ACKNOWLEDGMENT

The author wishes to express his gratitude for guidance and encouragement received from Prof. T. Iijima of the Tokyo Institute of Technology. He also wishes to express his thanks to Dr. H. Nishino, the Division Chief of the Information Science Division, the Electrotechnical Laboratory, for his assistance in this work, and to Dr. Y. Isomichi, a Senior Scientist of the Electrotechnical Laboratory, for suggesting the method of statistical calculation.

REFERENCES

[1] ISO Recommendation R1831, "Printing specifications for optical character recognition," Ref. No. ISO/R1831-1971(E), Nov. 1971.
[2] T. Iijima and I. Yamasaki, "A method for print quality evaluation of a large number of data," ISO/TC97/SC3/WG1/N173, Feb. 1972.
[3] "Comments by the Japanese member body on the revision of ISO/R1831," ISO/TC97/SC3/N80, Jan. 1973.
[4] J. L. Crawford, "Comments received on document N106, PIDAS print quality analysis by COL grid overlay," ISO/TC97/SC3/N115, Feb. 1974.
[5] M. Bohner, M. Sties, and K. H. Bers, "An automatic measurement device for the evaluation of the print quality of printed characters," *Pattern Recognition*, vol. 9, pp. 11–19, Jan. 1977.
[6] W. R. Throssel and P. R. Fryer, "The measurement of print quality for optical character recognition systems," *Pattern Recognition*, vol. 6, pp. 141–147, 1974.
[7] I. Yamasaki and T. Iijima, "Sampling mechanism of character image," *Trans. Inst. Electronics and Communication Engineers of Japan*, vol. 55-D, pp. 15–22, Jan. 1972; available in English in *Systems·Computers·Controls*, vol. 3, pp. 60–67, Jan.–Feb. 1972.
[8] ——, "A method for print quality evaluation of a large number of data," *J. Information Processing Society of Japan*, vol. 13, pp. 225–231, Apr. 1972; available in English in *Information Processing in Japan*, vol. 12, pp. 119–125, 1972.
[9] ——, "Print quality evaluation of a large number of data," *J. Information Processing Society of Japan*, vol. 13, pp. 525–532, Aug. 1972; available in English in *Information Processing in Japan*, vol. 13, pp. 7–12, 1973.
[10] I. Yamasaki, "A method for two-valuing of printed images," *J. Information Processing Society of Japan*, vol. 16, pp. 419–425, May 1975; available in English in *Information Processing in Japan*, vol. 15, pp. 152–157, 1975.
[11] ——, "A quantitative representation of quality for a set of printed characters," *J. Information Processing Society of Japan*, vol. 18, pp. 253–256, Mar. 1977; available in English in *Information Processing in Japan*, vol. 17, 1978.
[12] H. C. Andrews, "Multidimensional rotations in feature selection," *IEEE Trans. Comput.*, vol. C-20, pp. 1045–1051, Sept. 1971.
[13] "A method for assessing the information value of a character set (the C.O.M. method)," ISO/TC97/SC3/GT1 (ECMA-5)/N38, Dec. 1964.

A Sentence-to-Sentence Clustering Procedure for Pattern Analysis

SHIN-YEE LU, MEMBER, IEEE, AND KING SUN FU, FELLOW, IEEE

Abstract—**Cluster analysis for patterns represented by sentences is investigated. The similarity between patterns is expressed in terms of the distance between their corresponding sentences. A weighted distance between two strings is defined and its probabilistic interpretation given. The class membership of an input pattern (sentence) is determined according to the nearest neighbor or k-nearest neighbor rule. A clustering procedure on a sentence-to-sentence basis is proposed. A set of English characters is used to illustrate the proposed metric and clustering procedure.**

I. INTRODUCTION

IN A PREVIOUS paper [1], we proposed a syntactic clustering procedure, in which each formed cluster is characterized by a pattern grammar. Therefore, the procedure yields not only the clustering results but also a grammar for each cluster. In order to do so, a grammar must be inferred when a new cluster is initiated, and later it is updated whenever an input pattern is added to the same cluster. Error-correcting parsers are employed to measure the distance between an input pattern and the languages generated from the inferred grammars. The input pattern is then classified according to the nearest neighbor syntactic recognition rule. The emphasis of the syntactic clustering procedure is the use of grammar in which the hierarchy of the structure of patterns is described. In this paper, we shall propose a clustering procedure on a pattern-to-pattern basis.

In statistical pattern recognition [9], a pattern is represented by a vector called a feature vector. The similarity between two patterns is often expressed by a metric in the feature space. The selection of features and the metric has a fairly strong influence on the results of cluster analysis [2], [3]. In syntactic pattern recognition [10], a pattern is represented by a linguistic notion called a sentence. The sentence could be a string, a tree, or a graph of pattern primitives and relations. We have proposed the use of a distance between

Manuscript received August 24, 1977; revised December 5, 1977. This work was supported by the AFOSR under Grant 74-2661.

S. Y. Lu was with the School of Electrical Engineering, Purdue University, West Lafayette, IN 47907. She is now with the Department of Electrical and Computer Engineering, Syracuse University, Syracuse, NY.

K. S. Fu is with the School of Electrical Engineering, Purdue University, West Lafayette, IN 47907.

0018-9472/78/0500-0381$00.75

two sentences to express the similarity of their corresponding patterns. The distance between strings and the distance between trees have been studied [4], [5]. A set of English characters is used to illustrate the proposed clustering procedure. In the example, patterns are described by strings in PDL-like representations [6]. It is also our purpose to demonstrate the consistency of the distance defined between sentences and the similarity between the corresponding patterns.

The proposed sentence-to-sentence clustering algorithm is described in Section II. In Section III, the definition of distance between two strings is briefly reviewed and extended to the definition of weighted distance. An algorithm that computes the proposed weighted distance on strings is presented. This algorithm is an extension of Wagner and Fisher's algorithm [4]. A probabilistic interpretation of the weighted metric is also discussed. The stochastic deformation model described in Section III-B could be used as a linguistic decoder for speech recognition or a communication system [11]–[13]. Finally, an illustrative example on clustering is presented in Section IV.

II. A Sentence-to-Sentence Clustering Algorithm

A. A Nearest Neighbor Recognition Rule

Suppose that C_1 and C_2 are two clusters of syntactic patterns. Let patterns in C_1 and C_2 be represented by two sets of sentences $X_1 = \{x_1^1, x_2^1, \cdots, x_{n_1}^1\}$ and $X_2 = \{x_1^2, x_2^2, \cdots, x_{n_2}^2\}$, respectively. For an unknown pattern represented by sentence y, the nearest neighbor recognition rule assigns y to cluster C_1 if

$$\min_j d(x_j^1, y) < \min_l d(x_l^2, y),$$

and assigns y to cluster C_2 if

$$\min_j d(x_j^1, y) > \min_l d(x_l^2, y) \tag{1}$$

where $d(x, y)$ denotes the distance between sentences x and y.

In order to determine $\min_j d(x_j^i, y)$, for some i, the distance between y and every element in the set X_i has to be computed individually. For the case that patterns are represented by strings, the algorithm of computing string-to-string distance is given in Section III.

The nearest neighbor rule can be easily extended to a k-nearest neighbor rule. Let $X_i = \{x_1^i, x_2^i, \cdots, x_{n_i}^i\}$ be a reordered set of X_i such that $d(\tilde{x}_j^i, y) \leq d(\tilde{x}_l^i, y)$ if $j < l$, for all $1 \leq j, l \leq n_i$, then

$$\text{decide } y \in \begin{matrix} C_1 \\ C_2 \end{matrix} \text{ if } \sum_{j=1}^{K} \frac{1}{K} d(\tilde{x}_j^1, y) \lessgtr \sum_{j=1}^{K} \frac{1}{K} d(\tilde{x}_j^2, y). \tag{2}$$

B. The Main Algorithm

We shall describe a clustering procedure in which the classification of an input pattern is based on the nearest (or k-nearest) neighbor rule.

Algorithm 1: Input: A set of samples $X = \{x_1, x_2, \cdots, x_n\}$ and a threshold t. Output: A partition of X into m clusters, $C_1, C_2, \cdots, C_m$.

Method:

Step 1: Let $j = 1$, $m = 1$. Assign x_j to C_m.

Step 2: Increase j by one. Compute

$$D_i = \min_l d(x_l^i, x_j), \qquad \text{for all } i,\ 1 \leq i \leq m.$$

If D_k is the minimum among all D_i, and i) $D_k \leq t$, then assign x_j to C_k; or ii) $D_k > t$, then initiate a new cluster for x_j, and increase m by one.

Step 3: Repeat Step 2 until every element in X has been assigned to a cluster.

In Algorithm 1, a design parameter t, which has a strong influence on the results of cluster analysis, is required. Usually, the number of formed clusters decreases as threshold t increases. If the total number of clusters is known, t can be adjusted until the same number of clusters is generated.[1]

We refer to Algorithm 1 as clustering based on the nearest neighbor rule. When the D_i in Algorithm 1 is the average distance between x_j and the k nearest sentences in C_i, we call it the clustering based on the k-nearest neighbor rule.

III. Distance on Strings

In this section, Aho and Peterson's notion that defines a distance between two strings in terms of language transformations is briefly reviewed [7]. We shall extend the definition to that of a weighted metric. A probabilistic interpretation of the proposed weighted metric is discussed in Section III-B.

Definition 1: For two strings, x and y in Σ^*, we can define an *error transformation* T: $\Sigma^* \to \Sigma^*$ such that $y \in T(x)$. The following three error transformations are introduced.

1) *Substitution transformation:*

$$\omega_1\, a\, \omega_2 \vdash\!\!\xrightarrow{T_S} \omega_1\, b\, \omega_2, \qquad \text{for all } a, b \in \Sigma.$$

2) *Deletion transformation:*

$$\omega_1\, a\, \omega_2 \vdash\!\!\xrightarrow{T_D} \omega_1 \omega_2, \qquad \text{for all } a \in \Sigma.$$

3) *Insertion transformation:*

$$\omega_1 \omega_2 \vdash\!\!\xrightarrow{T_I} \omega_1\, a\, \omega_2, \qquad \text{for all } a \in \Sigma$$

where $\omega_1, \omega_2 \in \Sigma^*$.

Definition 2: The *distance* between two strings $x, y \in \Sigma^*$, $d(x, y)$, is defined as the smallest number of error transformations required to derive y from x.

A. A Weighted Metric

The metric defined in Definition 2 yields exactly the Levenshtein distance between two strings [8]. A weighted Levenshtein distance can be defined by assigning three nonnegative numbers to transformations T_S, T_D, and T_I, respectively. We have proposed a weighted distance that would reflect the difference of the same type of error transformation made on different terminals.

Definition 3: Let the *weights* associated with transforma-

[1] Of course, there is still a possibility that the total number of clusters is correct, but the grouping of patterns in each cluster is not.

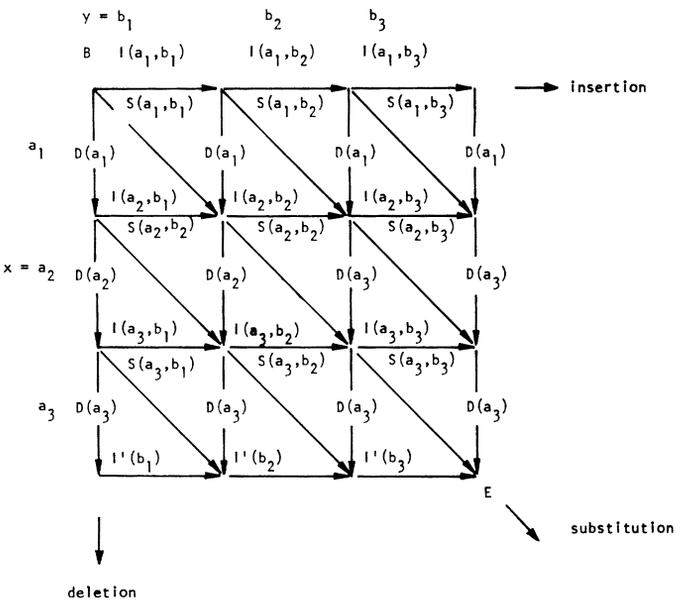


Fig. 1. Graphic interpretation of metric W.

tions on terminal a in a string $\omega_1\ a\ \omega_2$, where $a \in \Sigma$, ω_1 and $\omega_2 \in \Sigma^*$, and Σ is a set of terminals, be defined as follows:

1) $$\omega_2\ a\ \omega_2 \vdash^{T_S, S(a,b)} \omega_1\ b\ \omega_2,$$

where $S(a, b)$ is the cost of substituting a for b;

2) $$\omega_1\ a\ \omega_2 \vdash^{T_D, D(a)} \omega_1 \omega_2,$$

where $D(a)$ is the cost of deleting a;

3) $$\omega_1\ a\ \omega_2 \vdash^{T_I, I(a,b)} \omega_1\ b\ a\ \omega_2,$$

where $I(a, b)$ is the cost of inserting b in front of a;

4) $$X \vdash^{T_I, I(b)} X\ b,$$

where $I'(b)$ is the cost of inserting b at the end of a string.

Definition 4: Let x and y be two strings, and J be a sequence of error transformations used to derive y from x. Let $|J|$ be defined as the sum of the weights associated with the transformations in J. Then the *weighted distance* between x and y is

$$d_\omega(x, y) = \min_J \{|J|\}.$$

Definition 4 can be illustrated graphically. From point B to point E, each path in the lattice shown in Fig. 1 corresponds to a sequence of transformations used to derive y from x. A horizontal branch indicates an insertion transformation, a vertical branch indicates a deletion transformation, and a diagonal branch indicates a substitution transformation. The weight assigned to a particular type of error transformation on a particular symbol in x is labeled at its corresponding branch. Let J be a path in the lattice, and then $|J|$ is the sum of weight associated with each branch in J. $d_\omega(x, y)$ is the weight associated with the minimum-weight path.

The following algorithm, which is an extension of Wagner and Fisher's algorithm, computes the weighted distance between two strings.

Algorithm 2: Input: i) two strings $x = a_1 a_2 \cdots a_n$ and $y = b_1 b_2 \cdots b_m$ where $a_i, b_i \in \Sigma$ for all i. ii) a table of weights associated with transformations on terminals in Σ. Output: the weighted distance $d_\omega(x, y)$.

Method:

Step 1: $\delta(0, 0) = 0$.

Step 2: *DO* $i = 1, n$.

$$\delta(i, 0) = \delta(i - 1, 0) + D(a_i).$$

Step 3: *DO* $j = 1, m$.

$$\delta(0, j) = \delta(0, j - 1) + I(a_1, b_j).$$

Step 4: *DO* $i = 1, n$.

DO $j = 1, m$.

$$\varepsilon_1 = \delta(i - 1, j - 1) + S(a_i, b_j).$$

$$\varepsilon_2 = \delta(i - 1, j) + D(a_i).$$

$$\varepsilon_3 = \delta(i, j - 1) + I(a_{i+1}, b_j) \text{ if } i < n \text{ or}$$

$$\varepsilon_3 = \delta(i, j - 1) + I'(b_j) \text{ if } i = n.$$

$$\delta(i, j') = \min\ (\varepsilon_1, \varepsilon_2, \varepsilon_3).$$

Step 5: $d_\omega(x, y) = \delta(n, m)$ exit.

B. A Probabilistic Interpretation of the Proposed Metric

A stochastic model for the three types of error transformations described in Section III-A has been proposed [15]. We shall briefly review it and then give its interpretation with respect to the weighted metric.

1) A Stochastic Deformation Model: Following the notations introduced in Section III-A, the probabilities associated with the transformations T_S, T_D, and T_I with respect to terminal a are defined to be

1) $q_S(b|a)$, the probability of substituting terminal a by b,
2) $q_D(a)$, the probability of deleting terminal a,
3) $q_I(b|a)$, the probability of inserting b in front of terminal a, and
4) $q_I'(a)$, the probability of inserting terminal a at the end of a string.

The deformation probabilities of a single transformation on terminal a is consistent iff

$$\sum_{b \in \Sigma} q_S(b|a) + \sum_{b \in \Sigma} q_I(b|a) + q_D(a) = 1 \tag{3}$$

where Σ is a set of terminals.

Let $\alpha \in \Sigma^*$ be a substring. The probability that terminal a is transformed to α, denoted $q(\alpha|a)$, is defined as follows:

$$q(\alpha|a) = \begin{cases} q_D(a), & \text{if } \alpha = \lambda \\ \max\{q_S(b|a), q_I(b|a)q_D(a)\}, & \text{if } \alpha = b \\ q_I(b_1|a) \cdots q_I(b_{l-1}|a) \max[q_S(b_l|a), q_I(b_l|a)q_D(a)], & \text{if } \alpha = b_1 b_2, \cdots, b_l,\ l > 1. \end{cases} \tag{4}$$

The consistency of this multiple-transformation model defined in (4) can be proved from (3).[2] Therefore, we have

$$\sum_{\alpha \in \Sigma^*} q(\alpha \mid a) = 1. \tag{5}$$

The probability of inserting $\alpha, \alpha \in \Sigma^*$, at the end of a string is

$$q'(\alpha) = \begin{cases} 1 - q_I', & \text{when } \alpha = \lambda \\ (1 - q_I')q_I'(b_1)q_I'(b_2) \cdots q_I'(b_l), & \text{when } \alpha = b_1 b_2 \cdots b_l,\ l \geq 1 \end{cases} \tag{6}$$

where

$$q_I' = \sum_{a \in \Sigma} q_I'(a).$$

Furthermore,

$$\sum_{\alpha \in \Sigma^*} q'(\alpha) = (1 - q_I') + \sum_i (1 - q_I')q_I'(a_i) + \sum_i \sum_j (1 - q_I') + q_I'(a_i)q_I'(b_j) + \cdots$$

$$= (1 - q_I')(1 + q_I' + q_I'^2 + \cdots) = 1.$$

Assume that a sequence of error transformations made on a terminal in a string is independent from its context. Then for two strings x and y the probability of transforming x to y, where $x = a_1 a_2 \cdots a_n$ is

$$q(y \mid x) = \max_i \left[\prod_{j=1}^{n} q(\alpha_j^i \mid a_j) q'(\alpha_{n+1}^i) \right] \tag{7}$$

where $\alpha_1^i \alpha_2^i \cdots \alpha_n^i \alpha_{n+1}^i$, $|\alpha_j^i| \geq 0$, is a partition of y into $n + 1$ substrings, and $1 \leq i \leq r$. r is the number of different ways of partitioning y into $n + 1$ substrings.

The graph shown in Fig. 1 can also be used to interpret the proposed stochastic model after the label on each branch of the lattice is changed from the weight to the probability of its corresponding error transformation. To derive y from x, the sequence of error transformations that has the highest associated probability is the one defined by (7).

Fung and Fu have defined a distance for $a, b \in \Sigma$, $\delta(a, b)$, as a function of $q(b \mid a)$, the probability of substituting a for b, in their length preserved deformation model [14]. That is,

$$\delta(a, b) = -\log \frac{q(b \mid a)}{q(a \mid a)}. \tag{8}$$

We can interpret the function $\delta(a, b)$ as the weight associated with the substitution of a for b. Using our notation, we have

$$S(a, b) = -\log \frac{q_S(b \mid a)}{q_S(a \mid a)},$$

and the weights associated with deletion and insertion transformations, $D(a)$ and $I(a, b)$, can be defined as $-\log [q_D(a)/q_S(a \mid a)]$ and $-\log q_I(b \mid a)$, respectively. Furthermore, we define

$$I'(b) = -\log q_I'(b).$$

Consequently, we have

$$\begin{aligned} d_\omega(x, y) &= d_\omega(a_1, \alpha_1) + d_\omega(a_2, \alpha_2) + \cdots \\ &\quad + d_\omega(a_n, \alpha_n) + d_\omega(\lambda, \alpha_{n+1}) \\ &= -\log \frac{q(\alpha_1 \mid a_1)}{q_S(a_1 \mid a_1)} - \cdots - \log \frac{q(\alpha_n \mid a_n)}{q_S(a_n \mid a_n)} \\ &\quad - \log \frac{q'(\alpha_{n+1})}{1 - q_I'} \\ &= -\log \frac{q(y \mid x)}{q(x \mid x)} + A \end{aligned} \tag{9}$$

where x, y are two strings, A is a constant, and $\alpha_1, \alpha_2, \cdots, \alpha_n, \alpha_{n+1}$ is a partition of y such that the summation

$$\sum_{i=1}^{n} d_\omega(a_i, \alpha_i) + d_\omega(\lambda, \alpha_{n+1})$$

is the minimum among all the possible partitions of y.

Therefore, the stochastic deformation model could be regarded as a special case of the weighted metric defined in Section III-A. It provides a probabilistic interpretation for the metric.

2) *A Maximum-Likelihood Recognition Rule:*

a) Single-class case: Let $X = \{x_1, x_2, \cdots, x_n\}$ be a sample set, where $x_i \neq x_j$ if $i \neq j$. The rate of occurrence of x_i, $1 \leq i \leq n$, is represented by an *a priori* probability $p(x_i)$. Let y be a string not in X. Using the Bayes rule, the *a posteriori* probability that the true representation of y is x_i is

$$p(x_i \mid y) = \frac{q(y \mid x_i)p(x_i)}{\sum_{j=1}^{n} q(y \mid x_j)p(x_j)}. \tag{10}$$

The maximum-likelihood recognition rule assigns y to be x_i if

$$p(x_i \mid y) = \max_j p(x_j \mid y). \tag{11}$$

b) Multiple-class case: Assume that there are two classes C_1 and C_2 represented by sample sets $X_1 = \{x_1^1, x_2^1, \cdots, x_{n_1}^1\}$ and $X_2 = \{x_1^2, x_2^2, \cdots, x_{n_2}^2\}$, respectively. Each element x_j^l in X_l has an *a priori* probability $p(x_j^l \mid C_l)$. The *a priori* probabilities $P(C_l)$, $l = 1, 2$, are also known. The deformation probability that a string is deformed from some

[2] The proof is given in [15].

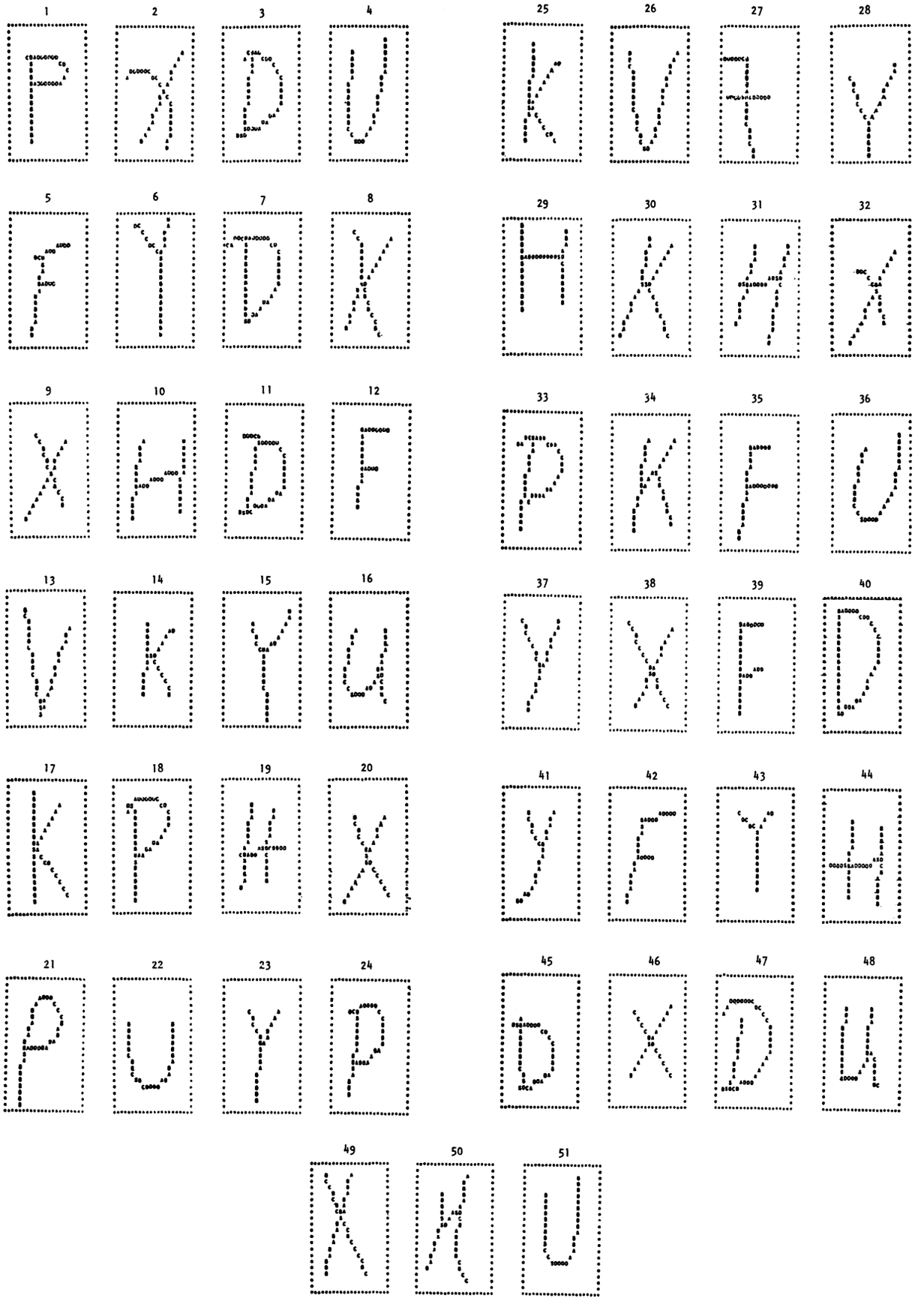


Fig. 2. 51 character patterns.

string in class C_l is $q(y \mid x_i^l, C_l)$. Then the probability that x_i^l is the true representation of y is

$$p(x_i^l, C_l \mid y) = \frac{q(y \mid x_i^l, C_l)p(x_i^l \mid C_l)P(C_l)}{\sum_{k=1}^{2}\left(\sum_{j=1}^{n_k} q(y \mid x_j^k, C_k)p(x_j^k \mid c_k)P(C_k)\right)}. \quad (12)$$

The maximum-likelihood recognition rule assigns y to be x_i^l if

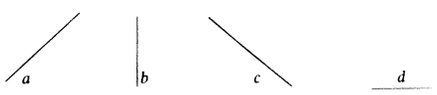
$$p(x_i^l, C_l \mid y) = \max_{k=1}^{2}\left(\max_{j=1}^{n_k} p(x_j^k, C_k \mid y)\right). \quad (13)$$

Again, the maximum-likelihood recognition rule can be considered as a probabilistic interpretation of the proposed nearest neighbor rule.

IV. An Illustrative Example

A set of 51 English character samples is used to illustrate the proposed clustering procedure. All the computations were carried out on a CDC 6500 computer with Fortran IV programming language. The same sample set has been used in a previous paper [1]. The characters are from nine different classes: D, F, H, K, P, U, V, X, and Y. Each character is a line pattern on a 20 × 20 grid. Starting from its lower left corner, each input pattern is initially chain-coded [18] cell by cell. After three consecutive cells have been coded, a pattern primitive of this line segment or branch is extracted.

Four pattern primitives which are line segments with different orientations



are selected. Following Shaw's PDL [6], three concatenation relations, +, ×, *, and the parentheses (and) are used. However, * is used here primarily for the situation of a "self loop," that is, a branch of which the head and tail coincide. The 51 sample patterns and their string representations are given in Fig. 2 and Table I, respectively, where the pattern number indicates the input sequence used in the clustering procedure. For Pattern 1—Character P in Fig. 2, it has three branches. The concatenations of branches are expressed as BRANCH 1 + (BRANCH 2 + BRANCH 3) * and the corresponding string representation of the pattern is $bbb + (b + dddbdd)*$.

A. The Distance

The proposed distance measure is performed on the linguistic representations of patterns (sentences), rather than on the patterns themselves. Consequently, whether or not the model yields a good measure is a matter of choosing appropriate representations. Fig. 3 illustrates a couple of examples where the unweighted distance between the two K's or between the two X's is smaller than that between a K and an X.[3]

[3] In this example, the substitution transformation between a primitive symbol a, b, c, and d, and a relational symbol +, ×, *, (, and) are not considered.

TABLE I
51 Character Patterns and Representations

Pattern No.	String Representation	Pattern No.	String Representation
1	bbb+(b+dddbdd)*	27	bbb+(dx(b+dd))xdd
2	aaa+cddxaaxcbb	28	bb+cbxaaa
3	(bbb+ddcbaa)*	29	bb+bbxddd+axbb
4	cbbbxdabbbb	30	baa+bbxaaxcc
5	bb+(b+dd)xd	31	ba+abxdd+axbbb
6	bbbb+ccxbb	32	aa+ccxaxc
7	(bbb+dxddcbbaa)*	33	bb+(bb+dxddcaad)*
8	ba+bbxaaxcc	34	bb+bbxa+axcb
9	aaa+cxaaxb	35	bb+(bb+d)xdd
10	b+bbxdddd+bbxb	36	cbbxdaabb
11	(bbb+dxddbbad)*	37	ba+bcxaa
12	bb+(bb+dd)xd	38	aa+ccxaaxcc
13	bbbbbxabaa	39	bb+(bb+dd)xdd
14	bb+bbxaaxcc	40	(bbbbb+dddcbaad)*
15	bbb+cxca	41	dab+cbxba
16	cbbxda+bbxb	42	bb+(b+ddd)xd
17	bb+bbbxaaxbcc	43	bbb+cxa
18	bb+(bbb+dddbaa)*	44	b+dxabxdd+bxbb
19	b+bbbxd+bxdxb	45	(bb+ddcbdd)*
20	aa+cbxaaxcc	46	aa+cxaaxcc
21	bbb+(bbadcbad)*	47	bbb+ddaabcddd
22	bbbxddabb	48	bbbxda+bbxb
23	bbb+bcxaa	49	baa+ccxbbxccc
24	b+(bb+ddcbad)*	50	ba+bxa+abxbbb
25	b+bbbxaaxbc	51	cbbxdabbbb
26	cbbbxabbba		

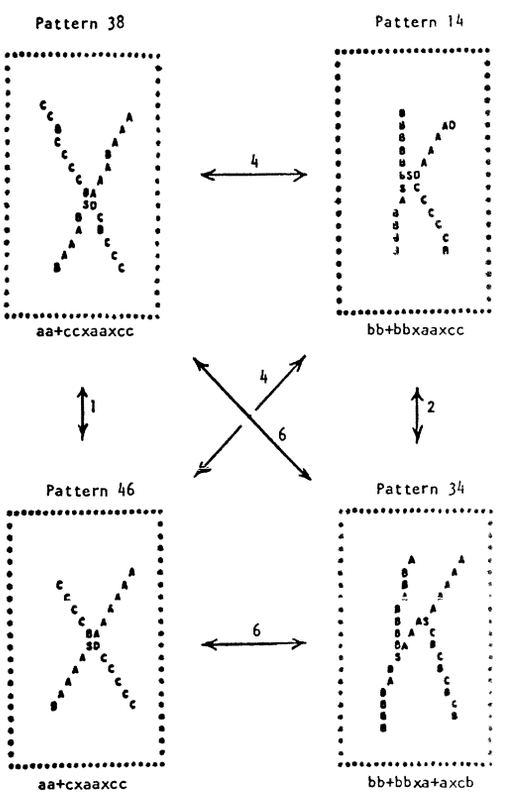


Fig. 3. Distances between similar and dissimilar patterns.

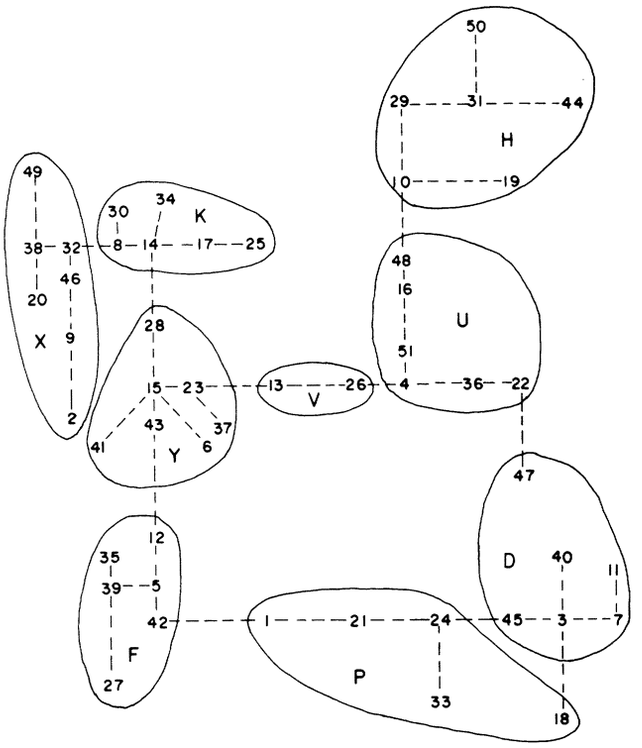


Fig. 4. Minimum spanning tree of 51 character patterns.

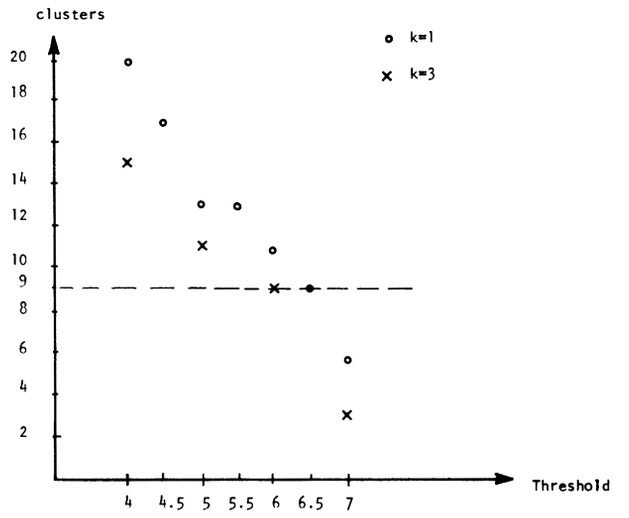


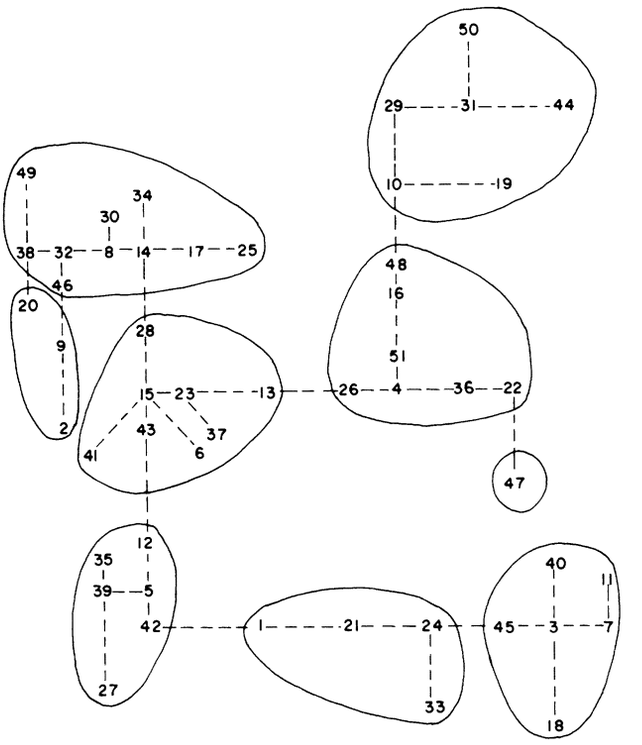
Fig. 5. Number of clusters versus threshold.

B. Minimum Spanning Tree

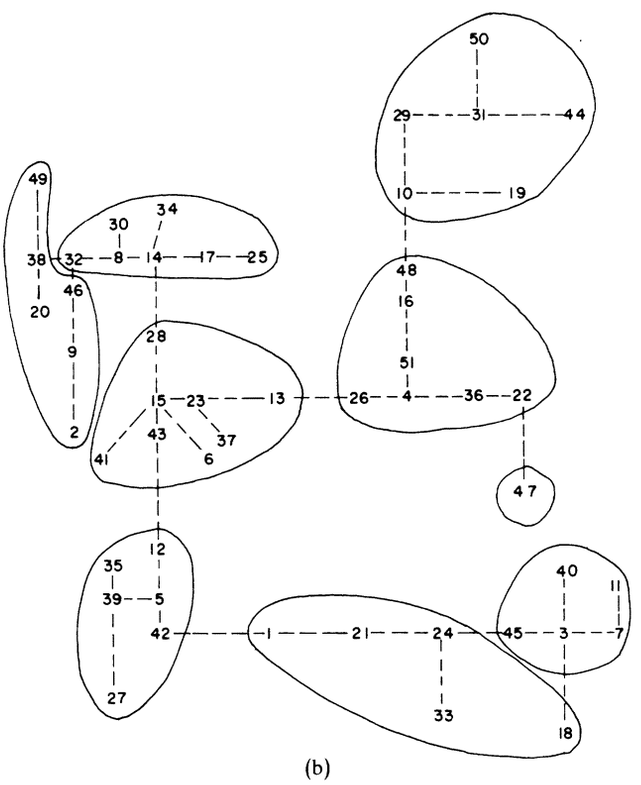
The unweighted distances between every pair of the sample patterns is computed by using Algorithm 2. A minimum spanning tree [3] for the 51 sample patterns can be constructed and is shown in Fig. 4. The true clusters are circled on the tree. We shall use it as a reference for our cluster analysis.

C. Clustering Using Unweighted Distance

According to the clustering procedure described in Section II, the first step is to search for an appropriate threshold value, then the cluster analysis based on this threshold is performed. Fig. 5 shows the relation between the thresholds used and the number of clusters obtained. Here, two experiments are conducted: one is the clustering based on the nearest neighbor rule, and the other is that based on the 3-nearest neighbor rule. The thresholds used in these two experiments are $t = 6$ and $t = 6.5$, respectively. The final clustering results are given in Fig. 6. When the nearest



(a)



(b)

Fig. 6. Clustering results of using unweighted distance. (a) Using nearest neighbor rule with threshold 6. (b) Using 3-nearest neighbor rule with threshold 6.5.

TABLE II
WEIGHTS ASSOCIATED WITH SUBSTITUTION TRANSFORMATION WHERE j IS SUBSTITUTED FOR i

S(i,j) (i \ j)	a	b	c	d	x	+	*	(	)
a	0	1.3	2	1	-	-	-	-	-
b	1.8	0	1.2	1	-	-	-	-	-
c	1.2	1.3	0	1	-	-	-	-	-
d	1.0	1.0	1.0	0	-	-	-	-	-
x	-	-	-	-	0	3.0	3.0	3.0	3.0
+	-	-	-	-	3.0	0	3.0	3.0	3.0
*	-	-	-	-	3.0	3.0	0	3.0	3.0
(	-	-	-	-	3.0	3.0	3.0	0	3.0
)	-	-	-	-	3.0	3.0	3.0	3.0	0

TABLE III
WEIGHTS ASSOCIATED WITH DELETION TRANSFORMATIONS

i	D(i)
a	0.6
b	0.7
c	1.3
d	1.2
x	1
+	1.2
*	0.9
(	0.9
)	0.9

TABLE IV
WEIGHTS ASSOCIATED WITH INSERTION TRANSFORMATIONS WHERE j IS INSERTED IN FRONT OF i

I(i,j) (i \ j)	a	b	c	d	x	+	*	(	)
a	0	1.0	1.0	1.0	1.0	1.0	3.0	3.0	3.0
b	1.0	0	1.0	1.0	1.0	2.5	3.0	3.0	3.0
c	1.0	1.0	0	1.0	1.0	1.0	3.0	3.0	3.0
d	1.0	1.0	1.0	0	1.0	1.0	3.0	3.0	3.0
x	1.0	1.0	1.0	1.0	3.0	3.0	3.0	3.0	3.0
+	1.0	1.0	1.0	1.0	3.0	3.0	3.0	3.0	3.0
*	1.0	1.0	1.0	1.0	3.0	3.0	3.0	3.0	3.0
(	2.0	2.0	2.0	1.0	1.0	2.0	2.0	0.5	2.0
)	1.0	1.0	1.0	1.0	2.0	2.0	3.0	3.0	0.5

TABLE V
WEIGHTS ASSOCIATED WITH ERROR TRANSFORMATIONS

j	a	b	c	d	x	+	*	(	)
I'(j)	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0

TABLE VI
RESULTS OF CLUSTERING USING WEIGHTED DISTANCE

Pattern no.	true class	the nearest pattern	distance from the nearest pattern	cluster result
1	P	-	(-)	1
18	P	1	2.7	1
21	P	1	4.2	1
24	P	21	3.0	1
33	P	24	1.7	1
2	X	1	(17.4)	2
9	X	2	4.4	2
20	X	9	2.4	2
32	X	20	1.3	2
38	X	20	0.0	2
46	X	32	0.7	2
49	X	38	3.6	2
3	D	1	(6.3)	3
7	D	3	1.0	3
11	D	7	2.3	3
40	D	3	1.0	3
45	D	3	2.7	3
47	D	45	5.9	3
4	U	2	(9.3)	4
16	U	4	4.0	4
22	U	4	2.7	4
36	U	4	2.1	4
48	U	16	1.3	4
51	U	36	0.6	4
5	F	1	(9.6)	5
12	F	5	0	5
27	F	5	3.0	5
35	F	12	1.2	5
39	F	5	0.0	5
42	F	5	0.0	5
6	Y	5	(7.0)	6
15	Y	6	5.4	6
23	Y	15	1.0	6
28	Y	15	1.7	6
37	Y	23	2.4	6
41	Y	28	3.9	6
43	Y	15	0.6	6
8	K	2	(6.6)	7
14	K	8	0.6	7
17	K	14	1.0	7
25	K	8	1.9	7
30	K	8	0.0	7
34	K	14	2.3	7
10	H	4	(7.1)	8
19	H	10	5.6	8
29	H	19	3.4	8
31	H	29	4.6	8
44	H	31	3.9	8
50	H	31	3.8	8
13	V	4	(6.1)	9
26	V	13	2.5	9

neighbor rule is used, the chain effect results in four X's, pattern 32, 38, 46, and 49, being clustered together with the K's. This situation is improved when the 3-nearest neighbor rule is used. However, both experiments failed to assign the two V's, pattern 13 and 26, in the same cluster, or to assign the distorted D, pattern 47, together with other D's.

D. Clustering Using a Weighted Distance

The result of clustering presented in Section IV-C can be improved by using a weighted distance. In this example, it is possible to find a set of weights that can correctly cluster all the patterns in the sample set. Tables II–V suggest such a set of weights. The result of cluster analysis is given in Table VI, where the numbers in the first column indicated the input sequences. The true character of each pattern is given in the second column. The nearest pattern for an input pattern is obtained by computing the distances between the input and all the preceding patterns, respectively, and picking the smallest. The actual distance between an input and its nearest pattern is listed in column four. If the distance is greater than the threshold, which is six in this experiment, a new cluster is initiated. The final result is shown in column five.

V. Summary and Remark

Cluster analysis for patterns represented by sentences is studied. Different from statistical pattern recognition, where a cluster analysis is performed on a set of vectors, a set of sentences is to be analyzed. However, once the sentences are extracted from patterns and the distance on sentences is defined, any conventional clustering criterion can be used. The proposed clustering procedure, though only applied to an example of patterns represented by strings, can be easily extended to that of patterns represented by trees using the tree distance defined in [5].

The use of a weighted distance can improve clustering results. In the example given in Section IV, the set of weights used is determined by a trial and error procedure. The computer time required for the cluster analysis is 46 s. To complete the analysis, it is necessary to formulate an algorithm that searches for a set of suitable weights from a set of training samples. The stochastic deformation model provides a theoretical basis for assigning weights as a function of error frequency when a large data base is available.

References

[1] K. S. Fu and S. Y. Lu, "A clustering procedure for syntactic patterns," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, Oct. 1977.

[2] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1972.

[3] E. Diday and J. C. Simon, "Clustering analysis," *Digital Pattern Recognition*, K. S. Fu, Ed. New York: Springer-Verlag, 1976.

[4] R. A. Wagner and M. J. Fisher, "The string to string correction problem," *J. Ass. Comput. Mach.*, vol. 21, Jan. 1974.

[5] S. Y. Lu and K. S. Fu, "Error-correcting tree automata for syntactic pattern recognition," in *Proc. IEEE Computer Society Conf. on Pattern Recognition and Image Processing*, June 6–8, 1977, RPI, Troy, NY.

[6] A. C. Shaw, "A formal picture description scheme as a basis for picture processing systems," *Inform. Contr.*, vol. 14, 1969.

[7] A. V. Aho and T. G. Peterson, "A minimum distance error-correcting parser for context-free languages," *SIAM J. Comput.*, vol. 4, Dec. 1972.

[8] A. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Sov. Phy. Dokl.*, vol. 10, pp. 707–710, Feb. 1966.

[9] H. C. Andrews, *Introduction to Mathematical Techniques in Pattern Recognition.* New York: Wiley, 1972.

[10] K. S. Fu, *Syntactic Methods in Pattern Recognition.* New York: Academic, 1974.

[11] L. R. Bahl and G. F. Jelinek, "Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition," *IEEE Trans. Inform. Theory*, vol. IT-21, July 1975.

[12] L. W. Fung and K. S. Fu, "Maximum-likelihood syntactic decoding," *IEEE Trans. Inform. Theory*, vol. IT-21, July 1975.

[13] R. L. Kashyap and M. C. Mittal, "A new method for error correction in strings with applications to spoken word recognition," in *Proc. IEEE Conf. Pattern Recognition and Image Processing*, June 6–8, 1977, Troy, NY.

[14] L. W. Fung and K. S. Fu, "Stochastic syntactic decoding for pattern classification," *IEEE Trans. Comput.*, vol. C-24, June 1976.

[15] S. Y. Lu and K. S. Fu, "Stochastic error-correcting syntax analysis for recognition of noisy pattern," *IEEE Trans. Comput.*, vol. C-26, Dec. 1977.

[16] E. Tanaka and T. Kasai, "Synchronization and substitution error-correction codes for the Levenshtein metric," *IEEE Trans. Inform. Theory*, vol. IT-22, Mar. 1976.

[17] V. A. Kovalevsky, "Sequential optimization in pattern recognition and pattern description," in *Proc. IFIP Congress 68*, North-Holland Publ. Co., Amsterdam, 1968.

[18] H. Freeman, "On the encoding of arbitrary geometric configuration," *IEEE Trans. Electron. Comput.*, vol. EC-10, pp. 260–268, 1961.