A RECOGNITION METHOD OF CONNECTED SPOKEN WORDS WITH SYNTACTICAL CONSTRAINTS BY AUGMENTED CONTINUOUS DP ALGORITHM

Sei-ichi Nakagawa

Dept. Informa. & Comp. Sci., Toyohashi University of Technology

Toyohashi, 440 Japan

ABSTRACT

The technique of dynamic time warping by using dynamic programming is powerful for isolated word recognition.

An augmented continuous dynamic programming algorithm is proposed for connected spoken word recognition with syntactical constraints. The algorithm is based on the same principle of two level DP and level building DP. Although our algorithm obtains a near optimal solution for the recognition principle based on pattern matching, it is computationaly more efficient than the conventional methods and also does not require many memory storages. Therefore it is useful for connected word recognition with syntactical constraints in a large vocabulary. The amount of computation is almost the same as that for isolated word recognition.

I INTRODUCTION

The technique of dynamic time warping by vising dynamic programming is powerful for isolated word recognition.

Vintsyuk proposed a connected spoken word recognition method based on this principle in 1971 [1]. In 1975, Sakoe also proposed independently another algorithm from a view point of pattern matching (called Two Level DP matching; TLDP)[2].

Recently, Myers and Rabiner have presented a new DTW-based connected word recognition algorithm, called Level Building DP matching (LB)[3]. This algorithm produces the same result as TLDP except for the difference of the matching window. This is less cpmputation, but not a real time oriented algorithm.

Therefore, Sakoe and Watari led a new real time oriented algorithm by changing the computation order of LB (called Clock-Wise DP matching; CWDP) [4], This uses far less computation, but more memory storages.

In last year, the auchor proposed two new algorithms based on the same principle as described above[5]. One is the Constant Time Delay DP matching (CTDP). This algorithm is the extention of TLDP, that is, executes the computation at every constant time delay (W frames) instead of every frame. Thus, the computation of local distance reduces a factor of 4 ~ 6. The another is 0(n) DP matching. The total amount of computation is the same as that of isolated spoken word recognition by DP matching, that is, a factor of R (width of matching window) for TLDP. However, this 0(n) DP matching cannot be applied to connected word recognition with syntactical constraints.

In this paper, we describe a new type method of connected word recognition. Although this method is an approximate solution for the recognition principle based on pattern matching, this requires less computation than others. Therefore it is useful for connected word recognition with syntactical constraints in a large vocabulary.

II FORMALIZATION OF CONNECTED SPOKEN WORD RECOGNITION BY PATTERN MATCHING

2.1 Notation

- n : n-th word
- N : Number of words, vocabulary size.
- X : Number of words in a test pattern.
- J^n : Length (in frames) of the reference pattern for the n-th word.
- R^n : Reference pattern for the n-th word. $R^n = b_1^n b_2^n \ldots b_J^n$
- I : Length of a test pattern.
- T : Test pattern. $T = a_1 a_2 \ldots a_I$
- $d^n(i,j)$: Local distance between a_i and b_j^n
- $D_x(i)$: Minimum cumulative distance, when $a_1 a_2 \ldots a_i$ is matched with any concatenation of x reference patterns.
- $N_x(i)$: Last reference word in the concatenation of references to satisfy $D_x(i)$.
- $B_x(i)$: Beginning point in the test pattern for $N_x(i)$ minus 1.
- $D^n(s:t)$: Minimum cumulative distance between $a_s \ldots a_t$ and the n-th reference pattern $b_1^n b_2^n \ldots b_J^n$
- $D_x^n(i,j)$: $\min_m (D_{x-1}(m) +$ minimum cumulative distance between $a_{m+1} \ldots a_i$ and $b_1^n b_2^n \ldots b_j^n)$

2.2 Formalization of Connected Word Recognition Principle

Let us consider the connected spoken word recognition problem such that it is to find the sequence of reference patterns, $R = R \; R \ldots R$ of length x which best matches (minimum cumulative distance) the test pattern T, over all possible concatenation of x reference patterns. If the number of words in the test pattern is not known, the minimization should be performed over all length x.

The basic procedure for finding R is to solve a time alignment problem between T and R using a dynamic time warping method (DP matching). Therefore we should use an asynmetric DP pass to apply dynamic programming for connected word recognition[2].

Fig.I shows various types of asynmetric DP passes. In this paper, we use the type (a) to

explain simply the algorithm.

The principle described above is defined as follows. Let $d^R(i,j)$ and $u(i)$ represent the local distance between a_i and j-th frame in R (concatenation of references) and a time warping function, respectively. The optimal concatenation of references is $\hat{R}$ such that

$$D(T,R) = \min_{u(i)} \sum_{i=1}^{I} d^R(i,u(i)) \qquad (1)$$

$$\hat{R} = \text{argmin } D(T,R)$$

Where, $0 \leq u(i)-u(i-1) \leq 2$, $u(1)=1$ and $u(I)=J^R$.

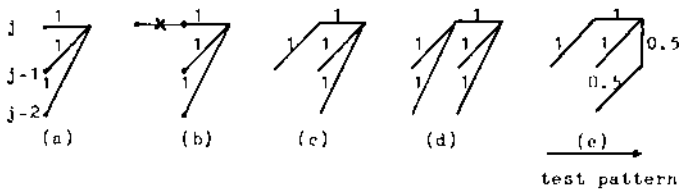


Fig.1 Asymmetric DP pass and weight

2.3 Recursive Formula for Solution

The problem is solved by the following DP equation

$$D_x(0)=0,\ B_x(0)=0 \qquad \text{for } x=1,2,\ \ldots\ X$$

$$D_x(i) = \min_{m,n} (D_{x-1}(m) + D^n(m+1:i)) = \min_{n} D^n_x(i,J^n)$$

$$\text{for } x=1,2,\ \ldots\ X \qquad (2)$$

$N_x(i)=\hat{n}$, $B_x(I)=\hat{m}$, where $\hat{n}$ and $\hat{m}$ satisfy the equation (2).

$D^n(m+1:i)$ is calculated by the following equation.

$$D^n(m+1:i) = \min_{u(k)} \sum_{k=m+1}^{i} d^n(k,u(k)) \qquad (3)$$

Where, $0 \leq u(k)-u(k-1) \leq 2$, $u(m+1)=1$ and $u(i)=J^n$.

After repeating DP equation (2) over $1 \leq i \leq I$, the recognition result is decided by the flaw in Fig.2.

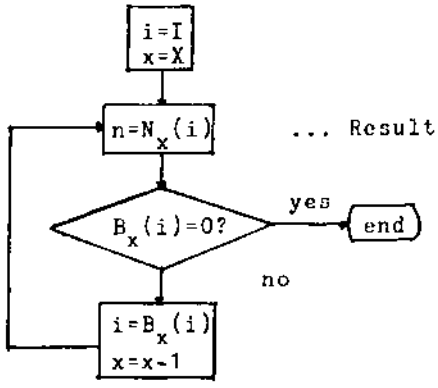


Fig.2 Decision process of recognition result (known length of string)

III FORMALIZATION OF CONNECTED SPOKEN WORD RECOGNITION WITH SYNTACTICAL CONSTRAINTS

3.1 Notation

As a syntax control mechanism, the following finite state automaton α is introduced.

$\alpha = \langle S, \Sigma, \Delta, q_0, F \rangle$

S : A set of states, $\{q_0, q_1, \ldots q_{|S|-1}\}$

Σ : A set of input words, $\{n\}$

Δ : A set of state transition rules, $S \times \Sigma \to S$, that is, $\{\Delta(q_i,n)=q_j\}$

q_0 : An initial state

F : A set of final states, $F \subset S$

$D_q(i)$: Minimum cumulative distance, when $a_1\ a_2\ \ldots\ a_i$ is matched with any concatenation of various reference patterns which reaches at state q.

$N_q(i)$: Last reference word in the concatenation of references to satisfy $D_q(i)$.

$B_q(i)$: Beginning point in the test pattern for $N_q(i)$ minus 1.

$Q_q(i)$: State which satisfies $D_q(i)$ by state transition to q, that is, $\Delta(Q_q(i), N_q(i)) = q$

3.2 Formalization of Connected Word Recognition Principle with Syntactical Constraints

Let a test pattern T be a spoken sentence produced by the regular grammar equivalent to finite state automaton α. A syntax control is introduced into the minimization problem (equation (1)), so that the word sequence {n(1) n(2) ... n(x)} is accepted by automaton α.

A sentence {n(1) n(2) ... n(x)} is said to be accepted by α if $\Delta(q_0,n(1))=q_i \subset S$, $\Delta(q_i,n(2))=q_j \subset S$, ... $\Delta(q_k,n(x-1))=q_g \subset S$, $\Delta(q_g,n(x))=q_f \subset F$. In other words, it is required that an over all state transition sequence caused by input word sequence {n(1), ... n(x)} should start at an initial state q_0 and terminate at a final state in F.

3.3 Recursive Formula for Solution

In order to realize this control, the DP equation (2) is modified as follows. The role of the index in the equation was no more than to count the word number in the word sequence. Equation (2) is searched for the optimal input word n and optimal time transition m→i along with counting up the number x by one. Thus, if we reread (or rewrite) x to q, we can search for the optimal input word n and optimal time transitio m→i along optimal state transition $q_k \to q$. The formula are given in the following.

$$D_{q_0}(0)=0,\quad B_{q_0}(0)=0$$

$$D_q(i) = \min_{n,m,q_k} (D_{q_k}(m) + D^n(m+1:i)) \text{ for all states such that } q=\Delta(q_k,n) \qquad (4)$$

$$N_q(i) = \hat{n},\quad B_q(i) = \hat{m},\quad Q_q(i) = \hat{q}_k$$

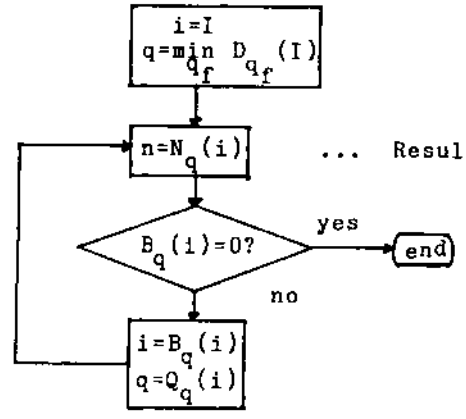


Fig.3 Decision process of recognition result (with syntactic constraints)

After repeating DP equation (4) over $1 \le i \le I$, the recognition result is decided by the flaw in Fig.3.

IV AUGMENTED CONTINUOUS DYNAMIC PROGRAMMING ALGORITHM

4.1 Notation

$\overline{D}^n(i,j)$: min (minimum cumulative distance between $a_m a_{m+1} \ldots a_i$ and $b_1^n b_2^n \ldots b_j^n$), where the base axis for an asymmetric DP pass is the reference' axis (see Fig.4).

$\overline{B}^n(i,j)$: argmin (minimum cumulative distance between $a_m a_{m+1} \ldots a_i$ and $b_1^n b_2^n \ldots b_j^n$)

$\overline{D}^n(i,j,k)$: k-th-min (minimum cumulative distance between $a_m a_{m+1} \ldots a_i$ and $b_1^n b_2^n \ldots b_j^n$)

$\overline{B}^n(i,j,k)$: arg-k-th-min (minimum cumulative distance between $a_m a_{m+1} \ldots a_i$ and $b_1^n b_2^n \ldots b_j^n$). Where arg-k-th-min f(m) means $\hat{m}$ such that $f(\hat{m})$ is the k-th minimum of f(m) for all m.

Let v(k) and Fig.4(a) be a time warping function and the DP pass, respectively. $\overline{D}^n(i,j)$ is defined as follows.

$$\overline{D}^n(i,j) = \min_{v(k)} \sum_{k=1}^{j} d^n(v(k),k)$$

Where, $0 \le v(k)-v(k-1) \le 2$, $1 \le v(1) \le i$ and $v(j)=i$.

$\overline{D}^n(i,j)=\overline{D}^n(i,j,1)$, $\overline{B}^n(i,j)=\overline{B}^n(i,j,1)$

$\overline{D}^n(i,j,1) \le \overline{D}^n(i,j,2) \le \ldots \le \overline{D}^n(i,j,K)$

$\overline{B}^n(i,j,k) \ne \overline{B}^n(i,j,h)$ for $k \ne h$

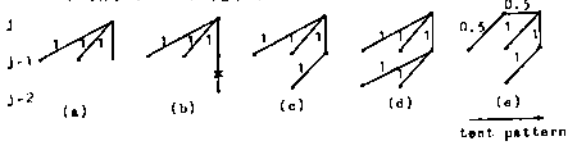


Fig.4 Asymmetric DP pass and weight

4.2 Word Spotting Algorithm

For speech understanding systems, the problem of detecting and locating a specific word in continuous speech has been considered by using a nonlinear time warping procedure (DP matching)[6,7]. This problem is referred to as the word spotting problem.

If we can calculate D (i,J) for any i-th frame in the test pattern and any n-th reference pattern, the word spotting problem would be solved. That is, if $D^n(i,J^n)$ satisfies the threshold for word detection, the location could be regarded as $\overline{B}^n(i,J^n) \sim i$ in the test pattern.

$\overline{D}^n(i,J^n)$ and $\overline{B}^n(i,J^n)$ are calculated as following DP equation.

1) Initialize
   $\overline{D}^n(-1,j)=\overline{D}^n(0,j)=\infty$ for n=1,2, ... N; j=1,2, ..J^n
2) Execute 3)4)5) for i=1,2, ... I
3) Execute 4)5) for n=1,2, ... N
4) $\overline{D}^n(i,1)=d^n(i,1)$
   $\overline{B}^n(i,1)=i$
5) for j=2,3, ... J^n
   $\hat{i} = \underset{i-2 \le i' \le i}{\text{argmin}} \overline{D}^n(i',j-1)$
   $\overline{D}^n(i,j)=\overline{D}^n(\hat{i},j-1) + d^n(i,j)$
   $\overline{B}^n(i,j)=\overline{B}^n(\hat{i},j-1)$

4.3 Application to Connected Word Recognition

Oka has presented a connected word recognition algorithm based on a word spotting algorithm (called Continuous DP)[8], However, his word spotting algorithm is incomplete on the application way of a dynamic programming technique and the word decision process is nondeterministic or heuristic.

We propose a new connected word recognition algorithm based on our word spotting algorithm. Our method is based on the formalization of equations (2), (A), but gives an approximate solution for their equations.

In the equations (2), (A), we must calculate D (m+1:i) for m=0,1,2, ... i-1. This cumulative distance depends on the matching length in the test pattern, that is, i-m. On the other hand, D (i,J) depends on the length of reference, that is, J. Therefore we can obtain approximately D (m+1:i) from D (i,J^n) as follows.

$$D^n(m+1:i) \leftarrow \overline{D}^n(i,J^n) \cdot \frac{i-m}{J^n}$$
$$m+1 = \overline{B}^n(i,J^n)$$

Although the location of $\overline{B}^n(i,J^n) \sim i$ in the test pattern is the most reliable candidate location tor the reference and i-th frame, m Is fixed to B (i,J^n) -1 in this modification. Therefore we try to estimate D^n(m+1:i) in some locations as follows.

$$D^n(m-r_1+1:i) \leftarrow \overline{D}^n(i,J^n) \cdot \frac{i-m+r_1}{J^n}$$

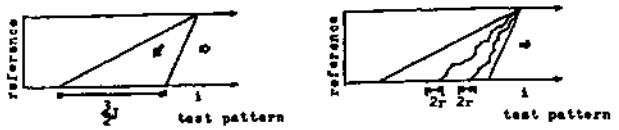
$D^n(m+1:i)$ $\quad D^n(i,J^n)$ $\quad \frac{i-m}{J^n}$

$$D^n(m+r_2+1:i) \leftarrow \overline{D}^n(i,J^n) \cdot \frac{i-m-r_2}{J^n}$$
$$m+1 = \overline{B}^n(i,J^n)$$

Where-r ~r_2 denotes the estimation range. If we would like'"to calculate D^n(m+1:i) for more various locations, we should modify our word spotting algorithm as follows.

4) for k=1,2, ... K
   $\overline{D}^n(i,1,k)=d^n(i,1)$
   $\overline{B}^n(i,1,k)=i$
5) Execute 5') for j=2,3, ... J^n
5') for k=1,2, ... K
   $\hat{i}', \hat{k}' = \underset{i-2 \le i' \le i,\ 1 \le k' \le K}{\text{arg-k-th-min}} \overline{D}^n(i',j-1,k')$
   $\overline{D}^n(i,j,k)=\overline{D}^n(\hat{i}',j-1,\hat{k}') + d^n(i,j)$
   $\overline{B}^n(i,j,k)=\overline{B}^n(\hat{i}',j-1,\hat{k}')$
   Where the following condition should be satisfied:
   $\overline{B}^n(i,j,k) \ne \overline{B}^n(i,j,h)$ for $1 \le h < k$.
   Thus D^n(m+1:i) is obtained from:
6) for k=1,2, ... K
   $m=\overline{B}^n(i,J^n,k)-1$
   $D^n(m-r_1+1:i) \leftarrow \overline{D}^n(i,J^n,k) \cdot \frac{i-m+r_1}{J^n}$
   .
   $D^n(m+1:i) \leftarrow \overline{D}^n(i,J^n,k) \cdot \frac{i-m}{J^n}$
   .
   .
   $D^n(m+r_2+1:i) \leftarrow \overline{D}^n(i,J^n,k) \cdot \frac{i-m-r_2}{J^n}$

Fig.5 illustrates the difference for the calculation of D (m+1:i) between Two Level DP and Augmented Continuous DP. We find ACDP is an approximate method for calculating D (m+1:i), $i-2J^n \le m \le i-J^n/2$. We think, however, it is a fairly good approximation. Where r corresponds to a skipped range between words and r_2 an overlapped range.



(a) Two Level DP (b) Augmented Continuous DP (K=2)

Fig.5 Range of m for the calculation of $D^n(m+1:i)$

4.4 A Connected Spoken Word Recognition Algorithm by Augmented Continuous DP

In this section, we present a connected spoken word recognition algorithm with syntactical constraints by Augmented Continuous DP.

1) Initialize

$D_{q_0}(0)=0, \quad B_{q_0}(0)=0$

$\overline{D}^n(-1,j,k)=\overline{D}^n(0,j,k)=\infty$, for $n=1,2,\ldots N$; $j=1,2,\ldots J^n$; $k=1,2,\ldots K$

2) Execute 3)~10) for $i=1,2,\ldots I$

3) Execute 4)~8) for $n=1,2,\ldots N$

4) $\overline{D}^n(i,1,k)=d^n(i,1)$, $\overline{B}^n(i,1,k)=i$, for $k=1,2,\ldots K$

5) Execute 6) for $j=2,3,\ldots J^n$

6) for $k=1,2,\ldots K$

$\hat{i}', \hat{k}' = \underset{i-2\le i,\,1\le k}{\arg k\text{-th}\min} \overline{D}^n(i',j-1,k')$

$\overline{D}^n(i,j,k)=\overline{D}^n(\hat{i}',j-1,\hat{k}') + d^n(i,j)$

$\overline{B}^n(i,j,k)=\overline{B}^n(\hat{i}',j-1,\hat{k}')$

Where, $\overline{B}^n(i,j,k)<\overline{B}^n(i,j,h)-r_1$ for $1\le h<k$

$\overline{B}^n(i,j,k)>\overline{B}^n(i,j,h)+r_2$

7) Execute 8) for $k=1,2,\ldots K$

8) for $r'=-r_1, -r_1+1, \ldots 0,1, \ldots r_2$

$m=\overline{B}^n(i,J^n,k) - 1$

$D^n(m+r'+1:i)=\overline{D}^n(i,J^n,k)\cdot\frac{i-m-r'}{J^n}$

9) Execute 10) for $q=q_1, q_2, \ldots q_{|S|-1}$

10) for $m'=\underset{1\le k\le K}{\min}\overline{B}^n(i,J^n,k)-r_1-1,\ldots \underset{1\le k\le K}{\max}\overline{B}^n(i,J^n,k)+r_2-1$;

$n=1,2,\ldots N$; $q'=q_1, q_2, \ldots q_{|S|-1}$

$\hat{m}', \hat{n}, \hat{q}' = \operatorname{argmin}(D_q(m') + D^n(m'+1:i))$

Where, $q=\Delta(q',n)$

$D_q(i)=D_{\hat{q}}(\hat{m}') + D^n(\hat{m}'+1:i)$

$N_q(i)=\hat{n}, \quad B_q(i)=\hat{m}', \quad Q_q(i)=\hat{q}'$

11) Word decision process (refer to Fig.3)

V COMPARISON WITH OTHER ALGORITHMS

Table 1 summarizes the amount of computation and memory storages for various algorithms. The amount of computation for phrase level (word decision level) is abbreviated. The amount of memory storages for reference patterns and working area is also abbreviated. In general, notice that the computation time of local distance (distance between frames) consumes about as five or ten times as that of cumulative distance per one execution and that the value of |A| is several times of N (dependent on task). Here we assumed that the window of DP matching was the slope of 1/?. to 2 (for example, (h)~(e) in Fig.2 and Fig.4).

Augmented Continuous DP (ACDP) matching algorithm is a better approximate algorithm of the solution for the equations (2) and (4). In other words, owing to the approximate solution, uncertain parts in word boundaries might be skipped for matching, that is, this has a function of unconstrained endpoints DP matching. This was extended to emmit some (K) candidate locations and matching scores for every word and frame. The suitable value of K is about 2 from view point of computation, memory storage and accuracy. In this case, the value of K' becomes about 2 to 4. We think that ACDP algorithm is a suitable algorithm for a task with large vocabulary or complex syntax.

REFERENCES

[1] T.K.Vintsyuk,"El orient-wise Recognition of Continuous Speech Composed of Words from a Specified Dictionary" Kibernetika (Cybernetics) No.2 (1971) 133-143

[2] H.Sakoe,"Two Level DP-matching - A Dynamic Programming Based Pattern Matching Algorithm for Connected Word Recognition" IEEE Trans. Vol.ASSP-27, No.6 (1979) 588-595

[3] C.S.Myers and L.R.Rabiner,"A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition" IEEE Trans. Vol.ASSP-29, No.2 (1981) 284-297

[4] H.Sakoe and M.Watari,"Clockwise Propagation DP-matching Algorithm for Connected Word Recognition" ASJ, S8]-65 (1981, in Japanese)

[5] S.Nakagawa,"Continuous Speech Recognition Methods by Constant Time Delay DP Matching and 0(n) DP Matching" ASJ, S82-17 (1982,in Japanese)

[6] T.Sakai and S.Nakagawa,"Speech Understanding System - LITHAN - and Some Applications, in Proc. 3rd IJCPR (1976) 621-625

[7] R.W.Christiansen and C.K.Rushforth,"Detecting and Locating Key Words in Continuous Speech Using Linear Predictive Coding" IEEE Trans. Vol.ASSP-25, No.3 (1977) 361-367

[8] R.Oka,"An Infinite-Connected Words Recognition System for Male Speakers Using Time-Space Dynamic Programming" in Proc. 6th IJCAI (1979)

Table 1 Comparison with various DP algorithms for the equation (4)

amount of computation	Two Level DP	Level Building DP	Clock-Wise DP	Constant Time Delay DP	Augmented Continuous DP
local distance	$INJ\cdot\frac{3}{4}J$	$IJ\|\Delta\|$	INJ	$INJ\cdot(1+\frac{3J-4}{4W})$	INJ
cumulative distance	$INJ\cdot\frac{3}{4}J$	$IJ\|\Delta\|$	$IJ\|\Delta\|$	$INJ\cdot\frac{3}{4}J$	$INJ\cdot K'$
memory (word)	$4I\cdot\|S\|$	$4I\cdot\|S\|$	$4I\cdot\|S\|+4J\cdot\|\Delta\|$	$4I\cdot\|S\|$	$4I\cdot\|S\|+4NJ\cdot K$
Note		with no loop		$W \le J_{min}/2$	K=1~3, K'=1~6 better approximation