# Correspondence

## A Model for Pattern Recognition Systems with Binary Pattern Vectors

W. G. S. BROWN AND E. A. PARRISH, JR.

### INTRODUCTION

Very often in pattern-recognition applications, the pattern or feature vector is constrained to be binary for practical reasons. In this case, the only possible difference between an unknown pattern and some reference pattern is for one to have "1"s where the other has "0"s.

In the case of many-valued feature vectors, a popular model is

$$X_i = S_i + N \tag{1}$$

where $X_i$ is the $i$th feature vector, $S_i$ is the $i$th deterministic prototype vector, and $N$ is a noise vector that causes $X_i$ to differ from the corresponding $S_i$. In the case of binary-valued feature vectors, the addition indicated in (1) must be modulo 2 and the model must be altered accordingly. Dropping subscripts for clarity,

$$X = S \oplus N. \tag{2}$$

### EXAMPLE

Let $N$ be a random binary vector and $S$ a deterministic prototype vector. It is desired to determine $S$ from the observed random measurement vector $X$. Towards this end, the following probabilities are useful:

$$P(x_j = 1 \mid s_j = 0) = P(x_j = 0 \mid s_j = 1) = P(n_j = 1) = k \tag{3}$$

$$P(x_j = 1 \mid s_j = 1) = P(x_j = 0 \mid s_j = 0) = P(n_j = 0) = 1 - k \tag{4}$$

where $x_j$, $s_j$, and $n_j$ are the $j$th components of $X$, $S$, and $N$, respectively. Since $s_j$ is binary, $E(x_j \mid s_j)$ is easily found to be

$$E(x_j \mid s_j) = (1 - k)s_j + (1 - s_j)k. \tag{5}$$

If $s_j = 1$, then the presence of noise causes $x_j = 0$, and conversely. Without noise the observed feature vector is the same as the prototype vector. The model verifies this behavior, since

$$\lim_{k \to 0} E(x_j \mid s_j) = s_j \tag{6}$$

and

$$\lim_{k \to 1} E(x_j \mid s_j) = \bar{s}_j. \tag{7}$$

Without knowledge of $k$, it is impossible to obtain quantitative results. However, the model gives considerable insight into the qualitative behavior of observed data. The ensemble average of observed data tend to cluster about the values $k$ and $(1-k)$. Knowing this behavior, the investigator knows what to expect and how to interpret data taken from samples. This clustering has, in fact, been observed by several investigators [1]–[4], and a value of $k$ subjectively specified to use as a threshold in determining prototype feature vectors.

### REFERENCES

[1] C. N. Liu, "A programmed algorithm for designing multifont character recognition logic," *IEEE Trans. Electron. Comput.*, vol. EC-13, pp. 586–593, Oct. 1964.

[2] C. N. Liu and G. L. Shelton, Jr., "An experimental investigation of a mixed-font print recognition system," *IEEE Trans. Electron. Comput.*, vol. EC-15, pp. 916–925, Dec. 1966.

[3] R. M. Bowman, "On $N$-tuple optimization and *a priori* error specification for minimum distance pattern classifiers," Ph.D. dissertation, Univ. Virginia, Charlottesville, June 1970.

[4] R. M. Bowman and E. S. McVey, "Calculation of multi-category minimum distance classifier recognition error for binomial measurement distribution," submitted to *IEEE Trans. Comput.*

Manuscript received February 25, 1971; revised August 5, 1971. The research on which this correspondence is based was supported by the Research Institute of the U. S. Army Engineering Topographic Laboratories under Contract DAAK02-70-C-0280.

W. G. S. Brown is with Applied Systems Technology, Springfield, Va.

E. A. Parrish, Jr., is with the Department of Electrical Engineering, University of Virginia, Charlottesville, Va.

## Transference of Learning Between Recognition Classes

J. R. ULLMANN

**Abstract—To demonstrate transference, experiments have been carried out in which the recognition of Highleyman's handprinted numerals was a few percent more accurate when numerals and letters were used as a training set than when only numerals were used as a training set.**

**Index Terms—Handprinted character recognition, learning machines, $n$-tuple methods, pattern recognition.**

If, for example, a training set of the letters $A, \cdots, Z$ is used to improve the recognition of the numerals $0, \cdots, 9$, this exemplifies *transference* of learning. We mean by transference the use of a training set of a recognition class to facilitate the recognition of patterns belonging to a different recognition class. In conventional learning systems the training set has sometimes been found to be too small. By using transference it may be possible to effect a virtual increase in training set size.

To introduce a means of achieving transference, let us consider a very simple character recognition system in which one specimen member of each recognition class is stored as a reference pattern. Let $X_r$ be the specimen member of the $r$th recognition class. An unknown pattern $X$ is written on a rubber sheet which is subject to each of the distortions $D_1, D_2, \cdots, D_i, \cdots, D_\delta$ in turn. When the rubber sheet is subjected to the distortion $D_i$, let the resulting pattern on the rubber sheet be $D_i(X)$. The recognition rule in this simple system is as follows.

*Rule 1:* Assign $X$ to the $r$th recognition class is, for any $i = 1, \cdots, \delta$, $D_i(X) = X_r$.

Let us stipulate that $\{D_1, \cdots, D_\delta\}$ is the set of all possible distortions such that Rule 1 never assigns an input pattern to the wrong recognition class. It is important that the same distortions $D_1, \cdots, D_\delta$ are applied to any input pattern $X$, whatever the recognition class of $X$. In this sense the distortions are independent of recognition class: the distortions of the rubber sheet are independent of what is written on the rubber sheet.

A machine that, given a training set of patterns, can automatically determine whether or not any given distortion is one of the distortions $D_1, \cdots, D_\delta$ is a machine that *learns* distortions. A machine that learns distortions may exhibit transference because, since the distortions are independent of recognition class, it does not matter to which class the training-set patterns belong. For instance, the distortions might be learned from a training set of letters $A, \cdots, Z$ and then used in the recognition of the numerals $0, \cdots, 9$.

We do not suggest that handprinted characters are *only* subject to rubber sheet distortions. For introductory pruposes we mention a rubber sheet only because it is easy to visualize.

We have not in fact attempted to design a machine that learns the distortions $D_1, \cdots, D_\delta$ because the number $\delta$ is presumably so large that it would be economically prohibitive to test whether $D_i(X) = X_r$ for each $i = 1, \cdots, \delta$ in turn. Instead we have investigated a more economical but less reliable recognition system that behaves very roughly as if it learns the distortions $D_1, \cdots, D_\delta$.

Let $S$ be a rectangular array of $N$ bit locations: an element of $S$ is a location that may contain either "1" or "0." The pattern on the rubber sheet is binarized onto $S$ so that the pattern on $S$ is a binarized version of the pattern on the rubber sheet. The array $S$ is always the same whatever the distortion of the rubber sheet, but the pattern on $S$ generally differs with distortion of the rubber sheet.

The distortion $D_i$ shifts the point $(u, v)$ on the undistorted rubber sheet to the point $(u', v')$ on the distorted rubber sheet, and we say that $(u', v')$ *corresponds* to $(u, v)$ in the distortion $D_i$. Because of the coarse-

Manuscript received January 15, 1971; revised September 10, 1971.

The author is with the Division of Computer Science, National Physical Laboratory, Teddington, Middlesex, England.

ness of the grid of $S$, the point $(u, v)$ corresponds only very roughly to an element of $S$. For present introductory purposes we say that if the element $s$ of $S$ corresponds roughly to $(u, v)$ and the element $s'$ of $S$ corresponds roughly to $(u', v')$, then $s'$ *corresponds* to $s$ in $D_i$.

We are concerned with $n$-tuples of elements of $S$, and we denote the $h$th element of the $j$th $n$-tuple by $s_{jh}$. In our notation, $s_{jh}(X) = 1$ or 0 according as the element $s_{jh}$ of $S$ contains 1 or 0 in the binarized pattern $X$ on $S$. $P$ is a set of $n$-tuples of *pairs* of elements of $S$, and the $n$-tuple $\{\{s_{j1}, s_{k1}\}, \cdots, \{s_{jh}, s_{kh}\}, \cdots, \{s_{jn}, s_{kn}\}\}$ belongs to $P$ only if there exists a value of $i$ in the range $i = 1, \cdots, \delta$ such that for all $h = 1, \cdots, n$, $s_{kh}$ corresponds to $s_{jh}$ in $D_i$. $P$ does not necessarily contain all $n$-tuples that satisfy this condition. A member $\{\{s_{j1}, s_{k1}\}, \cdots, \{s_{jh}, s_{kh}\}, \cdots, \{s_{jn}, s_{kn}\}\}$ of $P$ belongs to the subset $P_r$ of $P$ if and only if, for all $h = 1, \cdots, n$, $s_{jh}(X) = s_{kh}(X_r)$. $|P_r|$ is the number of members of $P_r$.

As in Rule 1 let us use one reference pattern per recognition class. Let $X_1, \cdots, X_q, X_r, \cdots, X_z$ be the binarized reference patterns for the $z$ recognition classes. If $n = N$, any member of $P$ can be regarded as a very rough expression of one of the distortions $D_1, \cdots, D_\delta$. If $n = N$ and $X$ does *not* belong to the $r$th recognition class, then $P_r$ is unlikely to have many members. This suggests the rule that follows.

*Rule 2:* Assign $X$ to the $r$th class if $|P_r| > |P_q|$ for all $q \neq r$

will give approximately the same results as Rule 1 if $n = N$. We can expect the substitution error rate of Rule 2 to increase as $n$ decreases below $n = N$. But whatever the value of $n$, the same set $P$ is used in the determination of $|P_r|$ for all $r = 1, \cdots, z$, and in this sense $P$ is independent of recognition class. Therefore, when $P$ is determined automatically from training sets, it does not matter to which recognition classes these training sets belong, and hence the possibility of transference.

To determine a set $P$ we have used a rough heuristic method, and the result only approximates a set $P$ as defined above. To introduce this method, and to avoid questions about the edges of the array $S$, let the array $S$ be folded so as to form a torroidal surface. Let us consider a hypothetical problem in which the recognition class $R_1$ consists of the $N$ $N$-bit patterns on $S$ which are generated by shifting a binarized character "1" into all of the $N$ possible positions on $S$, $R_2$ consists of the $N$ $N$-bit patterns on $S$ that are generated by shifting a binarized numeral "2" into all of the $N$ possible positions on $S$, and so on for $R_3, \cdots, R_r, \cdots, R_z$. Let

$$\{s_{j1}, \cdots, s_{jh}, \cdots, s_{jn}\}$$

and

$$\{s_{k1}, \cdots, s_{kh}, \cdots, s_{kn}\}$$

be two $n$-tuples of members of $S$ such that for all $h = 1, \cdots, n$, the position of $s_{kh}$ differs from the position of $s_{jh}$ by $\alpha$ rows and $\beta$ columns of $S$. A *state* of the $n$-tuple $\{s_{j1}, \cdots, s_{jn}\}$ is a set of $n$ bits located, respectively, in the locations $s_{j1}, \cdots, s_{jn}$. If the $n$-tuple $\{s_{j1}, \cdots, s_{jn}\}$ is in the $i$th of its $2^n$ possible states in the $g$th member, $X_{rg}$, of $R_r$, then $\{s_{k1}, \cdots, s_{kn}\}$ is in the $i$th state in that member of $R_r$ which is the same as a copy of $X_{rg}$ that has been shifted by $\alpha$ rows and $\beta$ columns. Hence it follows that $v_{jri} = v_{kri}$, where $v_{jri}$ is the number of members of $R_r$ in which $\{s_{j1}, \cdots, s_{jn}\}$ is in the $i$th state and $v_{kri}$ is the number of members of $R_r$ in which $s_{k1}, \cdots, s_{kn}$ is in the $i$th state. Since this holds for all $r$, $k$ and since

$$\sum_{r=1}^{z} \sum_{k=1}^{2^n} v_{jri} = zN,$$

it is obvious that

$$\sum_{r=1}^{z} \sum_{k=1}^{2^n} \min (v_{jri}, v_{kri}) = zN. \tag{1}$$

It does not follow rigorously, but it is strongly plausible, that if for *any* two $n$-tuples $\{s_{j1}, \cdots, s_{jn}\}$ and $\{s_{k1}, \cdots, s_{kn}\}$ (1) holds, then there must exist $\alpha$, $\beta$ such that for all $h = 1, \cdots, n$ the position of $s_{kh}$ differs from the position of $s_{jh}$ by $\alpha$ rows and $\beta$ columns of $S$. This suggests the following procedure for automatic determination of $P$: choose pairs of $n$-tuples at random and if for any pair of $n$-tuples $\{s_{ji}, \cdots, s_{jn}\}$, $\{s_{k1}, \cdots, s_{kn}\}$ (1) holds, then assign $\{\{s_{j1}, s_k\}, \cdots, \{s_{jn}, s_{kn}\}\}$ to $P$. This should fill $P$ with $n$-tuples $\{\{s_{j1}, s_{k1}\}, \cdots, \{s_{jn}, s_{kn}\}\}$ such that $\{s_{j1}, \cdots, s_{jn}\}$ differs from $\{s_{k1}, \cdots, s_{kn}\}$ only in position, and this would be appropriate for recognizing members of $R_1, \cdots, R_z$ by means of Rule 2.

For recognition of Highleyman's handprinted characters, we used training sets of Highleyman's characters instead of $R_1, \cdots, R_z$ in the determination of $P$. These training sets consisted of $\rho$ patterns per recognition class and the number of different recognition classes was $\sigma$, so that the total number of training set patterns was $\rho\sigma$. The array $S$ was plane, not torroidal, and was such that Highleyman's data consisted of patterns on $S$. For all $j$, $r$, $i$, $v_{jri}$ was the number of patterns belonging to the $r$th class training set in which the $n$-tuple $\{s_{j1}, \cdots, s_{jn}\}$ was in the $i$th state. To determine the members of a set $P$ we used a trivial elaboration of the procedure: choose pairs of $n$-tuples at random and if for any pair $\{s_{j1}, \cdots, s_{jn}\}$, $\{s_{k1}, \cdots, s_{kn}\}$ it is found that

$$\sum_{r=1}^{\sigma} \sum_{k=1}^{2^n} \min (v_{jri}, v_{kri}) \geqq \frac{\rho\sigma t}{100},$$

then assign $\{\{s_{j1}, s_{k1}\}, \cdots, \{s_{jn}, s_{kn}\}\}$ to $P$. The value of $t$ was made so large that if it had been made appreciably larger, an excessive amount of computer time would have been required for the determination of $P$.

In all of our experiments with Highleyman's data, we used $n = 3$ and the set $P$ had 4000 members. The recognition rule was a little more elaborate than Rule 2, since more than one reference pattern per recognition class was used. Details of experimental procedures and results are given elsewhere [1], and we summarize here only the pair of results that demonstrate transference most clearly.

*Experiment 1:* $P$ was determined using a training set of numerals only. The training set consisted of 12 specimens of each of the ten numerals, and thus $\rho = 12$ and $\sigma = 10$. An elaboration of Rule 2 was then used in the recognition of a test set composed of ten specimens of each of the ten numerals, and 39 percent of the members of the test set were correctly recognized.

*Experiment 2:* $P$ was determined using a training set of numerals $0, \cdots, 9$ and letters $A, \cdots, Z$, except that the letters $O$ and $I$ were omitted. This training set consisted of 12 patterns per recognition class, and thus $\rho = 12$ and $\sigma = 34$. Using the *same* reference patterns, the *same* test set, the *same* value of $t$, and the *same* recognition rule as in Experiment 1, 46 percent of the members of the test set were correctly recognized. Thus the use of letters as well as numerals in the determination of $P$ improved recognition of the numerals by 7 percent, and this constitutes a demonstration of transference.

If Experiments 1 and 2 had not used the *same* reference and test patterns, the 7 percent difference in results would have no significance. In both experiments the 4000 members of $P$ were automatically selected from the same set of candidates, but the actual sets $P$ were different in Experiments 1 and 2. Because 4000 is quite a large number, it seems unlikely that a different *random* selection of the 4000 members of $P$ could cause a 7 percent difference. To remove any possible doubt about this it would have been necessary to repeat Experiments 1 and 2 many times, selecting $P$ from a different set of candidates each time. The cost of this experimentation would have been great and we could not afford it.

## Reference

[1] J. R. Ullmann, "Transference of learning between recognition classes," available from IEEE Computer Society repository.

## Comment on "A Nonlinear Mapping for Data Structure Analysis"

I. WHITE

**Abstract—An alternative metric for use with Sammon's nonlinear mapping is suggested. Rather than Euclidean, the Hamming metric is proposed as a means of reducing the iteration time.**

**Index Terms—Dimensionality, mapping, multivariant data analysis, pattern recognition.**

Manuscript received August 6, 1971.

The author is with the Electronics Research Laboratory, Plessey Company, Havant, England.