J. Acoust. Soc. Jpn. (E) 5, 4 (1984)

Matching algorithms between a phonetic lattice and two types of templates— Lattice and graph

PACS number: 43.70.Sc

Y. Kobayashi and Y. Niimi
Department of Computer Science,
Kyoto Institute of Technology,
Matsugasaki, Sakyo-ku, Kyoto, 606 Japan
(Received 18 February 1984)

1. Introduction

In this paper we present two DP matching algorithms used in the speech recognition—a lattice-to-lattice matching and a lattice-to-graph matching. It is assumed that an input pattern be represented by a so-called phonetic lattice in both algorithms and that a reference be represented by a lattice in the former but by a graph in the latter.

The time warping algorithm based on a dynamic programming has enjoyed widespread popularity in the speech recognition, especially in the isolated word recognition. Spectral parameters have been dominantly used in word recognition systems, but storage area and process time increase inhibitive as the vocabulary grows. In continuous speech recognition systems it is impossible to prepare all the parametric variations of possible words, nothing to say of possible sentences. In these situations, phonemic/phonetic alphabets are usually used to represent utterances. However, the segmentation and the labeling errors take place in the acoustic analysis. A lattice is an invention to express such ambiguities where with each element are associated a label, its duration and its confidence score given by the acoustic analyzer.

In accordance, the word templates often contain some secondary phonetic sequences. Such structures are naturally described in lattice templates or graph templates. The matching mechanisms of a linear or graph template against a lattice have been implemented in some continuous speech recognition systems,^{1,2)} but we think they have not well-formalized. An exception is Sakoe's "stack DP matching algorithm"³⁾ that is an elegant formalization for the matching of a linear input pattern against a graph template. Our lattice-to-graph matching algorithm is a generalization of his method.

The DP algorithms presented here are the most

1	2	3	4	5	6	7
E 1	Q 2	P 3	U 4	S 6	N 9	I 11
			N 5	H 7	I	10
				Z 8		

Fig. 1 A phonetic lattice (/ekkusuni/).

fundamental in that both end points are fixed and that the basic DP paths are simple. Extensions of the algorithms are similar to those for sequential (non-lattice) patterns, although several discussions will be left on the relation of the pattern duration to the decision of matching window and the selection of slant-restricted paths.

2. Lattice-to-Lattice Matching

An example of a phonetic lattice is shown in Fig. 1. The horizontal lengths of the lattice elements are not proportional to their real durations. The unit length of the element represents an interval in which combination of lattice elements does not change. We call it a segment. In the figure, segment numbers are above the lattice and serial numbers of lattice elements are inside the boxes. With respect to a segment and an element of the input lattice, we define following symbols:

- I : total number of segments.
- $DI(i)$: duration of the i -th segment.
- L : total number of elements of the input lattice.
- $LAB(l)$: a phonetic label of the l -th element.
- $SCR(l)$: degree of confidence to the label (0~1).
- $DL(l)$: duration.
- $SL(l)$: starting segment number.
- $EL(l)$: end segment number.

For example, $SL(10)=6$ and $EL(10)=7$ for the 10-th element of the lattice in Fig. 1. Symbols for a reference lattice are similarly defined.

- J : total number of segments.
- $DJ(j)$: duration of the j -th segment.
- K : total number of elements of the reference lattice.
- $REF(k)$: a phonetic label of the k -th element.
- $W(k)$: expectation of the phonetic label (0~1).
- $DR(k)$: standard duration of the k -th element.
- $SR(k)$: starting segment number.
- $ER(k)$: end segment number.

$1-W(k)$ can be considered as a penalty of selecting the k -th element of the reference lattice. Following symbols are necessary for the DP mechanism.

$C(A, B)$: confusion matrix defining likelihood between labels used in the input and the reference lat-

Fig. 2 Basic DP paths.

tices. Each element of the matrix takes a value from 0 to 1.

$d(l, k) = 1 - C(LAB(l), REF(k)) * SCR(l) * W(k)$: local distance between the l -th element of the input and the k -th element of the reference.

$G(l, k)$: stack for accumulated distances.

By the benefit of the confusion matrix, the input and the reference alphabets need not be the same and gradual improvements of the acoustic analyzer can be achieved without changing the reference templates.

The initial conditions and the recursive formula for the lattice-to-lattice DP matching are depicted below. Figure 2 (a) shows the basic DP paths for this formula.

$$G(0, 0) = 0,$$

$$G(i, 0) = \infty \quad (i = 1, \dots, I),$$

$$G(0, j) = \infty \quad (j = 1, \dots, J),$$

$$G(i, j) =$$

$$\min \begin{cases} \min_{EL(l)=i} [G(SL(l)-1, j) + DL(l) * \min_{ER(k)=j} d(l, k)] \\ \min_{EL(l)=i, ER(k)=j} [G(SL(l)-1, SR(k)-1) + (DL(l) + DR(k)) * d(l, k)] \\ \min_{ER(k)=j} [G(i, SR(k)-1) + DR(k) * \min_{EL(l)=i} d(l, k)] \end{cases}$$

To be illustrative, the boxes (a), (b), and (c) in Fig. 3 correspond to each line of the formula, respectively—the first represents the horizontal moves on the input, the second the diagonal moves on both patterns and the third the vertical moves on the reference. Range of the innermost minimization operator in the first line of the formula might appear strange because those lattice elements satisfying $ER(k)=j$ could cover several segments. Similar discussion is possible as to the condition $EL(l)=i$ in the third line. However, the horizontal moves correspond to insertions of phonetic labels, and the vertical ones to omissions. So it is not unnatural that we adopt, as a penalty for such an event, the minimum of distances between labels of input and reference lattice elements ending at the i -th and the j -th segment, respectively. The path weights are determined only by length of the input lattice elements in the case of insertions, while by those of the reference lattice elements in the case of omissions.

The formula is efficiently calculated when elements of both lattices are sorted in the ascending order of the end segment numbers. The distance of two lattices $D(LAB, REF)$ is estimated by

$$D(LAB, REF) = G(I, J) \left(\sum_{i=1}^I DI(i) + \sum_{j=1}^J DJ(j) \right).$$

Fig. 3 Lattice-to-lattice matching mechanism.

If it is desired to normalize the distance by the number of segments, it is sufficient only to replace $DL(l)$ and $DR(k)$ with the number of segments which the lattice elements occupy. Then the normalization factor becomes $I+J$. An asymmetric DP algorithm is derived setting $DR(k) = 0$ ($k = 1, \dots, K$).

The matching window will be decided by the accumulated value of the durations on (i, j) plane.

3. Lattice-to-Graph Matching

Sakoe presented a DP matching algorithm between a linear pattern and a loop-free graph in Ref. 3). In this section, the authors generalize the algorithm to the case that the input is a lattice, and augment it so as to restrict warping paths in a window assuming that each branch of a graph have a standard duration. A linear expression of a graph is described with following symbols.

$GRH = B(1)B(2) \dots B(m) \dots B(M)$: graph.

M : the length of graph pattern.

$B(m)$: a phonetic label or a control character.

Control characters

“(” : start character of selective expression,

“/” : separator of alternatives,

“*” : omission of phonetic labels,

“)” : end character of selective expression.

$DB(m)$: standard duration of the label $B(m)$.

$W(m)$: expectation of the label ($0 \sim 1$).

An example of a linear expression of a graph is given in the caption of Fig. 4. Local distance and stacks are defined as follows:

$d(l, m) = 1 - C(LAB(l), B(m)) * SCR(l) * W(m)$: local distance.

$G(i, m)$: stack for accumulated distance.

$PD(p)$: control stack.

In order to restrict the matching window, we introduce a few other symbols.

$AT(i)$: accumulated duration up to the i -th segment of the input lattice.

$TMIN(m)$, $TMAX(m)$: the lower and the upper bounds of standard duration up to the m -th element of the graph, respectively.

F : a factor deciding width of matching window.

ALGORITHM

STEP. 1 (initialization)

$G(0, 0) = TMIN(0) = TMAX(0) = 0$,
 $G(i, 0) = \infty$ ($i = 1, \dots, I$),
 $G(0, m) = \infty$ ($m = 1, \dots, M$),
 $p = 1$, $PD(p) = 0$.

STEP. 2 (loop)

For $m = 1, \dots, M$, execute one of the following STEP. 3 to STEP. 7. On completion go to STEP. 8.

STEP. 3 (when $B(m) = "("$)

$G(i, m) = \infty$ ($i = 1, \dots, I$),
 $TMIN(m) = \infty$, $TMAX(m) = 0$,
 $p = p + 2$, $PD(p) = PD(p - 2)$,
 $PD(p - 1) = m$.

STEP. 4 (when $B(m)$ is a phonetic label)

$TMIN(m) = TMIN(PD(p)) + DB(m)$
 $TMAX(m) = TMAX(PD(p)) + DB(m)$,

For $i = 1, \dots, I$, if either $AT(i) < TMIN(m) * (1 - F)$ or $AT(i) > TMAX(m) * (1 + F)$ holds, then

$G(i, m) = \infty$,

otherwise

$G(i, m) =$

$$\min \begin{cases} \min_{EL(l)=i} [G(SL(l)-1, m) + DL(l) * d(l, m)] \\ \min_{EL(l)=i} [G(SL(l)-1, PD(p)) + DL(l) * d(l, m)] \end{cases}$$

$PD(p) = m$

STEP. 5 (when $B(m) = "/"$)

$TMIN(PD(p - 1))$
 $= \min [TMIN(PD(p)), TMIN(PD(p - 1))]$,
 $TMAX(PD(p - 1))$
 $= \max [TMAX(PD(p)), TMAX(PD(p - 1))]$,
 $G(i, PD(p - 1))$
 $= \min [G(i, PD(p)), G(i, PD(p - 1))]$
 $(i = 1, \dots, I)$,
 $PD(p) = PD(p - 2)$.

STEP. 6 (when $B(m) = "*"$)

Do nothing.

STEP. 7 (when $B(m) = ")"$)

$TMIN(m)$
 $= \min [TMIN(PD(p)), TMIN(PD(p - 1))]$,
 $TMAX(m)$
 $= \max [TMAX(PD(p)), TMAX(PD(p - 1))]$,
 $G(i, m)$
 $= \min [G(i, PD(p)), G(i, PD(p - 1))]$
 $(i = 1, \dots, I)$,
 $p = p - 2$, $PD(p) = m$.

Fig. 4 Lattice-to-graph matching mechanism.
 $GRH = RE(I/*)HE$

STEP. 8 (normalization)

The distance between lattice LAB and graph GRH is given by

$$D(LAB, GRH) = G(I, M) / AT(I)$$

Figure 2(b) shows the basic DP paths taken in STEP. 4. The path corresponding to omission is not included, for omissions are explicitly described in the graph using '*'. As it is impossible to define a unique standard accumulated duration at every node considering the property of a graph template, we calculate a lower estimate $TMIN(m)$ and an upper estimate $TMAX(m)$ and place loose limits upon the matching window for the m -th label of the graph. In the algorithm mentioned above, we have not tried to minimize the size of working area G for simplicity and speed of calculation.

Figure 4 illustrates an example of graph-to-lattice matching. 'min' in the figure designates selection of the smaller values of the two rows. 'BIGNUM' is for initialization of the row with a big number. The row of G which corresponds to a phonetic label contains the accumulated values along the path to each of its elements. The row corresponding to '(' is a temporary store for the minimum accumulated values among alternatives which are finally placed on the row corresponding to ')'. Note that the special symbol '*' is used only for the clarity, one can describe an omission without it.

4. Summary

We have developed two DP matching algorithms between a lattice and two types of reference patterns. Marking plural end segments in a lattice template and using an end-point-free matching algorithm, a loop-free graph and a lattice have almost the same descriptive power.

However, the concept of standard duration can be quite naturally implemented in the lattice. For example, a sequence of phonetic labels /aei/, often observed as an alternative of a sequence /ai/, is expected to have the same duration as the latter has. Construction and maintenance of lattice templates are considered to be easier, or one can even utilize the acoustic outputs themselves as templates. Another advantage of the lattice template is that symmetric basic DP paths can be adopted.

Discussions on the use of these algorithms in a con-

tinuous speech recognition system is given in Ref. 4).

References

- 1) W. A. Woods, "Speech understanding systems—Final report 1974–1976," BBN (1976).
- 2) Nakagawa, "A machine understanding system for spoken Japanese sentences," Doctoral Thesis, Kyoto Univ. (1976).
- 3) H. Sakoe, "A generalization of dynamic programming based pattern matching algorithm—Stack DP-matching—," Trans. of the Committee on Speech Research, Acoust. Soc. Jpn. **S83-23** (1983) (in Japanese).
- 4) Y. Kobayashi, K. Hatano, and Y. Niimi, "Continuous speech recognition based on the phoneme string matching bounded by robust phones," Trans. of the Committee on Speech Research, Acoust. Soc. Jpn. **S83-51** (1983) (in Japanese).