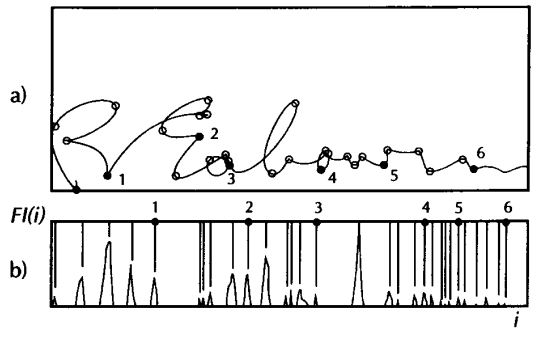

Fig. 6. Signature segmented with function $FI(i)(\theta_{\max} = 3\pi/8, K = 3)$.

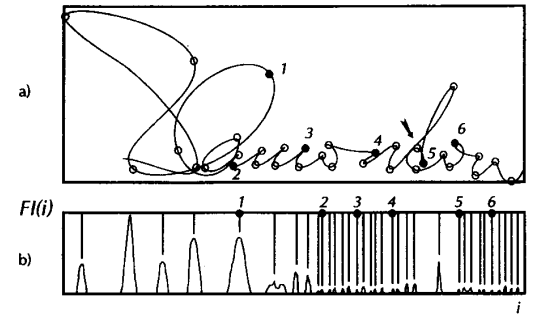

Fig. 7. Signature segmented with function $FI(i)(\theta_{\max} = 3\pi/8, K = 3)$.

the next one after #1) of Fig. 7 as well as the small and acute vertex #4. A drawback of the method (and perhaps of every method?) is its difficulty to segment an almost complete circle, as in the one shown after point #5 in Fig. 7. Indeed, function $FI(i)$ of Fig. 7(b) appears with two local maxima without the function passing through zero. The segmenting point indicated by an arrow on Fig. 7(a) was added manually afterwards for the sake of the discussion. To overcome this problem, it is necessary to diminish the value of $\theta_{\max}$, allowing $FI(i)$ to reach zero between the two peaks.

IV. Conclusion

We have presented an algorithm that makes it possible to estimate the perceptual importance of each of the points of a signature (or other types of continuous cursive handwriting) as a basis for its segmentation. The main idea of the algorithm is that for each point i of the signature, it tries to iteratively construct a vertex centred on that point with the help of neighboring points to either sides of it until certain geometric conditions are met. The method has been applied successfully to a signature database, and the location and relative importance of the segmentation points are generally in agreement with human perception. Moreover, they are also in accordance with our most recent segmentation theory [11]. An interesting application of the algorithm is to use it to quantify one of the difficulties (at the perception level) that could be experienced by a typical imitator in reproducing a signature [2], [4]. This difficulty index, together with an intrapersonal variation index, could be used to identify problematic signers in a particular signature database and adapt the thresholds of the ASV system to improve its overall performance.

One object of our continuing research effort is to implement the algorithm on a neural network and automatically fix the optimal thresholds of the only two parameters of the method.

References

[1] J. -J. Brault and R. Plamondon, "Handwritten curve partitioning based on geometrical and sequential information," in *Proc. Third Int. Symp. Handwriting Comput. Applications* (Montréal), July 1987, pp. 56–58.

[2] J. -J. Brault, "Proposition et vérification d'un coefficient de difficulté d'imitation des signatures manuscrites," Ph.D. thesis, Ecole Polytechnique de Montréal, Canada, 384 pages, Dec. 1988.

[3] J. -J. Brault and R. Plamondon, "How to detect problematic signers for automatic signature verification," in *Proc. 1989 Int. Carnahan Conf. Security Technology* (Zurich), Oct. 1989, pp. 127–132.

[4] ——, "A complexity measure of handwritten curves: Modeling of dynamic signature forgery," to be published in *IEEE Trans. Syst. Man Cybern.*, vol. 23, no. 2, Mar./Apr. 1993.

[5] A. Fishler and R. C. Bolles, "Perceptual organisation and curve partitioning," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, no. 1, pp. 100–105, Jan. 1986.

[6] H. Freeman and L. Davis, "A corner-finding algorithm for chain-coded curves," *IEEE Trans. Comput.*, vol. C-26, pp. 297–303, Mar. 1977.

[7] B. Kruse and C. V. K. Rao, "A matched filtering technique for corner detection," in *Proc. 4th Int. Conf. Patt. Recogn.* , 1978, pp. 642–644.

[8] T. Pavlidis and S. T. Horowitz, "Segmentation of plane curves," *IEEE Trans. Comput.*, vol. C-23, pp. 860–870, Aug. 1974.

[9] R. Plamondon and G. Lorette, "Automatic signature verification and writer authentication: The state of the art," *Patt. Recogn.*, vol 22, no. 2, pp 107–131, 1989.

[10] R. Plamondon, "A theory of rapid movements," in *Tutorials in Motor Behavour II* (G. E. Stelmach and J. Requin, Eds.). Amsterdam: Elsevier, North-Holland, 1992, pp. 55–69.

[11] ——, "A model-based segmentation framework for computer processing of handwriting," in *Proc. 11th Int. Conf. Patt. Recogn.* (The Hague, The Netherlands), Sept. 1992, pp. 303–307.

Fast Nearest-Neighbor Search in Dissimilarity Spaces

András Faragó, Tamás Linder, and Gábor Lugosi

***Abstract*—A fast nearest-neighbor algorithm is presented. It works in general spaces where the known cell (bucketing) techniques cannot be implemented for various reasons, such as the absence of coordinate structure and/or high dimensionality. The central idea has already appeared several times in the literature with extensive computer simulation results. This paper provides an exact probabilistic analysis of this family of algorithms, proving its $O(1)$ asymptotic average complexity measured in the number of dissimilarity calculations.**

***Index Terms*—Average complexity, dissimilarity spaces, fast nearest-neighbor search, pattern recognition, probabilistic analysis of algorithms.**

I. Introduction

Finding a nearest neighbor of a point among several others is a task one often encounters in a number of practical situations such as vector quantization of signals, pattern recognition, etc. In a Euclidean space, this is one of the so-called closest-point problems of computational

Manuscript received June 25, 1991; revised January 6, 1992. Recommended for acceptance by Editor-in-Chief A. K. Jain.

The authors are with the Technical University of Budapest, Budapest, Hungary.

IEEE Log Number 9209991.

geometry, and efficient algorithms are known both in the worst-case sense [1] and expected-time sense [2], [3]. However, there is a demand for nearest-neighbor algorithms that work well under more general conditions. In some practical applications, the underlying metric may be different from the Euclidean metric (possibly the measure of similarity between two points is not even a metric), and/or no direct coordinate structure may be given for the sample space. In trying to find efficient algorithms in these harder situations, several authors seem to have arrived at similar versions of an idea of nearest-neighbor search [4]–[8]. These algorithms make use of some geometric properties induced by the triangle inequality and seem to show the following behavior, which is most explicitly stated in Vidal [5]:

"The algorithm finds the nearest neighbor using an asymptotically constant number of distance calculations on the average."

Moreover, they are easily implementable even in high dimensional spaces, whereas, an optimal algorithm in a Euclidean d space in expected time sense (see, e.g., [3]) becomes impractical very rapidly as the dimension increases, which is typical for the cell technique or "bucketing" methods [9]. Vidal's quoted conclusion was drawn on the basis of extensive computer simulations and subsequently supported by practical experiments [10], but no theoretical justification has been published thus far. In this paper, we make an attempt to grasp the basic idea behind these more general algorithms and carry out an exact probabilistic analysis of the performance in a rather general framework.

To this end, in Section II, we introduce the notion of dissimilarity space, which can be considered to be a generalization of a metric space, give some examples, and describe our algorithm for fast nearest-neighbor search in such spaces. The algorithm has $O(n)$ preprocessing and storage cost, where n is the number of points. In Section III, we introduce a probabilistic model and show that the algorithm performs $O(1)$ dissimilarity calculations on average, that is, it has a constant expected complexity in the number of dissimilarity calculations. In Section IV, some practical remarks and comparisons are made.

II. The Algorithm

The standard problem of nearest-neighbor searching in a Euclidean space is to find, among n points, the nearest to a query point as quickly as possible. In a number of problems, however, we are given n sample points from a more general space (possibly with no direct coordinate structure), and the task is to determine which is closest to the query point in a certain (not necessarily metric) sense. In this case, the most efficient bucketing methods for closest point problems cannot be applied since typically, we can only calculate the "distances" between points. This model applies in all cases when the distance (or dissimilarity) measure is computationally or conceptually complex; thus, a "black box" model for the distance calculation is the only feasible assumption. A relevant practical example is the so-called dynamic time warping (DTW) distance used in speech recognition, when a distance calculation involves a dynamic programming shortest path search in a trellis [10]. DTW is a good example for a "distance" measure that is not a metric, but in some sense, it behaves like a metric. In order to describe problems of this kind, we introduce the notion of *dissimilarity space*, which is, in some sense, a generalization of the concept of metric space.

Definition 1: A nonempty set D with a function $\rho: D \times D \to \mathcal{R}$ is called a dissimilarity space if for any $x, y \in D$ the following conditions are satisfied:

$$\rho(x,y) \geq 0,$$
$$\rho(x,y) = 0 \text{ iff } x = y,$$
$$\rho(x,y) = \rho(y,x).$$

A dissimilarity space in which the triangle inequality holds is a metric space. Just as in metric spaces, a subset H of a dissimilarity space is called *bounded* if $\sup\{\rho(x,y) : x, y \in H\} < \infty$. The notion of the metric is relaxed here, but (obviously), one needs to impose some geometric structure and dimensionality on a dissimilarity space.

Definition 2: Let D be a dissimilarity space, and let $\alpha \geq \beta > 0$. The points $z_1, z_2, \ldots, z_k \in D$ are said to form a basis at level (α, β) for a set $H \subset D$ if for any $x, y \in H$

$$\alpha\rho(x,y) \geq |\rho(x,z_i) - \rho(y,z_i)|, \quad i = 1, \ldots, k \tag{1}$$

and

$$\max_{1 \leq j \leq k} |\rho(x,z_j) - \rho(y,z_j)| \geq \beta\rho(x,y). \tag{2}$$

Moreover, a dissimilarity space D is called *finite dimensional* if there exist $\alpha \geq \beta > 0$ and a positive integer k (depending on D only) such that for any bounded subset $H \subset D$, there are k points in D that form a basis at level (α, β) for H.

Example 1: It is not hard to see that $\mathcal{R}^d$ with the Euclidean metric is a finite (e.g., $d+1$) dimensional dissimilarity space. A possible basis for a bounded set $H \subset \mathcal{R}^d$ is formed, for example, by the vertices of a sufficiently large regular d-dimensional simplex containing H. Elementary geometric calculations show that level values $\alpha = 1$ and $\beta = 1/2$ can be chosen.

Example 2: Let P be a full dimensional bounded polytope in $\mathcal{R}^d$ with vertices $z_1, z_2, \ldots, z_k$. Denote by $\alpha_i(x,y)$ the angle subtended by $\overline{xy}$ at z_i. Set

$$\rho(x,y) = \max_i \alpha_i(x,y).$$

It is left to the reader that the points of P with dissimilarity measure ρ is a finite dimensional dissimilarity space. Another infinite family of examples is given by the following result, which is proven in Appendix A.

Theorem 1: Every finite-dimensional normed vector space is a finite-dimensional dissimilarity space with dissimilarity measure $\rho(x,y) = \|x - y\|$.

The nearest neighbor searching problem in a dissimilarity space is the following: We are given a set of n points $X_1, \ldots, X_n$, which are elements of a bounded set $H \subset D$. A nearest-neighbor algorithm should determine in an efficient way, using some preprocessing of the points, the closest of these points to a new query point X coming from H. Here, closeness means similarity, that is, the nearest neighbor of X is X_i if $\rho(X, X_i) \leq \rho(X, X_j)$, $j = 1, \ldots, n$. The common idea of the (coordinate free) algorithms [4]–[8] is that they restrict the search to some appropriately chosen *neighborhood* of the query point with the following crucial properties:

- The neighborhood is large enough to contain the nearest neighbor with certainty.
- The neighborhood is small enough to ensure that the average number of sample points contained remains asymptotically bounded.
- The neighborhood is defined constructively in terms of distances to points known during the preprocessing stage.

To grasp and analyze this common idea, we describe an algorithm that contains it in a pure form and is isolated from additional factors.

Let D be a finite dimensional dissimilarity space, and let the points $z_1, \ldots, z_k$ form a basis for the bounded set H at level (α, β). Our proposed algorithm is the following.

Algorithm 1: **Preprocessing** Compute and store all the values $\rho(X_i, z_j)$, $i = 1, \ldots, n$; $j = 1, \ldots, k$. As k is fixed, this means $O(n)$ preprocessing time and storage cost.

Nearest neighbor searching

INITIALIZATION: Set $\mathcal{T} \leftarrow \{X_1, \ldots, X_n\}$.

STEP 1: Compute the value of

$$\gamma(X_i) = \max_{j=1,\ldots,k} |\rho(X_i, z_j) - \rho(X, z_j)|$$

for each $X_i \in \mathcal{T}$.

STEP 2: Set $t_0 \leftarrow \min_i \gamma(X_i)$. Delete all the points X_l from $\mathcal{T}$ for which

$$\gamma(X_l) > \frac{\alpha}{\beta} t_0$$

holds.

STEP 3: Find the nearest neighbor of X in the remaining part of $\mathcal{T}$ by exhaustive search:

$$T^{NN} = \arg\min_{U \in \mathcal{T}} \rho(X, U)$$

STOP. T^{NN} is the result.

The next theorem shows the correctness of the algorithm.

Theorem 2: Algorithm 1 always finds the nearest neighbor.

Proof: We have to show that the correct nearest neighbor, which we denote by X_n^{NN}, is never deleted from $\mathcal{T}$ in Step 2. Set

$$X_n^* = \arg\min_{i=1,\ldots,n} \gamma(X_i).$$

In the definition of X_n^{NN} and X_n^*, in case of ambiguity, we choose a random index among the candidates. Suppose that $\gamma(X_n^{NN}) > \frac{\alpha}{\beta}\gamma(X_n^*)$, that is, Step 2 excludes X_n^{NN}. From this, using Definition 2, we have

$$\rho(X, X_n^{NN}) \geq \frac{1}{\alpha}\gamma(X_n^{NN}) > \frac{1}{\beta}\gamma(X_n^*) \geq \rho(X, X_n^*)$$

which is a contradiction. □

It is quite clear that in the worst case, that is, when no exclusion is carried out in Step 2, the algorithm executes n dissimilarity calculations. However, the next section shows that in a rather general probabilistic setup, the average case is substantially different from the worst case. In particular, the number of dissimilarity calculations remains constant on the average as n increases.

In a strict sense, the complexity of the algorithm is not only determined by the number of dissimilarity calculations but also by other computations in Steps 2 and 3. From a practical point of view, however, if $\rho(\cdot,\cdot)$ is a function of high complexity, then the running time of the algorithm is determined essentially by the number of dissimilarity calculations, as is shown by the simulation results cited above. We will address this question in Section IV.

III. Probabilistic Analysis

For the analysis of the average complexity, we have to set up a probabilistic model. Let $(D, \mathcal{S})$ be a measurable space, where the family of sets $\mathcal{S}$ is termed the collection of measurable subsets of D. It is assumed that the measurable sets of the finite dimensional dissimilarity space D include the closed balls $B(x, r) = \{y \in D : \rho(x, y) \leq r\}$ of radius r centered at x for all $r > 0$, $x \in D$. We assume further that $\rho : D \times D \rightarrow \mathcal{R}$ is a Borel measurable function on the product measurable space $(D \times D, \mathcal{S} \times \mathcal{S})$. Note that in the examples mentioned above, these conditions are satisfied.

Let $X, X_1, \ldots, X_n$ be independent identically distributed random elements taking their values from a *bounded* subset H of D. Introduce the notation

$$p(x, r) = P_X(B(x, r)) = \Pr\{X \in B(x, r)\}.$$

We assume that the following regularity condition holds for the common distribution of $X, X_1, \ldots, X_n$.

Condition 1: There exists a $d > 0$ and a function $f : D \rightarrow \mathcal{R}$ such that

$$\lim_{r \rightarrow 0} \frac{p(x, r)}{r^d} = f(x) > 0 \tag{3}$$

uniformly for almost all $x \in D$ (mod P_X).

Remark: Note that for $D \subseteq \mathcal{R}^d$ with an arbitrary norm-based metric and for random variables with density, Condition 1 indicates the uniform convergence in Lebesgue's density theorem (see [11]).

Now, we can state the main result.

Theorem 3 Let F_n be the number of dissimilarity calculations executed by Algorithm 1 for n points. If Condition 1 holds, then

$$\limsup_{n \rightarrow \infty} E(F_n) \leq k + \left(\frac{\alpha}{\beta}\right)^{2d}$$

where $E(\cdot)$ denotes expectation, and k, α, β are as in the description of the algorithm.

Remark: The theorem asserts that $E(F_n) = O(1)$. It follows from Example 1 that in $\mathcal{R}^d$, if the X_i have well-behaved density in the sense of Condition 1, then $\limsup_n E(F_n) \leq d + 1 + 4^d$.

Before proving the theorem rigorously, it is worth mentioning that the main idea is the following: We show that a ball of radius $c\rho(X, X_n^{NN})$ ($c > 0$ fixed) centered at the query point X contains asymptotically only a constant number of sample points on the average. To present the exact proof, we have to explore first some properties of finite dimensional dissimilarity spaces and probability distributions defined on them.

Definition 3: A set $A \subset D$ is called *discrete* if there is a constant $\rho_0 > 0$ such that $x, y \in A$, $x \neq y$ implies $\rho(x, y) \geq \rho_0$.

Lemma 1: Let A be a bounded discrete set in a finite dimensional dissimilarity space D. Then, A is finite.

The proof of the lemma is in Appendix B. The next lemma will be a useful technical tool in the proof of Theorem 3 and is proven in Appendix C.

Lemma 2: Let X be a random element taking its values from a finite dimensional dissimilarity space D. Suppose that $\Pr\{X \in A\} = 1$ for some bounded measurable subset A of D. Then for any fixed $r_1 > 0$ there exists an $\epsilon > 0$ such that

$$\Pr\{p(X, r_1) \geq \epsilon\} = 1.$$

Now we are armed to prove Theorem 3.

Proof of Theorem 3: Since the $\rho(X_i, z_j)$ values are given by the preprocessing, Step 1 of the algorithm requires only k dissimilarity calculations. Thus, it is enough to consider the number of points T_n not deleted from $\mathcal{T}$ in Step 2 for $F_n = k + T_n$. Let X_n^* and X_n^{NN} be as in the proof of Theorem 2. Using Definition 2, for each X_i remaining in $\mathcal{T}$ after Step 2, we have

$$\begin{aligned}\rho(X, X_i) &\leq \frac{1}{\beta}\gamma(X_i) \leq \frac{\alpha}{\beta^2}\gamma(X_n^*) \\ &\leq \frac{\alpha}{\beta^2}\gamma(X_n^{NN}) \leq \frac{\alpha^2}{\beta^2}\rho(X, X_n^{NN}).\end{aligned} \tag{4}$$

Put $c = \frac{\alpha^2}{\beta^2} \geq 1$. Denoting by T_n' the number of X_i with $\rho(X, X_i) \leq c\rho(X, X_n^{NN})$, by (4), we have $T_n \leq T_n'$, and thus, it suffices to show that

$$\lim_{n \rightarrow \infty} E(T_n') = c^d. \tag{5}$$

From now on, in the proof, I_B will denote the indicator of the set B, and the abbreviation $R_n = \rho(X, X_n^{NN})$ will be used. Now, using

the i.i.d. property of $X, X_1, \ldots, X_n$, we can write

$$E(T_n') = E\left(\sum_{i=1}^{n} I_{\{X_i \in B(X, cR_n)\}}\right) = nE\left(I_{\{X_n \in B(X,cR_n)\}}\right)$$
$$= nE\left(I_{\{X_n \in B(X,cR_n)\}} I_{\{X_n = X_n^{NN}\}}\right)$$
$$+ nE\left(I_{\{X_n \in B(X,cR_n)\}} I_{\{X_n \neq X_n^{NN}\}}\right). \quad (6)$$

The first term in (6) is obviously $n\frac{1}{n}$, whereas the second can be written as

$$nE\left(I_{\{X_n \in B(X,cR_{n-1})\}}\right) - nE\left(I_{\{X_n \in B(X,R_{n-1})\}}\right)$$

where the second term is again $n\frac{1}{n}$. Thus, (6) amounts to

$$E(T_n') = nE\left(I_{\{X_n \in B(X,cR_{n-1})\}}\right)$$
$$= nE\left(E[I_{\{X_n \in B(X,cR_{n-1})\}}|X]\right) = nE[p(X, cR_{n-1})]$$

where, in the last step, we used the independence of the X_i. Since $E[p(X, R_{n-1})] = \Pr\{X_n \in B(X, R_{n-1})\} = \Pr\{X_n = X_n^{NN}\} = 1/n$, we conclude that for $r > 0$

$$E(T_{n+1}') = \frac{E[p(X, cR_n)]}{E[p(X, R_n)]}$$
$$= \frac{E[p(X, cR_n) I_{\{R_n \leq r\}}] + E[p(X, cR_n) I_{\{R_n > r\}}]}{E[p(X, R_n) I_{\{R_n \leq r\}}] + E[p(X, R_n) I_{\{R_n > r\}}]}. \quad (7)$$

Now, by Lemma 2, $\epsilon > 0$ can be chosen such that

$$\Pr\{p(X, r) \geq \epsilon\} = 1$$

yielding

$$\Pr\{R_n > r\} = E[(1 - p(X, r))^n] \leq (1 - \epsilon)^n$$

that is, $\Pr\{R_n > r\}$ tends to zero exponentially quickly. Since the second terms in both the numerator and the denominator of (7) are bounded above by this probability and since the denominator is $1/n$ and the numerator is greater, it follows that

$$\lim_{n \to \infty} E(T_{n+1}') = \lim_{n \to \infty} \frac{E[p(X, cR_n) I_{\{R_n \leq r\}}]}{E[p(X, R_n) I_{\{R_n \leq r\}}]} \quad (8)$$

for arbitrary $r > 0$, provided that the limit on the right-hand side exists. However, by the uniform convergence in Condition 1, for any $\epsilon > 0$, an $r > 0$ can be chosen such that the following inequalities hold:

$$\frac{E[(1 - \epsilon) f(X)(cR_n)^d I_{\{R_n \leq r\}}]}{E[(1 + \epsilon) f(X) R_n^d I_{\{R_n \leq r\}}]}$$

$$\leq \frac{E[p(X, cR_n)) I_{\{R_n \leq r\}}]}{E[p(X, R_n)) I_{\{R_n \leq r\}}]}$$

$$\leq \frac{E[(1 + \epsilon) f(X)(cR_n)^d I_{\{R_n \leq r\}}]}{E[(1 - \epsilon) f(X) R_n^d I_{\{R_n \leq r\}}]}.$$

After cancellations, we obtain

$$\frac{1 - \epsilon}{1 + \epsilon} c^d \leq \frac{E[p(X, cR_n)) I_{\{R_n \leq r\}}]}{E[p(X, R_n)) I_{\{R_n \leq r\}}]} \leq \frac{1 + \epsilon}{1 - \epsilon} c^d. \quad (9)$$

Since ϵ is arbitrary, (8) and (9) together imply

$$\lim_{n \to \infty} E(T_n') = c^d,$$

and the proof is completed.

IV. Conclusion

The algorithm and its analysis should be considered to be an attempt to find the mathematical foundations of a family of fast nearest-neighbor algorithms working well *in practice* in high dimensions, under general conditions, using no coordinates of the sample points. As a measure of complexity, the number of dissimilarity ("distance") calculations has been chosen, ignoring all the side computations. This point of view can be defended considering the following facts. First, the practical simulation results in the cited references show that when the dissimilarity measure is of high computational complexity, the running time of the algorithm is essentially determined by the number of dissimilarity computations. Second, the side computations in Step 2 of the algorithm actually mean that one has to execute a full search in a transformed space where any $Y \in D$ is represented by the k-tuple $\tilde{Y} = (\rho(Y, z_1), \ldots, \rho(Y, z_k))$, and the distance is induced by the maximum norm. However, this problem is simpler than the original one, and it is possible to use the existing cell technique solutions of low complexity (for a survey, see [3]). Therefore, the number of dissimilarity calculations represents the *additional* complexity induced by the more general instance. Therefore, the results can be interpreted to mean that finding the nearest neighbor in these more general spaces is theoretically of the same complexity as doing so in Euclidean spaces. On the other hand, the new algorithmic idea is necessary because cell/bucketing methods cannot be implemented efficiently for the general problem.

Vidal *et al.* [10] investigated a version of the algorithm analyzed in this paper that was implemented for an isolated word recognition system of a 200-word vocabulary. Their conclusion was that the number of executed dissimilarity calculations was reduced by 94–96%. Since the DTW dissimilarity measure is rather complex compared with, e.g., the Euclidean metric, this reduction in the number of DTW calculations resulted in a one order of magnitude decrease in the running time.

It is intuitively clear from the analysis that in practice, the algorithm works well if the data is "well clusterable" because Step 2 of the algorithm is likely to delete a large proportion of the points from further investigation. This type of data is typical in pattern recognition tasks. The increased efficiency of a version of the algorithm for well-clusterable data was pointed out in [13].

Appendix A

Proof of Theorem 1: Let S be a finite dimensional normed space with norm $\|\cdot\|$. If we define the operations $c(x, y) = (cx, cy)$ and $(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$ on $S^2 = S \times S$ and introduce the norm $\|(x, y)\| = \|x\| + \|y\|$ on S^2, then again, a finite dimensional normed vector space is obtained. Now, fix a real number $0 < \tau < 1$ and for each $z \in S$, define the set $A_z \subset S^2$ by

$$A_z = \left\{(x, y) \in S^2 : x \neq y, \frac{\|y - z\| - \|x - z\|}{\|x - y\|} > \tau\right\}.$$

Furthermore, let $H \subset S^2$ be the following set:

$$H = \{(x, y) : \|x\| \leq 2, \|y\| \leq 1, \|x - y\| \geq 1/2\}.$$

We will use the following properties of these sets:

i) H is closed and bounded.
ii) A_z is open.
iii) $H \subset \bigcup_{z \in S} A_z$.

Clearly, i) and ii) follow from the definition. To see iii), it is enough to observe that for any $x \neq y \in S$, $(x, y) \in A_x$ holds.

Thus, the sets $\{A_z, z \in S\}$ form an open cover of H. It follows from the Heine–Borel theorem that there exists a *finite* subcover, that

is, there are points $z_1, \ldots, z_k$ with $H \subset \bigcup_{i=1}^{k} A_{z_i}$. This means that for any $(x, y) \in H$

$$\frac{\|y - z_j\| - \|x - z_j\|}{\|x - y\|} > \tau \tag{10}$$

holds for some $1 \leq j \leq k$. Now, take two points $x \neq y \in S$ with $\|x\| \leq 1, \|y\| \leq 1, \|x - y\| = \lambda < 1/2$. Set $x' = \frac{1}{\lambda}(x - (1 - \lambda)y)$. Then, $x = \lambda x' + (1-\lambda)y$ holds, that is, x divides the line segment $\overline{x'y}$ such that $\frac{\|x-y\|}{\|x'-y\|} = \frac{\lambda}{1-\lambda}$. As $\|x-y\| = \lambda$, this implies $\|x'-y\| = 1$, which yields $\|x'\| \leq 2$ for $2 \geq \|x' - y\| + \|y\| \geq \|x'\|$. Collecting these facts, we have $(x', y) \in H$. However, by (10), there is a z_j with

$$\frac{\|y - z_j\| - \|x' - z_j\|}{\|x' - y\|} > \tau. \tag{11}$$

Now, using the convexity of the norm, we can write

$$\begin{aligned} &\frac{\|y - z_j\| - \|x - z_j\|}{\|x - y\|} \\ &\geq \frac{\|y - z_j\| - (\lambda\|x' - z_j\| + (1 - \lambda)\|y - z_j\|)}{\lambda} \\ &= \|y - z_j\| - \|x' - z_j\| \\ &= \frac{\|y - z_j\| - \|x' - z_j\|}{\|x' - y\|} > \tau. \end{aligned} \tag{12}$$

Thus, if $x, y \in B = \{u \in S : \|u\| \leq 1\}$, then there is a $j \in \{1, \ldots, k\}$ such that

$$| \, \|x - z_j\| - \|y - z_j\| \, | \geq \tau\|x - y\| \tag{13}$$

holds. This follows from the definition of H and from (10) and (12). Since the triangle inequality guarantees (1) with $\alpha = 1$ and we have just proved (2) with $\beta = \tau$, we obtain that the points $z_1, \ldots, z_k$ form a basis at level $(\alpha, \beta) = (1, \tau)$ for the closed unit ball in S in the sense of Definition 2.

Now, let A be an arbitrary bounded subset of S with $r = \sup_{x \in A} \|x\|$. Then, $x, y \in A$ implies $\frac{1}{r}x, \frac{1}{r}y \in B$. Then, we have

$$| \, \|x - rz_j\| - \|y - rz_j\| \, | \geq \tau\|x - y\|$$

for some $j \in \{1, \ldots, k\}$, and we conclude that $\{rz_1, \ldots, rz_k\}$ is a basis for A at level $(1, \tau)$, which completes the proof.

Appendix B

Proof of Lemma 1: Construct a graph G such that the vertices are the points of A, and any two of them are connected by an undirected edge, i.e., G is complete. Color the edges of G with k colors $C_1, \ldots, C_k$, where k is the number of the basis points $z_1, \ldots, z_k$ according to Definition 2. The coloration is constructed as follows: An edge (x_μ, x_ν) is colored by C_j if

$$|\rho(x_\mu, z_j) - \rho(x_\nu, z_j)| \geq \beta\rho(x_\mu, x_\nu)$$

holds. By Definition 2, this must hold for some j, and in case of ambiguity, we chose the z_j with the smallest index.

Now, if G is infinite, then by Ramsey's theorem of graph theory [12], there exists an infinite complete monochromatic subgraph G' of G. It means that there is an infinite sequence $x_1, x_2, \ldots \in A$ of points such that for some basis point z_j, we have

$$|\rho(x_\mu, z_j) - \rho(x_\nu, z_j)| \geq \beta\rho(x_\mu, x_\nu) \geq \beta\rho_0 > 0$$

for $\mu = 1, 2, \ldots$, $\nu = 1, 2, \ldots$, $\mu \neq \nu$. Indexing the points so that

$$\rho(x_1, z_j) \leq \rho(x_2, z_j) \leq \rho(x_3, z_j) \leq \ldots$$

we have

$$|\rho(x_n, z_j) - \rho(x_1, z_j)| \geq (n - 1)\beta\rho_0 \to \infty$$

as $n \to \infty$. On the other hand, Definition 2 and the boundedness of A yields

$$|\rho(x_n, z_j) - \rho(x_1, z_j)| \leq \alpha\rho(x_n, x_1) \leq \alpha \sup_{x,y \in A} \rho(x, y) < \infty$$

which is a contradiction. Thus, G must be finite, which proves the finiteness of A.

Appendix C

Proof of Lemma 2: Set $r_2 = \frac{\beta}{2\alpha}r_1$, where α, β are the same as in Definition 2. If there is a point $x_1 \in A$ with $p(x_1, r_2) = 0$, then set $A_1 = A - B(x_1, r_2)$. If there is a point $x_2 \in A_1$ such that $p(x_2, r_2) = 0$, then again put $A_2 = A_1 - B(x_2, r_2)$. Repeat this procedure as long as possible, each time deleting a ball of radius r_2 and measure zero centered in the remaining subset of A. For two such centers x_i, x_j, $i \neq j$, we have $\rho(x_i, x_j) \geq r_2$ by the construction; thus, these centers form a bounded discrete set that must be finite by Lemma 1. Therefore, after a finite number of steps, we arrive at a set $A' \subset A$ such that $p(x, r_2) > 0$ holds for all $x \in A'$. On the other hand, $P_X(A') = 1$ still remains true. Now, set

$$a_0 = \inf_{x \in A'} p(x, r_1).$$

If $a_0 > 0$, then the assertion of the lemma holds with $\epsilon = a_0$; therefore, it remains to be seen that $a_0 = 0$ is impossible.

Assume indirectly that $a_0 = 0$. Then, there exist sequences $y_n \in A'$, $\epsilon_n > 0$, $n = 1, 2, \ldots$, such that

$$p(y_n, r_1) < \epsilon_n \quad \text{and} \quad \lim_{n \to \infty} \epsilon_n = 0. \tag{14}$$

Now, pick a point $u_1 \in A'$. If

$$|B(u_1, r_2) \cap \{y_n\}| < \infty$$

then put $A_1' = A' - B(u_1, r_2)$, and pick a point $u_2 \in A_1'$. Again, if

$$|B(u_2, r_2) \cap \{y_n\}| < \infty,$$

then put $A_2' = A_1' - B(u_2, r_2)$ and so on, as long as possible. As above, the construction guarantees that the centers u_i form a discrete bounded set; therefore, by Lemma 1, we must get stuck after a finite number of steps. Thus, there is a point $u \in A'$ such that the ball $B(u, r_2)$ contains an infinite subsequence $\{y_n'\}$ of $\{y_n\}$. We now show that

$$B(u, r_2) \subset B(y_n', r_1), \quad n = 1, 2, \ldots \tag{15}$$

holds. Pick a point $x \in B(u, r_2)$. It suffices to be seen that $\rho(x, y_n') \leq r_1$. Indeed, by Definition 2, for an appropriate basis point z_j, we have

$$\begin{aligned} \beta\rho(x, y_n') &\leq |\rho(x, z_j) - \rho(y_n', z_j)| \\ &\leq |\rho(x, z_j) - \rho(u, z_j)| + |\rho(u, z_j) - \rho(y_n', z_j)| \\ &\leq \alpha\rho(x, u) + \alpha\rho(y_n', u) \leq 2\alpha r_2 \end{aligned}$$

that is

$$\rho(x, y_n') \leq \frac{2\alpha}{\beta} r_2 = r_1$$

which proves (15). From this and (14), we obtain

$$p(u, r_2) < \epsilon_n'$$

with $\epsilon_n' \to 0$ as $n \to \infty$, which implies $p(u, r_2) = 0$. This contradicts the construction of A', and the lemma is proved.

REFERENCES

[1] D. Dobkin and R. J. Lipton, "Multidimensional searching problems," *SIAM J. Comput.*, vol. 5, no. 2, pp. 181–186, June 1976.

[2] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Software*, vol. 3, no. 3, pp. 209–226, Sept. 1977.

[3] J. L. Bentley, B. W. Weide, and A. C. Yao, "Optimal expected—Time algorithms for closest point problems," *ACM Trans. Math. Software*, vol. 6, no. 4, pp. 563–580, Dec. 1980.

[4] I. K. Sethi, "A fast algorithm for recognizing nearest neighbors," *IEEE Trans. Syst. Man Cyber.*,, vol. SMC-11, pp. 245–248, Mar. 1981.

[5] E. Vidal, "An algorithm for finding nearest neighbors in (approximately) constant average time," *Patt. Recogn. Lett.*, vol. 4, no. 3, pp. 145–157, July 1986.

[6] A. Faragó, T. Linder, G. Lugosi, and T. Pikler, "On the algorithmic problems of the nearest neighbor method," *Híradástechnika (Telecommunication)*, vol. XXXIX, no. 8, 1988; in Hungarian.

[7] K. Motoishi and T. Misumi, "Fast vector quantization algorithm by using an adaptive search technique," presented at *IEEE Int. Symp. Inform. Theory* (San Diego, CA), Jan. 14–19, 1990.

[8] T. Linder and G. Lugosi, "Classification with a low complexity nearest neighbor algorithm," presented at *IEEE Int. Symp. Inform. Theory* (San Diego, CA), Jan. 14–19, 1990.

[9] L. Devroye, *Lecture Notes on Bucket Algorithms*. Boston: Birkhäuser, 1986.

[10] E. Vidal, H. Rulot, F. Casacuberta, and J. Benedi, "On the use of metric-space search algorithm (AESA) for fast DTW-based recognition of isolated words," *IEEE Trans. Acoust. Speech. Signal Processing*, vol. ASSP-36, pp. 651–660, 1988.

[11] R. L. Wheeden and A. Z. Zygmund, *Measure and Integral*. New York: Marcel Dekker, 1977.

[12] C. Berge, *Graphs and Hypergraphs*. Amsterdam: North Holland, 1973.

[13] E. Vidal and M. J. Lloret, "Recent results on the application of a metric-space seach algorithm (AESA) to multispeaker data," in *Recent Advances in Speech Understanding and Dialog Systems* (H. Niemann, Ed.). New York: Springer Verlag, 1988.

Learning Bias in Neural Networks and an Approach to Controlling Its Effects in Monotonic Classification

Norman P. Archer and Shouhong Wang

Abstract—**As a learning machine, a neural network using the backpropagation training algorithm is subject to learning bias. This results in unpredictability of boundary generation behavior in pattern recognition applications, especially in the case of small training sample size. This research sugests that in a large class of pattern recognition problems, such as managerial and other problems possessing monotonicity properties, the effect of learning bias can be controlled by using multiarchitecture monotonic function neural networks.**

Index Terms—**Backpropagation, learning bias, monotonically separable, monotonic boundary, monotonicity, neural network.**

Manuscript received February 11, 1991; revised August 11, 1992. Recommended for acceptance by Associate Editor E. Delp.

N. P. Archer is with the School of Business, McMaster University, Hamilton, Canada L8S 4M4.

S. Wang is with the Faculty of Business, University of New Brunswick, St. John, Canada E2L 4L5.

IEEE Log Number 9208016.

I. INTRODUCTION

Despite a considerable amount of recent research directed towards pattern recognition applications of neural networks, the predictability of classification results from neural networks is still an open question [1]. Neural network researchers are painfully aware of this problem and have been trying to improve available algorithms to deal with statistical data [2]–[4]. However, the major limitation of these methods is that assumptions are necessary about certain distribution parameters, and the selection of these parameters influences the results for a particular problem.

Research has shown that information that is based only on limited training sample data is often not sufficient in classification. With standard backpropagation neural networks, it is impossible to control boundary functions unless some assumptions are made about their shapes, and it has been difficult to develop meaningful generalizations in this area. However, problem domain knowledge may be very useful in developing the realistic assumptions needed for such generalizations in pattern recognition.

There has been a substantial body of research using heuristics rather than statistical principles to improve the classification performance of neural networks. For example, Kawabata [5] used interpolation training to make such improvements, but using local information to regulate neural network behavior depends heavily on the training sample's density [5]. Casasent and Barnard [6] suggested on adaptive clustering training method, but specific knowledge about classification prototypes is required when applying this method. Unlike the foregoing methods, we introduce a neural network model utilizing monotonicity, which is a generic characteristic of many decision-making situations, to improve the performance of backpropagation neural networks [7] in solving classification problems.

II. LEARNING BIAS

The behavior of the neural network learning process is relatively unpredictable (cf. e.g., [9]). This means that the classification boundary is determined not only by the statistical constitution of the training data, but it is also influenced by other factors, including the following:

a) Architecture of the neural network model (e.g., number of hidden nodes)
b) parameters (e.g., learning rate) of the learning algorithm
c) initial state of the neural network
d) sequence of training data points
e) the stopping criteria of the learning procedure.

These factors bring some inherent knowledge or learning rules to bear on the machine learning process. These may or may not be pertinent to a particular task or a specific problem and are referred to as *learning bias* [10], [11].

There is a close relationship between learning bias and "biased" classification boundary results. Neural networks with their individual learning bias do not generate identical classification boundaries from the same training data sets. Thus, the classification boundary generated by a standard neural network is most likely to be biased because we have no knowledge about how to control the learning bias in order to generate an "unbiased" classification boundary.

In the BPLMS learning algorithm [7], the neural network weights are gradually modified according to the current training sample data, the current neural network state, and the currently adopted learning rate η. With this algorithm, the learning procedure stops when a final training sample point is correctly classified, that is, the error