

A TRELLIS-BASED IMPLEMENTATION OF MINIMUM ERROR RATE TRAINING

Kazuya Takeda Tetsunori Murakami Shingo Kuroiwa Seiichi Yamamoto

KDD R&D Laboratories
2-1-15 Ohara Kamifukuoka-shi, Saitama, 356 Japan

ABSTRACT

A new implementation of ME (Minimum Error rate) training is proposed. The most important difference from conventional ME training is the use of a trellis-based calculation for the discriminant function, instead of the Viterbi based calculation of the conventional training. The key idea of the training is to use a matrix representation of state transit probabilities of an HMM for calculating the discriminant function so as to simplify the differential operation on the misclassification and loss functions. From the non-segmental characteristics of the discriminant function, loss functions for substitution, insertion and/or deletion errors are easily calculated by substituting, inserting and/or deleting the matrices for the corresponding HMM units of the loss function. Based on the proposed training, therefore, both string level and unit level error minimizations are easily integrated.

1. INTRODUCTION

While various general technologies related to constructing an HMM-based speech recognizer have been invented, *tuning* a system to a specific task or vocabulary becomes to play a relatively important role in implementing an application system. From the standpoint of tuning a system parameter such as pdfs in word or subword HMMs, recent improvements in discriminative training provide significant improvement in preventing of serious errors in the task domain of the system.

Among several discriminative training algorithms, such as maximum mutual information (MMI) training[1] or corrective training [2], minimum error rate (ME) training based on GPD [3], [4] has been center of the research for its inherent error minimization characteristics. In a recent paper [5], furthermore, Chou et al. extended the ME algorithm to minimum *string* error rate training and demonstrated its effectiveness.

GPD-based ME training has been based on the segmental procedure of Viterbi scoring, because of its simple mathematical formula for differentiating the loss function. From the original definition of hidden Markov modeling, however, forward-likelihood obtained by trellis operation would seem to be better for the training and scoring. The training procedure that we propose in this paper is an alternative approach using the trellis calculation for forward-likelihood as the criterion of missclassification in ME training. By utilizing matrix parsing technology in the training we provide not only a simple analytic formula for differentiation of the score function, but also

top-down hypothesizing for near-miss competitive strings so that the training is directed to reduce certain error sources.

The remainder of this paper has the following content. In Section 2, the authors will compare the recognition performances obtained by Viterbi score, and forward-likelihood and show the superiority of the latter. In Section 3, trellis-based ME training is described after brief descriptions of conventional ME training and matrix parsing. In Section 4, experimental results are summarized, followed by the discussion in Section 5.

2. TRELLIS VS. VITERBI

Most of the HMM-based speech recognition systems adopt Viterbi scoring mainly because it is easy to combine with various time synchronous search algorithms. From its original idea of modeling speech production by *hidden* Markov process, however, the forward-likelihood seems to offer better scoring for speech recognition. In [6], for instance, Schwartz et al. mention the suboptimality of Viterbi scoring, showing twice as many errors as forward-likelihood in finding the 100-best sentences in their N-best algorithm. Before describing trellis-based ME training the authors compare the recognition accuracy of Viterbi scoring and forward-likelihood to show the importance of optimizing HMM parameters in terms of forward-likelihood.

For the experiment, 6000 phrases of connected Japanese digit utterances were used. The length of all digit strings is four. The utterances were spoken by 21 male speakers with the same handset and recorded over the public telephone network. A total of 5320 phrases of 19 speakers were used to train 10 digit and pause HMMs and 680 phrases of 2 other speakers were used for the comparison. HMMs were trained by 4 iterations of HRest after bootstrapping with HRest of HTK 1.4. HVite and the matrix-based evaluation method described in Section 3 were used for evaluating Viterbi score and forward likelihood, respectively.

The results are listed in Table 1. As shown in the table, forward-likelihood achieves a more accurate result than Viterbi scoring. Especially, Deletion and Substitution errors were reduced more than half, phrase error rate is one half that of Viterbi scoring when the length of string is given (b).

Table 1. Comparison between Viterbi and forward likelihood scoring for recognizing connected digit utterance under the conditions that length of string is unknown (a), and that the length of string is given (b).

(a) word error rate					
	Del.	Ins.	Sub.	%Acc	%Corr
Viterbi	7	5	6	97.5	98.2
forward	2	11	3	97.8	99.3

(b) phrase correct rate.	
	% Phrase Correct
Viterbi	94.1
forward	97.5

3. TRAINING PROCEDURE

3.1. ME Training

The procedure of the ME training is outlined below. First, in the ME training, the discriminant function is defined as

$$g(O, S_k, \Lambda) = \log f(O, S_k, \Lambda), \quad (1)$$

where O is the observation sequence, S_k is the k -th candidate word string, f is the scoring function and Λ is a set of HMM parameters. Note that Θ , i.e. optimal state sequence, in the conventional formula of the segmental approach is not involved in the above equation.

The below misclassification measure is, therefore, defined as the difference between the discriminant functions of the correct unit sequence and the average value of the discriminant functions of the incorrect sequences.

$$d(O, \Lambda) = -g(O, S_{cor}, \Lambda) + \log \left\{ \frac{1}{N-1} \sum_{S_k \neq S_{cor}} e^{\eta g(O, S_k, \Lambda)} \right\}, \quad (2)$$

where S_{cor} is the correct word sequence for the token. Once the misclassification measure, $d(\cdot)$, is given for each training token, the error function can be evaluated by

$$l(O, \Lambda) = \frac{1}{1 + e^{-\gamma d(O, \Lambda)}}, \quad (3)$$

and summing up all the values of the error function over all token, we can evaluate the loss function

$$L(O, \Lambda) = \sum_{\text{all tokens}} l(O, \Lambda). \quad (4)$$

Finally, replace the value of each parameter by moving in the direction given by differentiating the loss function with respect to each parameter;

$$\gamma_{n+1} = \gamma_n - \epsilon_\gamma (n) \frac{\partial L(O, \lambda)}{\partial \lambda}. \quad (5)$$

To obtain $\frac{\partial L}{\partial \lambda}$, we have to evaluate the differentiation of the discriminant function $g(\cdot)$ given by (1) with respect to each HMM parameter λ , and therefore differentiations

of the score function $f(\cdot)$ over all competitive strings. In the conventional segmental implementations, the Viterbi scoring function $f(\cdot)$ is given as a product of transition probabilities ($\{a_{ij}\}$) and output probabilities ($\{b_{ij}(O_t)\}$) and thus the logarithm of the score is given as the sum of the logarithm of each term,

$$\log f(O, S, \Lambda) = \sum_{t=1}^T \log(a_{s(t), s(t+1)}) + \sum_{t=1}^T \log(b_{s(t)}(O_t)) \quad (6)$$

for the optimal state sequence $\{s_t\}_{t=1}^T$. For this form of score function, differentiation is straightforward.

In forward-likelihood scoring, however, we have to sum up the scores of all possible state sequences, which is referred as trellis calculation, and differentiation of the score functions is not simple. Therefore, we introduce the matrix operation for calculating forward likelihood as shown in the next section.

3.2. HMM scoring based on matrix operation

In [7], Singer et al. formulated an integrated framework for HMM scoring and parsing by a simple matrix operation which they called Matrix Parsing. In matrix parsing, the probability calculation of an HMM or a sequence of HMMs is formalized through matrix operation on $(T+1) \times (T+1)$ matrices, a likelihood matrix, for the input speech of length T . The element of the r -th row of the c -th column of the likelihood matrix stands for the probability of staying in the HMM state from time r to time c . Thus the likelihood matrix for a one clock state transition from state i to j at arbitrary time for observation O is given by

$$q_{ij}(O) = a_{ij} \times \begin{bmatrix} 0 & b_{ij}(O_1) & 0 & \dots & 0 \\ 0 & 0 & b_{ij}(O_2) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & b_{ij}(O_T) \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad (7)$$

where a_{ij} is the state transition probability and $b_{ij}(O)$ is the output probability. Obviously, for a one clock transition, the likelihood matrix has a non-zero value at only the $(r, r+1)$ element, and, in general, likelihood matrices are upper triangular.

The likelihood matrix of staying at state i for an arbitrary time period (from time r to time c) can be represented as

$$p_{ii} = I + \sum_{t=1}^T q_{ii}^t, \quad (8)$$

where I is an identity matrix.

Thus, for a unit HMM consisting of 'left-to-right' M -state with $M-1$ loops, the likelihood matrix R is obtained as the product of q_{ij} and p_{ii} :

$$R = (p_{11} q_{12}) \dots (p_{M-1, M-1} q_{M-1, M}), \quad (9)$$

assuming that the output probabilities of each state transition depend only on the source state. For a string model consisting of N -unit HMMs, the likelihood matrix F is also the product of R matrices of units as:

$$F(O, S, \Lambda) = R_1 \dots R_s \dots R_N, \quad (10)$$

where F is a likelihood matrix corresponding to the concatenative HMM of unit sequence S , with Λ as the parameter set of HMMs and O as the observation sequence. The forward-likelihood is the likelihood that transit from the initial state at time 0 to the final state at time T , i.e. $(1, T+1)$ element of the likelihood matrix F , is given by

$$f(O, S, \Lambda) = [1, 0 \dots 0] F(O, S, \Lambda) \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}. \quad (11)$$

The merit of the above interpretation of HMM scoring is to give an analytic form for forward-likelihood calculation which is usually given by the recursive operation. Based on this form, we will implement trellis-based ME training in the next section.

3.3. Trellis-Based ME Training

For the minimum string error training, in [5], Chou et al. used N-best decoding for finding competitive strings. Although the same approach can be used for the proposed method, we introduce an alternative approach to find competitive strings. Because any string other than the correct string can be generated by inserting, deleting and/or substituting words to/from the correct string, we can find the competitive strings and their scores by evaluating the discriminant function resulting from each of the three operations. As shown in Figure 1, the resulting discriminant function of each of the three operations is easily calculated by doing the same operation to/from the product of R_i matrices. Thus, evaluating the recognition accuracy of the given HMM set not only for substitution errors [8], but also insertion and deletion errors, can be conducted by the matrix operations. Furthermore, differential operation on the loss function can be achieved by summing up all the differentiations of the discriminant functions of competitive strings as below.

In order to implement ME training, we need to evaluate differentiations of score function $f(O, S, \Lambda)$, which is not simple with the recursive form of calculating forward-likelihood. With a given analytic form for the forward-likelihood score (11), however, the differentiation operation of the loss function can easily be evaluated based on the matrix operation, which is not straightforward in the recursive form, as

$$\frac{\partial F(O, S, \Lambda)}{\partial \lambda_s} = R_1 \dots \frac{\partial R_s}{\partial \lambda_s} \dots R_N, \quad (12)$$

where λ_s is a parameter of the HMM for the s -th unit in the string S . From the definition of R ,

$$\frac{\partial R_s}{\partial \lambda_{s,i}} = p_{11} q_{12} \dots \frac{\partial(p_{ii} \cdot q_{ii+1})}{\partial \lambda_{s,i}} \dots p_{M-1M-1} q_{M-1M} \quad (13)$$

$$\frac{\partial(p_{ii} \cdot q_{ii+1})}{\partial \lambda_{s,i}} = p_{ii} \cdot \frac{\partial q_{ii+1}}{\partial \lambda_{s,i}} + \frac{\partial p_{ii}}{\partial \lambda_{s,i}} \sum_{t=1}^T (t q_{ii}^{t-1}) \cdot q_{ii+1}, \quad (14)$$

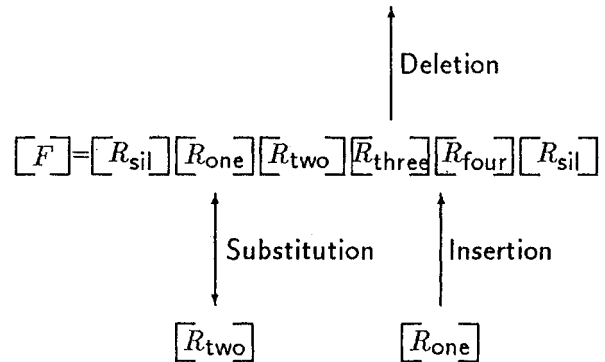


Figure 1. Matrix calculation of discriminant functions corresponding to insertion deletion and substitution errors. The correct string is /sil-one-two-three-four-sil/, and the error string for substitution is /sil-two-two-three-four-sil/, for insertion /sil-one-two-three-one-four-sil/, and for deletion /sil-one-two-four-sil/.

where

$$\frac{\partial q_{ij}(O)}{\partial \lambda_{s,i}} = a_{ij} \times \begin{bmatrix} 0 & \frac{\partial b_{ij}(O_1)}{\lambda_{s,i}} & 0 & \dots & 0 \\ 0 & 0 & \frac{\partial b_{ij}(O_2)}{\lambda_{s,i}} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{\partial b_{ij}(O_T)}{\lambda_{s,i}} \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (15)$$

Thus, the differentials of the discriminant functions of competitive strings with respect to $\lambda_{s,i}$, i.e. the parameter of the i -th state of the s -th unit HMM, are easily obtained by replacing the corresponding matrix in F with the differential one.

4. EXPERIMENT

A preliminary experiment has been performed to confirm the implementation of the algorithm, using a rather small amount of data, 120 Japanese connected digit utterances of 19 male speakers out of the 6000 utterances used in Section 2. To reduce the amount of calculation, we used three mixture components for the distribution of 8 MFCC parameters and their derivatives, and trained only mean and weight values (i.e. all variances are fixed through training.) The initial HMMs are trained by 6000 male utterances using HRest. The training procedure was as follows:

- Step 1 Calculates differentiations of loss function.
- Step 2 Generates a set of new parameters descending in the direction of the differentiation with several different values of ϵ .
- Step 3 Selects the best parameters by evaluating the recognition accuracy and return to Step 1.

In Figure 2, the loss values and the numbers of word errors are plotted for insertion, substitution and deletion errors at each reestimation. Since the most dominant

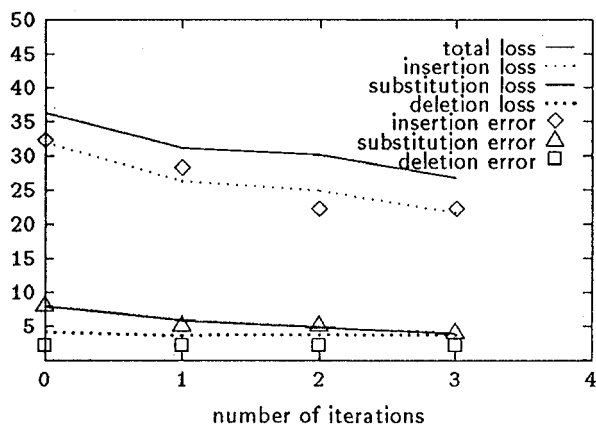


Figure 2. Reduction of the loss value and the number of errors for each reestimation. The vertical axis is corresponding to both the value of the loss function and the number of errors.

error of the initial HMM was insertion error, the reduction of the loss value due to insertion error was the main source of the reduction of total loss value. By the training, the number of insertion errors was reduced to the two third of the original HMM (from 32 to 22), as well as the loss value due to the insertion error (from 31.9 to 21.7). The correspondance between number of errors and the loss value also held in the case of the substitution error, where number of errors was reduced by three and one in the first and the third iteration, respectively. The loss value due to the deletion error was not significantly changed while the training (from 4.15 to 3.60), and the number of the deletion error was not reduced. The same tendency of error reduction was obtained from the recognition experiment using the same number of open data, where the insertion error was reduced from 40 to 23, substitution error was reduced from 19 to 14 and no error reduction was obtained for deletion error.

5. DISCUSSION

The proposed method has two important merits, 1) the training is nonsegmental, and 2) the discriminant functions of competitive strings can be easily calculated using the discriminant function of the correct string. As for the first merit, we believe that non-segmental scoring for forward-likelihood performs better than segmental one. Because early papers concluded insignificant difference between Viterbi and forward-likelihood, for example [9], conducted mainly isolated word based experiments, in which the non-segmental technique is only used for state sequence but not for inter HMMs. As pointed out in Section 2, under some critical conditions of continuous speech recognition where the recognition accuracy exceeds 95 %, the difference between Viterbi and forward-likelihood scoring is no more insignificant.

We also believe that the second merit is important for minimizing the string error rate. In string error minimization, a correct string potentially has a great number of competitive strings, which are sometime very different

from the correct one and from each other, even though their scores are comparable with each other. This means that we have to train the system dealing various kinds of errors at once. On the other hand, explicitly assuming the type and source of errors for generating competitive strings, the proposed method can proceed with training step by step; first train model A and B to reduce their substitution, then train model C to prevent insertion errors, etc.

Since we have not completed the evaluation for full implementation with a sufficient amount of training data, the most important future work will be comparative experiments between proposing ME training and segmental training.

ACKNOWLEDGEMENT

The authors are grateful to Dr. Urano and Dr. Murakami, Director and Deputy Director of KDD R & D Laboratories. They are also grateful to Professor Kurematsu of ECU for his helpful discussions.

REFERENCES

- [1] L.R.Bahl, P.F.Brown, P.V.Souza and R.L.Mercer, "A new algorithm for the estimation of hidden Markov model parameters", *proc. ICASSP*, pp.493-496, 1988
- [2] K.F.Lee and S.Mahajan, "Corrective and reinforcement learning for speaker-independent continuous speech recognition", *Proc. EuroSpeech 89*, Vol.1 pp.490-493, 1989
- [3] H.Fraanco and A.Serralheiro, "Training HMMs using a minimum recognition error approach", *Proc. ICASSP 91*, pp.357-360, 1991
- [4] W.Chou, BH.Juang and CH.Lee, "Segmental GPD training of an hidden markov model based speech recognizer", *Proc. ICASSP 92*, pp.473-4476, 1992
- [5] W.Chou, C.H.Lee and B.H.Juang, "Minimum error rate training based on N-best string models", *Proc. ICASSP 93*, pp.II652-II655, 1993
- [6] R. Schwartz and S.Austin, "A comparison of several approximate algorithms for finding multiple (N-best) sentence hypotheses", *Proc. ICASSP 91*, pp.701-704, 1991
- [7] H.Singer and S.Sagayama, "Matrix parser and its application to HMM-based speech recognition", *Proc. ICASSP 93*, pp II295-II298, 1993
- [8] Y.Minami, T.Matsuoka and K.Shikano, "Phoneme HMM evaluation algorithm without phoneme labeling", *Proc. ICSLP 92*, pp.1535-1538, 1992
- [9] L.R.Rabiner, S.E.Levinson and M.M.Sondhi, "On the application of vector quantization and hidden Markov models to speaker independent, isolated word recognition", *The Bell System Technical Journal*, Vol.62, 4, pp.1075-1105, 1983