th-6th dimensions, 7th-9th dimensions, and 8th-10th dimensions.

After constructing the universal codebook and Huffman codes, encoding and decoding are carried out. Both processes are almost the same as the HMM-based coder proposed in the previous section. In the encoding process, LSP coefficients vector is split into non-shifted vectors, and these are encoded by using the universal codebook.

In the decoding process, several dimensions are overlapped in the decoded vectors. For example, a number of dimensions is set to 4. The decoding process is carried out three times and three vectors are generated. These vectors correspond to 1st-4th, 5th-8th, and 7th-10th dimensions. In these vectors, 7th and 8th dimensions are decoded twice. In the final output, such dimensions are output an average of two decoded coefficients.

EXPERIMENTS FOR SHIFTED VECTOR METHOD

In order to investigate an effectiveness of the shifted vector method, several experiments were carried out. All experimental conditions were the same as the previous experiments.

Figure 4 shows spectral distortions given by the shifted vector method and fixed dimension method proposed in the previous section. In this figure, "l = *x*" denotes a shifted vector method in which a number of dimension were set to *x*, and "3:3:4" denotes the fixed dimension method.

From this figure, the shifted vector method decreased bit rate, but increased spectral distortion. In this method, the universal codebook is used for all vectors. Therefore, a codebook size (a number of HMMs) should be larger than that of the fixed dimension method. However, the codebook size of the universal codebook could not be increased because of computational time limitation. We have to continue a constructing a huge size of codebook in order to investigate the performance of the proposed method. From these results, a larger length of vector showed lower spectral distortion in the same bit rate. The best performance in this experiment was 1.33 dB with 3.3 bit/frame, a length of vector was set to 7.

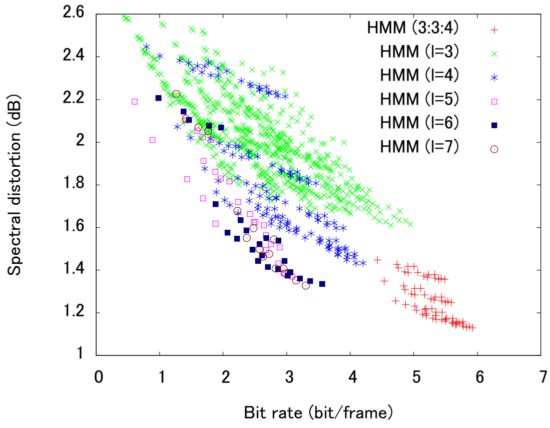

Figure 4: Spectral distortion given by shifted vector method

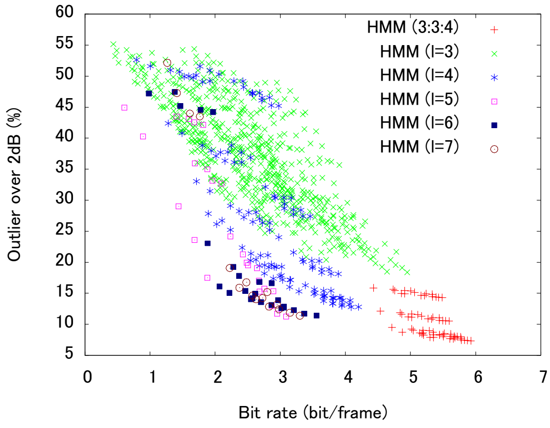

Figure 5: Outlier rates for the shifted vector method

Figure 5 shows outlier rates for shifted vector method. It became very higher than that of fixed dimension method. From this result, the shifted vector method cannot use for speech coding because of low quality. It should be improved something in order to decrease both spectral distortion and outlier rate.

CONCLUTION

In this paper, a new speech coder based on the ML-BEATS is proposed. It is based on a segment quantization, and codebook is represented by HMM. At first, training samples are clustered into several clusters by using the ML-BEATS method. The ML-BEATS splits a vector sequence into several sub-sequences, and then these sub-sequences are clustered in order to construct a codebook. Each cluster center is represented by a left-to-right HMM. In the encoding process, input speech is matched with HMMs in the codebook, and then HMM index and duration information are sent to the decoder. In the decoding process, a decoded sequence is generated from HMM parameters by applying the HMM-based speech synthesis method. From the experimental results, the HMM-based speech coder gave 1.13 dB spectral distortion with 5.83 bit/frame. It is 0.11 dB higher spectral distortion than that given by the G.729 coder, but bit rate decreased only 32%.

In order to consider a shifting problem of LSP dimensions, we also propose a new codebook construction method. Many training vectors are extracted from training samples by shifting dimensions, and all vectors are used for constructing a universal codebook. The universal codebook can deal with any shifted vectors because all possibilities are included in the training data. From the experimental results, the shifted vector method encoded an input speech with very low bit rate, but it gave higher spectral distortions.

ACKNOWLEDGMENT

This work was supported by Grant-in-Aid for Young Scientists (B), 20760229.

REFERENCES

1. D. Y. Wong, B. H. Yuang, and D. Y. Cheng, "Very low data rate speech compression with LPC vocoder and matrix quantization," in *Proc. ICASSP*, pp.65–68. (1983)
2. C. Tsao and R. M. Gray, "Matrix quantizer design for LPC speech using the generalized Lloyd algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol.ASSP-33, no.3, pp.537–545. (1985)
3. S. Roucos, R. Schwartz, and J. Makhoul, "Segment quantization for very-low-rate speech coding," in *Proc. ICASSP*, pp.1565–1570. (1982)
4. M. Kohata, M. Suzuki, A. Ito and S. Makino, "A New Segment Quantization Using Lempel-Ziv Algorithm and Its Application to Quantization of Line Spectral Frequencies," *IEEE Trans. Communications*, Vol.55, No.4, pp.661-664. (2007)
5. T. Hayashi, H. Mori, M. Suzuki, S. Makino and H. Aso, "Speech Recognition Using Acoustic Similarity-Based Primitives," *IEICE Trans. Syst. & Inf.*, Vol.J83-D-II, No.11, pp.2137-2145. (2000) (in Japanese)
6. T. Hayashi, H. Mori, M. Suzuki, S. Makino and H. Aso, "Speech recognition using acoustic segment model based on Hidden Markov Network," in *Proc. ICSP*, pp.299-304. (1999)
7. K. Tokuda, T. Kobayashi and S. Imai, "Speech parameter generation from HMM using dynamic features," in *Proc. ICASSP*, pp.660-663. (1995)
8. "Coding of Speech at 8kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP)", ITU-T Recommendation G.729. (1996)
9. M. Suzuki, S. Makino, A. Ito, H. Aso and H. Shimodaira, "A new HMnet construction algorithm requiring no contextual factors," *IEICE Trans. Inf. & Syst.*, Vol.E78-D, No.6, pp.662-668. (1995).