STUDIA PHONOLOGICA XXV (1991)

Phoneme Probability Presentation of Continuous Speech based on Phoneme Spotting

Yasuo ARIKI

SUMMARY

This paper describes a new presentation of continuous speech in terms of the probability of all phoneme types as a function of time. The presentation is called a phoneme probability presentation (PPP) and can be used for phoneme recognition of continuous speech. As a technique to produce the PPP, we have employed hidden Markov models (HMM) with time duration information. This information is essential to spot the phonemes and to produce the PPP. With this information the HMMs of all the phoneme types can compute their probability in parallel and in time synchronism. The PPP can serve as phoneme filters which can produce phoneme probability from continuous speech.

I. INTRODUCTION

In continuous speech recognition systems, speech models (knowledge) of phonemes, diphones, triphones or words are applied to continuous speech signals, together with language models (knowledge) such as finite state automata (FSA), bigram or tirgram models. These two levels, speech and language modles, are strongly interrelated in the recognition process. As a step toward analysing the interrelation, here we consider two kinds of recognition methods derived from different application of the two models, in a separate or integrated manner.

When the two models are applied separately in the system a phoneme lattice may be produced as an intermediate presentation by applying the speech models. Then the language models are applied to interpret the lattice. [1] Since segmentation and labelling of continuous speech into a phoneme lattice causes an enormous increase in hypotheses, simple attempts to reduce the hypotheses in a deterministic way leads to loss of information. In order to keep as much information as possible, more robust presentation of phonetic information is required for continuous speech, as an alternative to a deterministic lattice representation. Phoneme probability presentation shows the probability of all the English phoneme types, for example 44 types (20 vowels and 24 consonants), as a function of time in continuous speech. It represents one method to keep as much phonetic information as possible by presenting the phoneme ambiguity on position and labels in a probabilistic manner.

The second method is an integrated application both of the speech models and the language models to continuous speech. [2] [3] [4] [5] This method is po-

Yasuo ARIKI (有木康雄): Associated Professor, Department of Electronics and Informatics, Ryukoku University.

werful due to the integration of these two models with global optimisation at the sentence level. However, the computation time is large because all the phonemes contained in all the words in the lexicon must be processed. The total number of phonemes included in the lexicon is certainly larger than the number of phoneme types. In order to reduce the computation time, duplication of the phonemes in the lexicon must be avoided. This requires some intermediate presentation of phonetic information for phoneme types, instead of all the phonemes in the lexicon at the expense of the global optimisation. The phoneme probability presentation is one of the methods which can avoid the duplicate computation of the phoneme probability.

The method proposed here, termed the *phoneme probability presentation* (*PPP*), results from the above two requirements. It can present continuous speech in terms of probabilities of phoneme types as a function of time. The PPP can be derived by spotting each phoneme type on the continuous speech using Hidden Markov Models with time duration constraints. On this presentation, phoneme position and its ambiguity are presented as a probabilistic time function. This presentation can be viewed as a phoneme filter which produces a probability time function for each phoneme type from the continuous speech signal. The phoneme lattice could be easily generated by searching for the best phoneme sequence, using dynamic programming techniques. Realisation of word lattice, word spotting and language model based word parsing are also feasible on this presentation.

II. Definition of PPP

The PPP is two-dimensional probability presentation of phoneme types as a function of time. One axis corresponds to 44 English phoneme types and the other is the time axis. Deterministic phoneme segmentation or phoneme lattice can be presented in a similar two-dimensional plane. We derive the PPP definition by modifying the deterministic algorithm of phoneme segmentation.

Fig. 1 shows an Example of phoneme segmentation. The vertical line shows the 44 phoneme types and the horizontal line corresponds to time scale. The resultant segment is referred to by a number from 1 to K. The start time and the

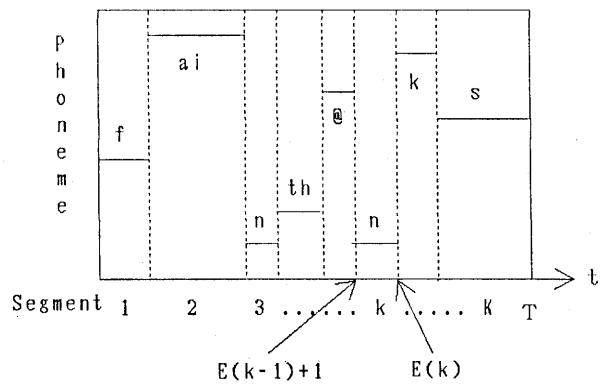

Fig. 1. Example of phoneme segmentation.

end time of the kth segment are $E(k-1)+1$ and $E(k)$ respectively. The probability of the phoneme p_n to the kth segment is represented as follows:

$$D((E(k-1)+1, E(k)),p_n) \tag{1}$$

where $D(\ )$ is a function to compute the probability. The probability of the phoneme sequence shown in Fig. 1 is represented as:

$$\sum_{k=1}^{K} D((E(k-1)+1, E(k)),p_n) \tag{2}$$

For the purpose of the segmentation of continuous speech into the phoneme sequence, the best method is to optimise EQ.(2) in terms of the number of phonemes K, position of phoneme boundaries Δ_K and the phoneme type p_n. Then the optimisation equation is obtained as follows:

$$A(T)=\max_{K} \max_{\Delta K} \max_{p_n} \sum_{k=1}^{K} D((E(k-1))+1, E(k)),p_n) \tag{3}$$

It is well known that the above equation can be simplified and represented by the following Dynamic Programming (DP) recursive equation:

$$A(t)=\max_{l}\{A(l)+\max_{p_n} D((l+1,t),p_n)\} \tag{4}$$

where l corresponds to the phoneme boundary. Two-level DP[6], level building [7] and one-stage DP[8] are proposed to compute the number of phonemes, the optimised position of phoneme boundaries and the associated phoneme type. Equation (4) can be modified and approximated at the expense of global optimisation as follows:

$$\begin{aligned} A(t) &= \max_{p_n} \max_{l}\{A(l)+D((l+1,t),p_n)\} \\ &\approx \max_{p_n}\{A(l)+\max_{l} D((l+1,t),p_n)\} \\ &= \max_{p_n}\{A(s)+P_s(p_n,t)\} \end{aligned} \tag{5}$$

Here $P_s(p_n,t)$ is the highest probability of the phoneme type p_n to the segments ending at time t. The corresponding starting time is denoted as s. We call this the phoneme probability presentation PPP since the $P_s(p_n,t)$ can present the probability of all the phoneme types as a function of time together with the best starting time, as a result of spotting phoneme type frame by frame. Equation (5) indicates that continuous speech can be approximately segmented into a phoneme sequence by using locally optimised phoneme probability presentation PPP, $P_s(p_n,t)$.

III. Algorithm to Produce PPP

The PPP is produced by computing the phoneme probability $P_s(p_n,t)$ for all the phoneme types p_n at every time t. The best starting time s must be sought for every ending time t. As a technique to compute the phoneme probability $P_s(p_n,t)$, we employ hidden Markov models (HMM). We can mention the following reasons for employing HMM.

(1) HMM can compute the degree of matching between phoneme models and speech segments as a probability.
(2) HMM can absorb the spectral variabilities caused for the same phoneme even

by the same speaker.

(3) HMM can normalise the time duration difference in matching between speech model and speech data using dynamic programming.

The HMM can be thought as a time varying information source which can be modelled as a Markov process with the output probability at each state. Speech data are used to estimate the output probability at each state and the state transition probability of the Markov model.[9] However, it lacks information about how long the speech data can be likely to stay at each state [10][11]. The time duration information is effective in English phoneme recognition and phoneme lattice production for continuous speech. However, the most important point is that the time duration information is essential to produce the PPP.

In the PPP, the Viterbi decoding algorithm is applied to continuous speech at time t and the phoneme probability is calculated by the HMM of the phoneme type p_n over the speech segment ending at time t. The segment starting time s is varied and the time s is determined with the highest probability. In order to reduce the computation time, the starting time is freed in the Viterbi decoding algorithm. This means that the shortest duration is preferred at each state of the HMM. To prevent this situation, short durations and long durations must have imposed a large penalty. This idea leads to time duration modelling in HMM.

IV. HMM with Time Duration Modelling

In a basic HMM, the probability that speech data can be produced from one state decreases exponentially with time t according to the expression of $p(1-p)^t$, where the p is a transition probability from the state to the next state. This does not represent the ture property of the state because the short duration is always preferred. There should be a more representative duration time, preventing short durations as well as long durations at the state. Such a form of the time duration probability can be modelled by Gaussian distribution.

The conventional discrete HMM, whose output probability is computed on vector quantised codewords, can be extended to use time duration probabilities which are also represented in discrete form. The HMM parameters such as transition probability $\bar{a}_{ij}$, output probability $\bar{b}_j(k)$, and time duration probability $\bar{d}_j(\tau)$ can be estimated using the following expressions which can be directly obtained from the Baum-Welch learning algorithm.[12]

$$\bar{a}_{ij}=\sum_{t=1}^{T-1}\sum_{\tau}r_{t,\tau}(i,j)/\sum_{t=1}^{T-1}\sum_{\tau}\sum_{j}r_{t,\tau}(i,j) \tag{6}$$

$$\bar{b}_j(k)=\sum_{t-1}^{T}\sum_{\tau}r_{t,\tau}(j)c_{t,\tau}(k)/\sum_{t=1}^{T}\sum_{\tau}r_{t,\tau}(j)\tau \tag{7}$$

$$\bar{d}_j(r)=\sum_{t=1}^{T}r_{t,\tau}(j)/\sum_{t=1}^{T}\sum_{\tau}r_{t,\tau}(j) \tag{8}$$

where $c_{t,\tau}(k)$ is the number of vector quantised output symbols v_k from time $t+1$ to $t+\tau$. The $r_{t,\tau}(i,j)$ is the time duration probability that speech data can be produced from the state s_j for duration τ, after transition from the state s_i to s_j at time t. The $r_{t,\tau}(i,j)$ and $r_{t,\tau}(i)$ are expressed as follows:

$$r_{t,\tau}(i,j)=\alpha_i(i)a_{ij}d_j(\tau)\prod_{l=1}^{\tau}b_j(O_{t+1})\beta_{t+\tau}(j) \tag{9}$$

$$r_{t,\tau}(i)=\sum_j r_{t,\tau}(j,i) \tag{10}$$

In the above expression, $\alpha_t(j)$ and $\beta_t(i)$ are the forward and backward probabilities respectively and expressed as followed:

$$\alpha_t(j)=\sum_{\tau}\sum_{\substack{\text{all } i\\ i\neq j}}\alpha_{t-\tau}(i)a_{ij}d_j(\tau)\prod_{l=1}^{\tau}b_j(O_{t-\tau+l}) \tag{11}$$

$$\beta_t(i)=\sum_{\tau}\sum_{\substack{\text{all } j\\ j\neq i}}a_{ij}d_j(\tau)\prod_{l=1}^{\tau}b_j(O_{t+l})\beta_{t+\tau}(j) \tag{12}$$

In the case where uniform probability distribution is used as the initial time duration probability for $d_j(\tau)$ and the final probability is estimated from expression (8), this results in the Ferguson model.[10] In the case where a Gaussian probability distribution is used, this is termed here as the Gaussian model. The Gaussian model can prevent errors produced by short token durations or long token durations by virtue of its statistical nautre.

V. Experimental Result

5.1 *Phoneme Recognition of Continuous Speech by Viterbi Decoding Algorithm*

As a guide for comparison, phoneme recognition experiments of continuous speech were carried out using basic discrete HMMs with and without time duration modelling. The experimental speech data used here consisted of 98 sentences which were spoken twice by a single speaker: these were hand-labelled to a phoneme level. Half of the sentences were used as training data and the remaining half were used for testing. The conditions of the acoustic analysis, vector quantisation and Hidden Markov Modelling are shown in Table 1. For the phoneme recognition algorithm, used is one-stage Viterbi decoding algorithm which can search for the best phoneme sequence matched to the continuous speech as described in section II.

Table 2 shows the results together with those for no time duration HMM for comparison. In the table, the term *recognition* indicates the ratio of the number of correctly located phoneme segments to the total number of phonemes contained

Table 1 Conditions for acoustic analysis (AA), vector quantisation (VQ) and Hidden Markov Modelling (HMM).

A A	Sampling frequency	16kHz
	High-pass filter	$x_i-0.97x_{i-1}$
	Feature parameter	LPC cepstrum (20th)
	Frame length	20 ms
	Frame shift	5 ms
	Window type	Hamming window
V Q	Codebook size	256 codewords
	Distance measure	Euclidean distance
	Data amount	100 frames/phoneme
H M M	Number of states	3 states
	Learning method	Baum-Welch Algorithm
	Recognition method	Viterbi Algorithm

Table 2 Results of phoneme recognition of continuous speech by time duration modeling (%).

	true	recognition
HMM without time duration	36.4	48.3
HMM with time duration	43.2	56.6

Table 3 Results of Phoneme recognition of continuous speech on PPP (%).

	true	recognition
DP on PPP	43.0	56.9

in the testing data. The *True* is the ratio of the number of correctly located phoneme segments to the total number of phoneme segments (hypothesis) extracted from the testing data. The high value of this *true* indicates that the quality of the phoneme hypothesis is high in the sense of how well the correctly located phoneme segments are included in the hypothesis. The time duration modelling increases the hypothesis quality by 6.8% and the recognition rate by 8.3%.

5.2 *Phoneme Recognition of Continuous Speech on PPP*

The PPP is produced by computing the function $P_s(p_n,t)$ which is the probability of all phoneme types p_n as a function of time t together with the best starting time s. In the computation algorithm, all phoneme probabilities are computed parallelly, indepednently and time synchronously . In order for phoneme recognition of continuous speech, dynamic programming can be applied to the PPP, because phoneme sequence probability is sub-optimised in the recursive expression as shown in EQ. (5). The algorthm of phoneme recognition is summarised as follows:

(1) Compute the PPP using the function of $P_s(p_n,t)$.
(2) Select the best phoneme p_n which maximises $A(s)+P_s(p_n,t)$ at time t. Each phoneme p_n has different starting time s to the ending time t.
(3) Set the phoneme sequence probability $A(t)$ as $A(t)=A(s)+P_s(P_n,t)$.
(4) Repeat (2), (3) unitil final time is encountered.

We applied the above algorithm for phoneme recognition of continuous speech under the same condition as the experiment described in 5.1. The result is shown in Table 3. The hypothesis quality *true* and the hit-rate of the recognised phoneme segments to the true phoneme, *recognition*, are almost same compared with the phoneme recognition result shown in Table 2 obtained by one-stage Viterbi algorithm.

This indicates that the sub-optimisation of phoneme segmentation on PPP is a good approximation to the full optimisation by one-stage Viterbi algorithm. It also indicates that the PPP contains enough information to show clues to phoneme existence and then to produce the phoneme segmentation. It can be said that the PPP serves as phoneme filter to produce the phoneme probability from continuous speech, and works as specialist of phoneme detection.

VI. Conclusion

The necessity for phoneme probability presentation PPP and the method to produce it have been described. The advantage of the PPP is that it is an inter-

mediate presentation which retains enough information for subsequent processing such as phoneme recognition. Another advantage is its time synchronous parallel computation. We have shown experimentally that the phoneme recognition of contonuous speech using dynamic programming on the PPP has almost the same performance as phoneme recognition using the one-stage Viterbi decoding algorithm with full optimisation. Further work will be to show that word spotting or word sequence recognition can be done on the PPP.

REFERENCES

[1] H.S. Thompson, D. McKelvie and F.R. McInnes, "Robust Lexical Access for Continuous Speech using Dynamic Time Warping and Finite-State Transducers," Eurospeech89, pp. 59–62, 1989.

[2] A. Averbuch, et al., "Experiments with the TANGORA 20,000 Word Speech Recognizer," ICASSP87, pp. 701–704, 1987.

[3] Y. L. Chow, et al., "BYBLOS: The BBN Continuous Speech Recognition System," ICASSP87, pp. 89–92, 1987.

[4] K.F. Lee and H.W. Hon, "Large-Vocabulary Speaker-Independent Continuous Speech Recognition using HMM," ICASSP88, pp. 123–126, 1988.

[5] A.M. Sutherland, M. Campbell Y. Ariki and M.A. Jack "OSPREY: A Transputer based Continuous Speech Recognition System," ICASSP90, pp. 949–952, 1990.

[6] H. Sakoe "Two-level DP-matching- a Dynamic Programming based Pattern Matching Algorithm for Connected Word Recognitions," IEEE Trans. Acoust., Speech & Signal Process., ASSP-27, 6, pp. 588–595, 1979.

[7] C.S. Myers and R. Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition," IEEE Trans. Acoust., Speech & Signal Process., ASSP–29, 2, pp. 284–297, 1981.

[8] H. Ney, "The Use of a One-stage Dynamic Programming Algorithm for Connected Word Recognition," IEEE Trans. Acoust., Speech & Signal Process., ASSP–32, 2, pp. 263–271, 1984.

[9] F. Jelinek, "Continuous Speech Recognition by Statistical Methods," Proc. IEEE, Vol. 64, No. 4, pp. 532–556, 1976.

[10] S.E. Levinson, "Continuously Variable Duration Hidden Markov Models for Automatic Speech Recognition," Computer Speech and Language· No. 1, pp. 29–45, 1986.

[11] M.J. Russel and A.E. Cook, "Experimental Evaluation of Duration Modelling Techniques for Automatic Speech Recognition," ICASSP87, pp. 2376–2379, 1987.

[12] Yasuo Ariki, "Effect of Time Duration and Intrinsic Features for English Phoneme Recogni tion," STUDIA PHONOLOGICA, XXIV, pp. 70–82, 1990.