AN ERROR CORRECTION APPROACH BASED ON THE MAP ALGORITHM COMBINED WITH HIDDEN MARKOV MODELS

Tadashi Yonezaki , Koji Yoshida and Toshio Yagi

Telecom Research Lab., Matsushita Communication Ind.Co.,Ltd.
600 Saedo-cho,Tsuzuki-ku,Yokohama,224,JAPAN
yonezaki@telecom.mci.mei.co.jp

ABSTRACT

The error correction approach which based on a hidden Markov model (HMM) is proposed. The occurrence probability of a code sequence, which is delivered by the HMMs, is used as the measure for the maximum a posteriori probability (MAP) algorithm. The MAP algorithm is based on the assumption that the source is a discrete–time finite–state Markov process, and the HMM which models a Markov source suits well for speech data. Therefore this combination would be useful for a speech coding system. The proposing approach is adapted to the code sequence quantizing line spectrum frequency (LSF) parameters. When the code sequence is sent over a binary symmetry channel (BSC), the proposing approach with 16–state HMMs improves in code error rate and degradation of cepstrum distortion at about 27% and 39% respectively for 3% random errors.

1. INTRODUCTION

Joint source and channel coding has been investigated and developed by many researchers, eg [3][4]. Most of these researches are dedicated to reduce the redundancy laid over a code sequence. On the other hand, there are new approaches that a source model is used not for reducing the redundancy but for a channel error correction. These approaches are independent of a source coding algorithm, therefore they could be easily applied to the conventional coding system. We briefly describe two of these researches which assume a Markov source. Alajaji et al[1] proposed combination of a Markov model and a forward error control (FEC) code. The Markov model is related with the LSF's quantized by a scalar quantizer. The first and second order Markov model is used, and their approach realizes coding gain of 2–5 dB over soft–decision decoder. Garcia–Frias et al[5] proposed combination of hidden Markov models and parallel concatenated codes. The parallel concatenated codes are based on a convolutional code. The probabilities derived from the HMMs are used as the metric for a branch of trellis which expanded with production of HMM states and states of the convolutional code. They showed the combination of a source and channel model approached to the theoretical limit of the coding efficiency.

In this paper, we also propose utilizing a redundancy among a code sequence derived from a source model. Same as [5], we use a HMM as a source model. Although the HMMs produces a bit sequence in [5], it produces a code sequence in this paper. Consequently, the HMMs is difficult to combine with a convolutional code whose trellis branches correspond to a bit symbol. The corrected code sequence is estimated as the code sequence which most likely to occur. Avoiding too much dependency on a source model for the occurrence probability of the code sequence, we are also concerned about a channel models.

The occurrence probability is often referred to a posteriori probability (APP), and the algorithm which estimates the code sequence maximizing APP is called the maximum a posteriori probability (MAP) algorithm. This estimation is accomplished by iteration of replacing a code sequence and calculating the APP. In the following section, we modify the calculation of a APP based on the HMMs suitable for the MAP algorithm.

2. AN ERROR CORRECTION APPROACH WITH A SOURCE MODEL

The MAP algorithm is the one of optimal decoding algorithms. This algorithm estimates the code sequence ($\hat{\mathbf{X}}$) which maximizes a APP as the transmitted code sequence. The APP is usually referred to a conditional probability of the received code sequence ($\mathbf{Y}$) which could be calculated on a definition of a channel model. We propose it should be also referred to a probability of the code sequence produced by a source model ($\mathbf{M}$).

Consequently, the APP is calculated by

$$P(\hat{\mathbf{X}}) = P(\hat{\mathbf{X}}|\mathbf{M})^{\gamma} \cdot P(\hat{\mathbf{X}}|\mathbf{Y})^{(1-\gamma)}, \tag{1}$$

where γ is a weighting for a source model.

The MAP algorithm is constructed on the property of the APP calculation method which is based on a forward–backward algorithm, therefore a source and channel model must have this structure. In the following subsection, we derive the algorithms for calculating conditional probabilities on a source and channel model.

2.1. Probability based on a Source Model

Once the model parameters of a HMMs are defined, the HMMs ($\mathbf{M}$) provides a occurrence probability $P(\hat{\mathbf{X}} \mid \mathbf{M})$ of a code sequence $\hat{\mathbf{X}}$. This probability is effectively calculated with the definition of a "forward–going" and a "backward–going" distributed probabilities over the HMM states[2, pp.687–689]. These probabilities are denoted in this paper as $\{\alpha(q,n) \mid q = 1, \cdots, N_s\}$ and $\{\beta(q,n) \mid q = 1, \cdots, N_s\}$ respectively, where N_s is the number of HMM states and n is referred to time or frame number. The $\alpha(q,n)$ and $\beta(q,n)$ are defined as

$$\begin{aligned} \alpha(q,n) &= \sum_{s=1}^{N_s} \alpha(s, n-1)\ a_{sq}\ b_{sq}(\hat{x}_n) && (2) \\ \beta(q,n) &= \sum_{s=1}^{N_s} \beta(s, n+1)\ a_{qs}\ b_{qs}(\hat{x}_{n+1}), && (3) \end{aligned}$$

where a_{sq} is the transition probability between HMMs from state s to state q and $b_{sq}(\hat{x}_n)$ is the output probability of the code $\hat{x}_n$ on certain state transition. The initial distributions of the probabilities are $\{\alpha(q,0) = \pi_q \mid q = 1, \cdots, N_s\}$ and $\{\beta(q,N) = 1 \mid \text{for all legal final states } q\}$ respectively, where $\sum_1^{N_s} \pi_q = 1$ and N is the length of the code sequence. The occurrence probability of the code sequence $\hat{\mathbf{X}}$ is calculated with α and β by

$$\begin{aligned} & P(\hat{\mathbf{X}}|\mathbf{M}) \\ &= \sum_{q=1}^{N_s} \alpha(q,n) \cdot \beta(q,n) \\ &= \sum_{q=1}^{N_s} \sum_{s=1}^{N_s} \alpha(s, n-1)\ a_{sq}\ b_{sq}(\hat{x}_n) \cdot \beta(q,n) && (4) \end{aligned}$$

for arbitrary n.

The α and β would underflow for the long code sequence, therefore these probabilities are normalized at every frame by

$$\begin{aligned} \bar{\alpha}(q,n) &= \frac{1}{S_\alpha^{(n)}} \alpha(q,n) && (5) \\ \bar{\beta}(q,n) &= \frac{1}{S_\beta^{(n)}} \beta(q,n), && (6) \end{aligned}$$

where

$$\begin{aligned} S_\alpha^{(n)} &= \sum_{q=1}^{N_s} \alpha(q,n) && (7) \\ S_\beta^{(n)} &= \sum_{q=1}^{N_s} \beta(q,n). && (8) \end{aligned}$$

This normalization make the equation (4) be

$$\begin{aligned} & P(\hat{\mathbf{X}}|\mathbf{M}) \\ &= \prod_{i=1}^{n} S_\alpha^{(i)} \cdot \{\sum_{q=1}^{N_s} \bar{\alpha}(q,n) \cdot \bar{\beta}(q,n)\} \cdot \prod_{j=n}^{N} S_\beta^{(j)} \\ &= \prod_{i=1}^{n-1} S_\alpha^{(i)} \cdot \prod_{j=n}^{N} S_\beta^{(j)} \cdot \\ & \quad \sum_{q=1}^{N_s} \sum_{s=1}^{N_s} \bar{\alpha}(s, n-1)\ a_{sq}\ b_{sq}(\hat{x}_n) \cdot \bar{\beta}(q,n). && (9) \end{aligned}$$

2.2. Probability based on a Channel Model

A occurrence probability of a code sequence $\hat{\mathbf{X}}$ on a channel model is calculated as a conditional probability of a received code sequence $\mathbf{Y}$. For example, when a channel model is defined as a memoryless binary symmetry channel (BSC) with crossover probability r, a conditional probability is

$$\begin{aligned} P(\hat{\mathbf{X}}|\mathbf{Y}) &= \prod_{i=1}^{N} P(\hat{x}_i|y_i) \\ &= \prod_{i=1}^{N} r^{D_h^{(i)}} (1-r)^{(b - D_h^{(i)})}, && (10) \end{aligned}$$

where $D_h^{(i)}$ is the Hamming distance between $\hat{x}_i$ and y_i, b is the bit length of a code. In this example, we assume a memoryless channel therefore equation (10) can be divided into two parts corresponding to forward and backward at any frame. Even a noisy channel model with memory, we can derive a forward–backward property of the conditional probability, for instance, from a trellis of a viterbi decoding algorithm for a convolutional code. We should mention that a forward–backward algorithm on a channel model should be based on a code sequence, as an algorithm on a source model is.

2.3. Probability for MAP estimation

The MAP algorithm maximizes equation (1) iteratively. We denote $\bar{\mathbf{X}}^{(\mathbf{k})}$ as the estimated code sequence of kth iteration. A $\bar{\mathbf{X}}^{(k+1)}$ is constrained to differ with the $\bar{\mathbf{X}}^{(k)}$ only on one code $x_n^{(k)}$. APPs are calculated for all possible code sequence $\bar{\mathbf{X}}^{(k+1)}$, and the most likely code sequences selected. In stead of using the APPs themselves, we use the ratio between the APPs of a $\bar{\mathbf{X}}^{(k+1)}$ and the $\bar{\mathbf{X}}^{(k)}$. The algorithm is terminated when the APP reaches to the local maximum probability, that is the ratio becomes one. The ratio is easily calculated by

$$\begin{aligned}
&\frac{P(\bar{\mathbf{X}}^{(k+1)})}{P(\bar{\mathbf{X}}^{(k)})} \\
&= \frac{P(\bar{\mathbf{X}}^{(k+1)}|\mathbf{M})^{\gamma} P(\bar{\mathbf{X}}^{(k+1)}|\mathbf{Y})^{1-\gamma}}{P(\bar{\mathbf{X}}^{(k)}|\mathbf{M})^{\gamma} P(\bar{\mathbf{X}}^{(k)}|\mathbf{Y})^{1-\gamma}} \\
&= \left(\frac{\sum_{s=1}^{N_s} \bar{\alpha}(s, n-1)\; a_{sq}\; b_{sq}(\bar{x}_n^{(k+1)}) \cdot \bar{\beta}(q, n)}{S_{\alpha}^{(n)} \cdot \sum_{q=1}^{N_s} \bar{\alpha}(q, n) \cdot \bar{\beta}(q, n)} \right)^{\gamma} \\
&\quad \times \left(\frac{P(\bar{x}_n^{(k+1)}|y_n)}{P(\bar{x}_n^{(k)}|y_n)} \right)^{(1-\gamma)} \qquad (11)
\end{aligned}$$

for arbitrary code $\bar{x}_n^{(k+1)}$. The initial code sequence is $\bar{\mathbf{X}}^{(0)} = \mathbf{Y}$.

3. EXPERIMENTAL RESULTS

The proposed algorithm is adapted to error correction of code sequences suffered from a random bit error. The code sequence corresponds to the 10th order LSF parameter sequences which are quantized with 8-bits codebook. Parameters are derived from a speech data every 20ms with 25ms–lengths Hamming window. We use 1200 utterances for training the HMMs and the codebook and 200 utterances (100 each of the trained and unknown data) for evaluation. The ergodic HMMs which has 2, 4, 8, 16 states are used as a source model. The memoryless BSC whose crossover probability, ie a bit error rate (BER), is varied from 0.1 to 3% is used as a channel model. We assume the algorithm knows the BER of input code sequences. There are no additional forward error control codes (FEC) at all.

3.1. Improvement of the Code Error Rate

Figure 1 shows the improvement of the code error rate (CER). The parameter is quantized with 8-bits codebook, hence the CER 21.7% corresponds to the BER 3%. The plain line in the figure 1 indicates without error correction.The code error correction performance is improved with increasing the number of HMM states. The CER at 21.7% is reduced to 19.8%, 18.3%, 17.0%,16.0% with 2, 4, 8, 16–states HMM respectively. Typically, the proposing algorithm suppresses the CER about 26.3% with 16–states HMM at the CER 21.7%.

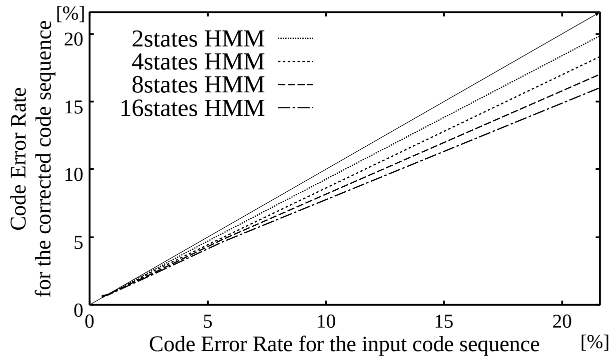

Figure 1: Improvement of the code error rate

Figure 1 also shows slight degradation of the CER at a low CER. It could be occurred by too much dependence on a source model which hard to duplicate a source property. It will be eliminated by the combination with a channel coding.

3.2. Recovering the degradation on Cepstrum Distortion

In this subsection, we evaluate the recovery of the degraded cepstrum distortion by a channel error. Cepstrum distortions are calculated based on the parameter sequence which are decoded from a code sequence without errors. Figure 2 shows the degradation of the code sequence suffered from channel errors on a cepstrum distortion. Same as 3.1, the performance is improved with increasing the HMMs state number. At the BER 3%, it shows 13%, 23%, 33%, 39% reduction of cepstrum distortion for the number of HMM states 2, 4, 8, 16 respectively. These results show the better performance than the CER. Table 1 ensure it. Table 1 shows the relation between the performance and amount of a degradation. We can see from table 1, proposing algorithm corrects the code which cause larger distortion. This is the reason why the performance evaluated on recovery of degradation is better than on improvement of CER. This property will be advantageous because the human auditory would be more sensitive to the larger degradation.

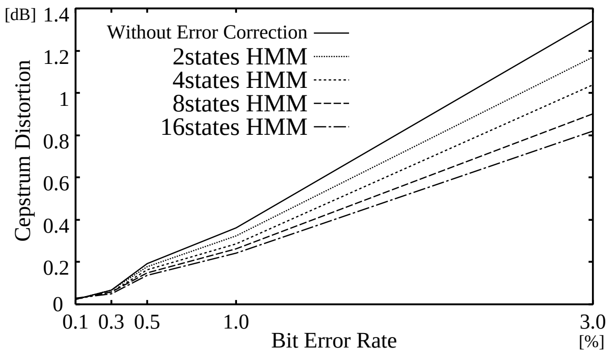

Figure 2: Suppression of the degradation on cepstrum distortion

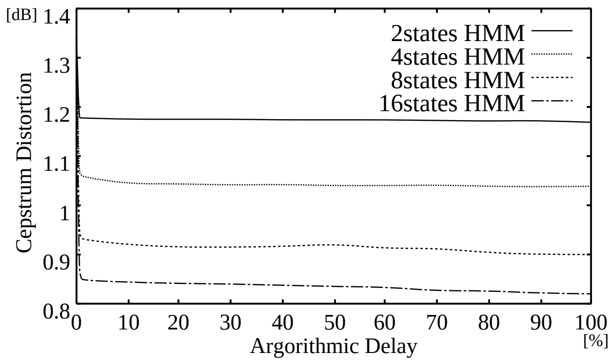

Figure 3: The performance vs. the algorithmic delay: the axis for the algorithmic delay refers to the ratio of the number of the processing frames to the entire frame number

Table 1: Recovery for amount of a degradation

Type	CD [dB]	Outlier [%] 2 – 4 [dB]	Outlier [%] > 4 [dB]
without error correction	1.34	5.78	15.81
2states	1.09	5.77	13.14
4states	0.93	5.69	11.09
8states	0.78	5.69	9.21
16states	0.68	5.57	8.03

3.3. Algorithmic Delay

Figure 3 shows the relation between a performance and an algorithmic delay. Each speech data has different length, therefore the axis for the algorithmic delay refers to the ratio of the number of the processing frames to the entire frame number. The average frame number of evaluated data is 227.38 (about 4.5 seconds). Figure 3 indicates only a few flames delay (about 1%) is enough for realizing the nearly optimal performance.

4. CONCLUSION

We propose the error correction approach based on the HMMs. This approach uses a redundancy among code sequence therefore can be adapted to any conventional source coding algorithm. We evaluate it by correcting the code sequence which is suffered by random bit error through the BSC. It realizes 39% reduction of cepstrum distortion degraded by channel errors without any error correction codes.

5. REFERENCES

[1] Fady I. Alajaji, Nam C. Phamdo, Thomas E. Fuja, "Channel Codes That Exploit The Residual Redundancy In CELP-encoded Speech", IEEE Trans. SAP, vol. 4, no.5, pp.325–336, 1996.

[2] John R. Deller, Jr., John G. Proakis, John H.L. Hansen, "Discrete-Time Processing Of Speech Signals", Macmillan Publishing Company, 1993.

[3] James G. Dunham, Robert M. Gray, "Joint Source And Noisy Channel Trellis Encoding", IEEE Trans. IT, IT-27, no.4, pp.516–519, 1981.

[4] Thomas R. Fischer, Michael W. Marcellin, Min Wang, "Trellis-Coded Vector Quantization", IEEE Trans. IT, IT-37, no.6, pp.1551–1566, 1991.

[5] Javier Garcia-Frias, John D. Villasenor, "Combining Hidden Markov Source Models And Parallel Concatenated Codes", IEEE Trans. on Commun. Lett.,vol. 1, no.4, pp.111–113, 1997.