Random Embedding Machines for Pattern Recognition

Yoram Baram
Department of Computer Science, Technion, Israel Institute of Technology, Haifa 32000, Israel

Real classification problems involve structured data that can be essentially grouped into a relatively small number of clusters. It is shown that, under a local clustering condition, a set of points of a given class, embedded in binary space by a set of randomly parameterized surfaces, is linearly separable from other classes, with arbitrarily high probability. We call such a data set a *local relative cluster*. The size of the embedding set is shown to be inversely proportional to the squared local clustering degree. A simple parameterization by embedding hyperplanes, implementing a voting system, results in a random reduction of the nearest-neighbor method and leads to the separation of multicluster data by a network with two internal layers. This represents a considerable reduction of the learning problem with respect to known techniques, resolving a long-standing question on the complexity of random embedding. Numerical tests show that the proposed method performs as well as state-of the-art methods and in a small fraction of the time.

1 Introduction

Finding a separable and generalizable representation for class-labeled data is an open problem, central to the areas of pattern recognition, classification, neural networks and learning theory. It has been motivated by Rosenblatt's (1958) perceptron, consisting of a single linear threshold unit and a learning algorithm, which is known to find a solution in finite time when the classes are linearly separable (Minsky & Papert, 1988). Rosenblatt further proposed using a layer of randomly parameterized linear threshold units and separating the resulting representations by the perceptron learning rule. A similar approach, replacing the perceptron learning rule by a stabilized version of it, was reported to perform well experimentally on complex problems by Gallant and Smith (1987), who, for a training set size M, proposed an empirical estimate of $M/4$ embedding cells. Yet while exact embedding by hyperplanes has been addressed in learning theory (Pitt & Warmuth, 1990), a theory justifying the use of random embedding appears to have been lacking.

The effectiveness of networks having a single internal layer of linear threshold units has been questioned in several works. Gori and Scarselli (1998) have argued that such networks are unable to produce closed sepa-

Neural Computation **13, 2533–2548** (2001) © 2001 Massachusetts Institute of Technology

ration surfaces reliably. Bartlett and Ben-David (1999) have recently shown that the difficulty of finding a separating set of more than a single linear threshold unit increases exponentially with the input dimension, even when a solution exists. Such complexity seems to contradict the underlying motivation for the introduction of artificial neural networks: fast and simple learning algorithms that may be biologically plausible. Indeed, more recent efforts in the development of pattern recognition tools have diverted from that original vision, aiming mainly at increased accuracy, seemingly giving up on speed and simplicity.

We propose that many of the theoretical and the practical shortcomings of the original concepts can be resolved by observing that most pattern recognition problems deal with highly structured data and that, as witnessed in many instances (see, e.g., Motwani & Raghavan, 1995, and the simulated annealing method, Kirkpatrick, Gelatt, & Vecchi, 1983), random parameterization reduces complexity and increases computation speed. We propose local grouping of the data about a subset of points, sequentially selected at random from the remaining set of ungrouped points. We call these points *clustering points* and the associated data groups *local relative clusters*. We introduce a measure of the relevant structural property, which we call the *local clustering degree*, and show that, under a condition on this measure, a local relative cluster can be linearly separated and generalized, with arbitrarily high probability, by a system of randomly parameterized separation surfaces, whose number is inversely proportional to the squared local clustering degree. We consider, in particular, two-class problems embedded in binary space by hyperplanes, but the analysis similarly applies to multiclass problems and to embedding by other surfaces. We show that, under local relative clustering, linear separation is implemented by a simple voting mechanism. This naturally suggests the separation of multiclustered data by a network having two internal layers of linear threshold units, which implements a reduced version of the nearest-neighbor classifier. Performance on real data is finally compared to those of the nearest-neighbor method and support vector machines.

2 Local Relative Clustering

We consider two disjoint sets U^+ and U^- in the nth-dimensional real space R^n and their union set U. Suppose that the parameters $v \in R^n$, $b \in R^1$ of a hyperplane $(v, u) = b$, where (v, u) denotes inner product, obey some probability law, which we call the *embedding rule*. Let the probability that a given pair of points $u^{(i)}, u^{(j)} \in U$ are on the opposite sides of the hyperplane be $p(u^{(i)}, u^{(j)})$. Given an embedding rule, a subset $U' \subset U^+$ is said to be locally relatively clustered of degree ϵ at a point $u^* \in R^n$ with respect to U^-, if there exists some $\epsilon > 0$, such that

$$p\left(u^*, u^{(j)}\right) - p\left(u^*, u^{(k)}\right) \geq \epsilon \text{ for any } u^{(j)} \in U^-,\ u^{(k)} \in U'. \tag{2.1}$$

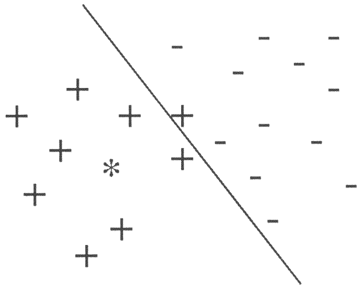


Figure 1: A clustering point and an embedding hyperplane.

A point u^* that satisfies equation 2.1 is said to be a clustering point of U'.

The relevance of these definitions to the notion of clustering is quite obvious. The condition states that each of the points of the locally clustered set is more likely to be on the same side of a random hyperplane as the clustering point than any of the points of the other class. A random hyperplane separates a local relative cluster of a given class from the other class in a probabilistic sense.

Example 1. Figure 1 shows two sets of points, U^+ and U^-, and a point $*$. Let the embedding rule select a point of U^- at random (that is, with probability $1/M^-$, where M^- is the cardinality of U^-), connect it to $*$ by a straight line segment, and pass a hyperplane perpendicular to this line segment at the midpoint. By this embedding rule, $*$ is a clustering point of U^+. In fact, by this embedding rule, every point of U^+ is a clustering point of U^+.

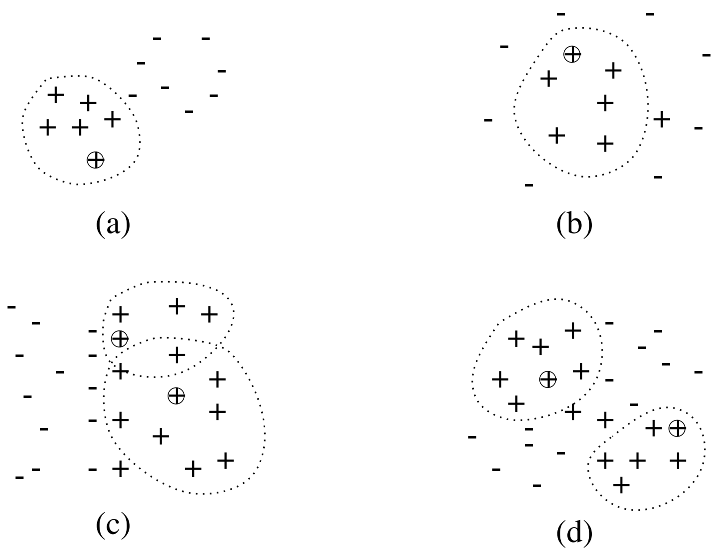
Now suppose that N random hyperplanes, parameterized by $v^{(i)}$, $b^{(i)}$, $i = 1, \ldots, N$, are independently generated according to the embedding rule. A member u of U will be transformed into (or embedded in) the binary space $\{\pm 1\}^N$ by a layer of N linear threshold units, the ith of which performs the function

$$x_i = \begin{cases} 1 & \text{if} \quad \left(v^{(i)}, u\right) \geq b^{(i)} \\ -1 & \text{if} \quad \left(v^{(i)}, u\right) < b^{(i)}, \end{cases} \tag{2.2}$$

where $(v^{(i)}, u)$ denotes the inner product between the two vectors. The resulting vector x is the internal representation of the input vector u.

An internal representation x will be transformed into a scalar output y by

$$y = \begin{cases} 1 & \text{if} \quad (w, x) \geq 0 \\ -1 & \text{if} \quad (w, x) < 0. \end{cases} \tag{2.3}$$



(a)

(b)

(c)

(d)

Figure 2: Examples of local relative clusters.

w is a vector of output weights. An input u will be said to be classified correctly if $y = 1$ when $u \in U^+$ and $y = -1$ when $u \in U^-$.

How can a clustering point and embedding hyperplanes be selected? There are many possibilities. The following seems to be a natural choice: Select a data point u^* from a given class as a clustering point. Select a set of points $u^{(i)},\ i = 1, \ldots, N$ of the other class at random (independently, with the probability that a given point is selected being the reciprocal of the number of points). Pass an embedding hyperplane between each of these points and the clustering point, intersecting the line segment connecting them perpendicularly at the midpoint. The equation of the ith hyperplane is $(v^{(i)}, u) = b^{(i)}$, where $v^{(i)} = u^{(i)} - u^*$ and $b^{(i)} = 0.5(u^{(i)} + u^*)(u^{(i)} - u^*)$.

What is a local relative cluster $U' \in U^+$ associated with a given clustering point u^*? It is a subset $U' \in U^+$ for which condition 2.1 holds for some ϵ. Clearly, every data point has an associated local relative cluster of some degree $\epsilon > 0$ of the same class. This set is not empty, since it contains at least the point itself (for which the clustering degree is 1).

The data set is assumed to be given, so there is nothing random about it. Randomness is introduced through the selection of the data points used in the selection of the clustering point and in the parameterization of the embedding set.

Example 2. Consider the four cases depicted in Figure 2. In Figure 2a, it is quite obvious that the entire set U^+ is locally relatively clustered (for some $\epsilon > 0$) with respect to U^- at each of the points of U^+. In Figures 2a through 2d, local relative clusters of degree $\epsilon > 0$ of the encircled points are surrounded by dotted lines.

Clearly, the definition of a local relative cluster is not restricted to embedding sets consisting of hyperplanes alone. Other surfaces, open and closed (e.g., spheres, rectangles), may be used. The analysis of the following sections holds in this general context.

3 Linear Separability of the Internal Representations Corresponding to a Local Relative Cluster

The internal representations $x^{(i)}$, $i = 1\ldots, M$, form two sets, X^+ and X^- in the space $\{\pm 1\}^N$, whose members correspond to U^+ and U^-, respectively. Two sets X_1 and X_2 in $\{\pm 1\}^N$ are said to be linearly separable if there exists some hyperplane in R^N having X_1 on one of its sides and X_2 on the other. Practical separation will require some margin between the separating hyperplane and each of the two sets, as provided by the following result.

Theorem 1. *If a set $U' \in U^+$ is locally relatively clustered of degree ϵ at some point $u^* \in R^n$ with respect to U^- and*

$$N > \frac{2\ln(1/P)}{\epsilon^2}, \tag{3.1}$$

then, with probability greater than $(1-P)^2$, there exists some $\delta > 0$ such that

$$\left(x^*, x^{(k)}\right) - 0.5N \le -\delta N \quad \textit{for all } x^{(k)} \in X^- \tag{3.2}$$

and

$$\left(x^*, x^{(j)}\right) - 0.5N \ge \delta N \quad \textit{for all } x^{(j)} \in X', \tag{3.3}$$

where x^* is the internal representation of u^* and X' and X^- are the sets of internal representations corresponding to U' and U^-, respectively.

Theorem 1 states that the sample complexity of the separation problem is inversely proportional to the squared local clustering degree. An immediate consequence of theorem 1 is Corollary 1:

Corrollary 1. *Suppose that the set $U' \in U^+$ is locally relatively clustered at each of its points with respect to the set U^-. Then, under condition 3.1, the sets X' and X^- are linearly separable by any of the internal representations of U', with probability greater than $(1-P)^2$.*

Corollary 1 suggests that under local relative clustering, one of the internal representations of the set U' be used as the output weights vector w. Clearly, by the embedding method proposed in the previous section (passing the embedding hyperplanes perpendicularly to the line segments connecting the local clustering point to points of U^- at the midpoints), the internal representation of the local clustering point is simply a vector of dimension N whose components are all -1. This is a considerable simplification over a possible application of known learning paradigms, such as Rosenblatt's rule, to the internal representations.

Good performance on the labeled data is of little use if it cannot be extended to new, unlabeled points. Such extension has been called *generalization*. According to Vapnik and Chervonenkis (1971), a classifier generalizes if, for any positive scalar ϵ,

$$\lim_{M\to\infty} \sup_{\theta} \left\{ \left| P(\theta) - \hat{P}(\theta) \right| > \epsilon \right\} = 0, \tag{3.4}$$

where θ is the parameter vector, $P(\theta)$ is the underlying probability of misclassification, and $\hat{P}$ is the empirical probability of misclassification calculated from a training set. We have the following result:

Theorem 2. *The classifier defined by equations 2.2 and 2.3, with N linear threshold units, N satisfying equation 3.1, generalizes a local cluster, provided that the local clustering degree remains bounded above zero as the data size increases.*

The situation assumed in theorem 2 can be visualized by increasing the number of points of U^+ in each of the local clusters surrounded by the dotted lines in Figure 2.

4 Random Embedding Machines for the Classification of Multicluster Data

A local classifier, consisting of a single internal layer of linear threshold units and a single external such unit, is defined by a clustering point, randomly selected from U^+, and by the set of embedding hyperplanes. The input weights of the internal units are the parameters of the embedding hyperplanes, and the input weights of the external unit, which we call a *local clustering unit* are all -1. Now suppose that there are training points of U^+ that are not assigned correctly by this classifier. Of these points, select one, make it a second clustering point, and construct its embedding set. If this new local classifier does not correctly classify at least a specified number of the misclassified points, it is eliminated ("perform or die"). More clustering points, and hence, clustering units, can be added with embedding sets, until only a certain fraction of the data points remains misclassified. The clustering points represent a second internal layer of linear threshold

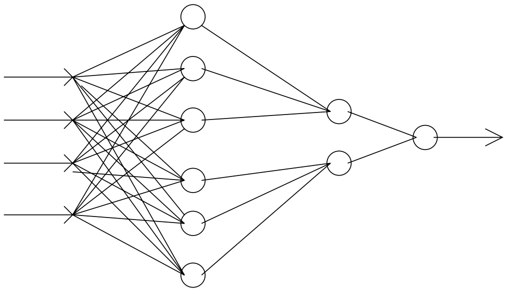


Figure 3: Random embedding machine with two internal layers. The first layer represents the embedding set, the second the clustering points.

units, connected to an external unit that performs the "or" operation on their outputs (which is achieved by a threshold value 1). Such a network is depicted in Figure 3.

Consider again the class arrangements depicted in Figure 2. Suppose that a clustering point is selected at random from the set U^+ and that an embedding set is defined by hyperplanes perpendicular to the lines connecting the clustering point to randomly selected data points of U^- at the midpoints. Case a will be solved by a network with a single internal layer of linear threshold units and a single external such unit. Case b is also likely to be resolved by such a network, but may require a second clustering point, depending on the choice of the first. Case c will require several clustering points, hence, a second internal layer. Case d is likely to be resolved by two clustering points, one for each of the local clusters of U^+, in the following manner: The first clustering point is picked from one of the the two local clusters of U^+, say, the upper left. It is represented by the upper unit of the second internal layer in Figure 3. Most of the points of the lower right local cluster of U^+ will not be assigned correctly by the first local clustering unit. A second clustering point will be selected at random from those points. It is represented by the lower unit of the second internal layer in Figure 3. Most of the points of that local cluster will be assigned correctly by this unit. The external unit, connected to the second layer by unity weights and having a unity threshold, will correctly assign most of the data set.

A network of the type depicted in Figure 3 may be constructed for each class (a multiclass problem can be treated similarly to two-class problems). The external unit for each class may output the maximal value of the outer product between the representation in the first internal layer and the weights vectors of the local clustering units. The output units for the different classes will then compete, the one with the largest output value winning. If a point is assigned to both classes or to none, it will remain unclassified. Indecision can be beneficial (Baram, 1998). Since each of the local classifiers employs random parameters and embeds the input vector in N-dimensional binary space, we call the network a *random embedding machine*.

4.1 A Value for N. A question of obvious practical importance in designing a random embedding machine is what value to choose for N, the embedding set size. Clearly, we would like to choose the minimal N allowed by the bound, equation 3.1. Consider first the numerator of the right-hand side of equation 3.1. For a wide range of P values, say, $10^{-6} < P < 10^{-2}$, the value of $\ln(1/P)$ varies within a relatively small range, $4.6 < \ln(1/P) < 13.8$. An intermediate value, $\ln(1/P) = 10$, corresponding to some $P < 10^{-4}$, yields a linear separability probability of value $(1 - P)^2 > 0.9998$. Turning to the denominator of equation 3.1, we note that, for a given clustering point, each value of ϵ defines a local relative cluster, as discussed in section 2. Clearly, $0 \le \epsilon \le 1$. A large value of ϵ will yield, through equation 3.1, a small value for N. However, the corresponding local relative cluster may be too small, and an unnecessarily large number of such clusters may be required in order to contain most of the training set, leading to an unnecessarily large classifier. On the other hand, a small value for ϵ may yield, through equation 3.1, an unnecessarily large N. Yet the corresponding local relative clusters may still be small, and the resulting classifier may again be unnecessarily large. A sensible choice for ϵ is, then, an intermediate value, say, $\epsilon = 0.5$, which, together with the requirement $P < 10^{-4}$, yields

$$N = 20. \tag{4.1}$$

On a scale of 10, 20, 30, and so on, the value $N = 20$ produced the best results for both synthetic and real-life examples, presented in section 5.

4.2 On the Computational Complexity. It should be quite obvious that the computational complexity of the algorithm described in the previous section depends not only on the size of the embedding set but also on the number of clustering points that will be generated, which depends, in turn, on the structure of the data. In order to formalize a correspondence between the structure of the data and the number of clustering points, we introduce the following definition: A data set is *uniformly locally clustered* of degree ϵ and order K if there exist some ϵ and some K such that the data set is contained in the union of K local relative clusters of degree ϵ.

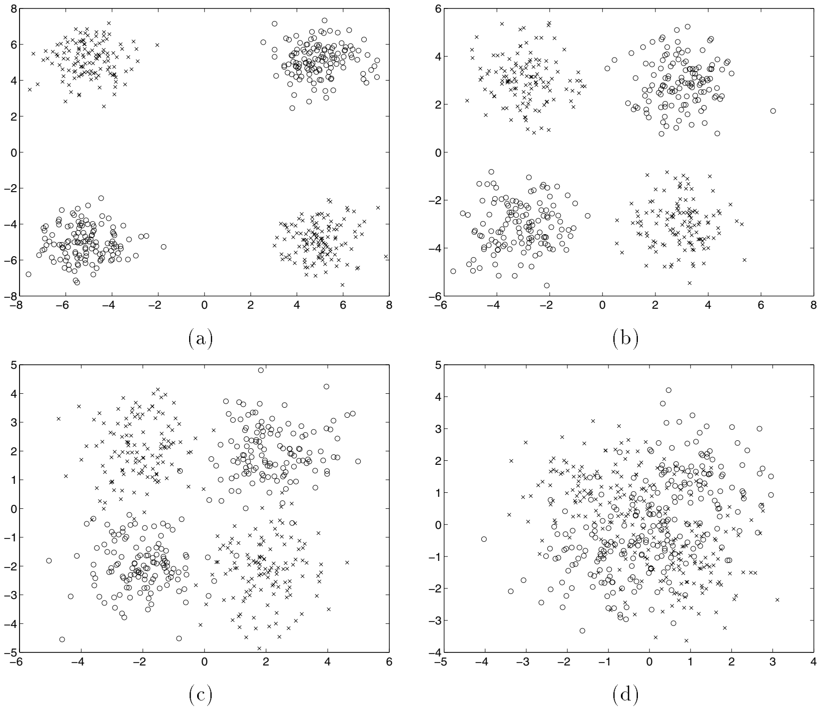
In each of the cases depicted in Figure 2, define the groups of points of U^+, which are not contained in the areas surrounded by the dotted lines, as additional local relative clusters of degree ϵ, where ϵ is no greater than the clustering degrees of the surrounded sets. Then Figures 2a and 2b show uniformly locally clustered data sets U^+ of degree ϵ and orders 1 and 2, respectively, and such data sets of order 3 are shown, respectively, in Figures 2c and 2d.

An immediate consequence of theorem 1 is that if the data are uniformly locally clustered of degree ϵ and order K, the complexity of the training algorithm, which guarantees correct classification with probability greater than $(1-P)^2$, is $O\left(\frac{2\ln(1/P)}{\epsilon^2}M\right)$ and that of the testing algorithm (the operational complexity) is $O\left(\frac{2\ln(1/P)}{\epsilon^2}K\right)$.

4.3 Voting by Weak Local Experts and Reduced Nearest Neighbor. Given a new input u, each embedding unit finds if u is on the same side of the corresponding hyperplane as the associated clustering point u^*. For embedding hyperplanes that cut the line segments between randomly selected points of U^- and u^* perpendicularly at the midpoint, the internal representation of u^* is clearly a vector of -1 components. This means that the external unit corresponding to a relative local cluster performs a majority vote among the embedding units. Each of the embedding units is equipped with the quality that it classifies correctly at least one pair of points of the opposite classes, that is, the pair that defines the corresponding embedding hyperplane. This qualifies the unit as a "weak local expert," with a presumed probability of success slightly higher than 0.5 with respect to a local relative cluster. Theorem 1 tells us that the combined operation of a sufficiently large number of these units by voting brings the probability of success to an arbitrarily high value.

The minimum-distance local separator of a point u of U^+ from U^- is the intersection of the half-spaces defined by the hyperplanes cutting the line segments connecting u to each of the points of U^- perpendicularly at the midpoint. The nearest-neighbor criterion divides R^n into minimum-distance local separators of the data points. The hyperplanes associated with the separator of a point u embed it in a space of dimension equal to the size of U^-. The reduced nearest-neighbor classifier (Baram, 2000) employs a subset of the local separators of the points of U^+, which covers U^+ (the latter property makes the classifier consistent). Clearly, only a few of the embedding hyperplanes end up contributing to the shape of a minimum-distance local separator. Now suppose that instead of starting with all the embedding hyperplanes, only a randomly selected subset of them is used in the construction of a minimum-distance local separator. Also, as in the reduced nearest-neighbor classifier, instead of employing the local separators of all the points of U^+, only a few are used, provided that their union covers U^+ (hence, the classifier is still consistent). This is the proposed random embedding machine. It can be viewed as a random approximation of the reduced nearest-neighbor classifier.

Finally, we note that a random embedding machine is fundamentally different from the networks proposed by Gallant (1986), Gallant and Smith (1987), Mézard and Nadal (1989), Marchand, Golea, and Rujan (1990), and Frean (1990), all of which involve variations of the perceptron learning rule and consequently are rather slow. The random embedding machine achieves its remarkably fast performance by a different "learning" concept,



(a) (b)

(c) (d)

Figure 4: Data for example 1.

applying very simple operations to the data points.

5 Examples

5.1 Example 1. Two-class data in an "XOR" arrangement were synthetically generated by independently sampling four gaussian densities of variance 1, as shown in Figure 4.

In the four cases, the centers of the densities were at distances 10, 6, 4, and 2, respectively, from each other. The training set and the test set consisted of 500 points each. Ten independent samples of the data sets were generated, and 10 independent selections of the embedding set associated with each clustering point were generated for each sample. The results were averaged over these 100 cases, consisting of 1000 points each.

A random embedding machine consisting of two internal layers with four local clustering units (one per local cluster) evolved through the mechanism described in the previous section (starting with one such unit per class, adding units if points of that class are misclassified). Embedding sets of sizes 10, 20, 30, . . . , 100 were examined. The number of classification er-

Table 1: Classification Results for Example 1.

	Case a	Case b	Case c	Case d
REM results ($N = 20$)				
Success	100%	99.35%	95.15%	67.95%
Flops	123,580	107,093	109,671	93,401
Nearest-neighbor results				
Success	100%	99.32%	93.2%	65.88%
Flops	2,250,668			
SVM results (gaussian RBF kernels)				
Success	99.38%	98.80%	92.74%	66.67%
Flops	1,224,860	1,198,435	1,190,847	1,833,650

rors was minimal for $N = 20$ in the harder cases, (Figures 4c and 4d), while it was invariant to the embedding set size in the easier cases, (Figures 4a and 4b). The classification results for the random embedding machine (REM), the nearest-neighbor classifier, and a support vector machine (SVM, Vapnik, 1995) employing gaussian radial basis functions with optimized parameters (Joachim, 1998) are given in Table 1. The REM performed with high precision when the classes were highly clustered, and as could be expected, performance degraded as the classes became highly mixed. The results achieved by the REM were better than those obtained by the nearest-neighbor classifier and by the SVM in the harder cases, and similar in the easier cases. The number of operations required by the REM was considerably smaller than those required by the other two methods in all cases. Testing was practically instantaneous for both the REM and the SVM classifiers.

5.2 Example 2. The Pima Indians Diabetes Database (UCI, 1999) has 768 instances of eight real–valued measurements, corresponding to 268 ill patients and 500 healthy ones. Seven exclusive test sets were selected for cross-validation, each consisting of 100 instances, the rest of the data serving in each case as the training set. As in the case of the previous example, best results were obtained for $N = 20$. A REM, having four local clustering units in the second internal layer, evolved through the construction mechanism described in the previous section. The classification results are given in Table 2, in comparison to those of the nearest-neighbor method and the optimized SVM employing radial basis functions. The REM required a con-

Table 2: Classification Results for Example 2.

	REM ($N = 20$)	Nearest-Neighbor	SVM
Success	65.51%	66.76%	65.57%
Flops	276,465	2,204,540	1,723,967

siderably smaller number of operations, yet its accuracy is comparable to that produced by the other two algorithms.

6 Conclusion

We have shown that separation and generalization can be achieved with arbitrarily high probability by embedding the data in binary space employing a system of elementary separation surfaces, such as hyperplanes, whose number depends on a local clustering property. This leads to the classification of multicluster data by a random embedding machine, whose performance is remarkably faster than that of known methods with little cost in accuracy.

Appendix

A.1 Proof of Theorem 1. Let $x^{(j)}$ be the point closest to x^* in X^- and let $x^{(l)}$ be the point farthest from x^* in X'. Let us define N variables ξ_k, $k = 1, \ldots, N$ such that

$$\xi_k = \begin{cases} 0 & \text{when} \quad x_k^* = x_k^{(j)} \\ 1 & \text{when} \quad x_k^* \neq x_k^{(j)}, \end{cases} \tag{A.1}$$

and N variables ζ_k, $k = 1, \ldots, N$ such that

$$\zeta_k = \begin{cases} 0 & \text{when} \quad x_k^* = x_k^{(l)} \\ 1 & \text{when} \quad x_k^* \neq x_k^{(l)}. \end{cases} \tag{A.2}$$

Then the Hamming distance between x^* and $x^{(j)}$ is

$$d_h\left(x^*, x^{(j)}\right) = \sum_{k=1}^{N} \xi_k, \tag{A.3}$$

and the Hamming distance between x^* and $x^{(l)}$ is

$$d_h\left(x^*, x^{(l)}\right) = \sum_{k=1}^{N} \zeta_k. \tag{A.4}$$

Note that ξ_k, $k = 1, \ldots, N$, are statistically independent, due to the statistical independence between the embedding hyperplanes (statistical independence between the hyperplanes follows from the statistically independent choices of the points of U^- used, together with u^*, in defining the hyperplanes, as described in section 2). Similarly, ζ_k, $k = 1, \ldots, N$, are statistically independent. Let

$$p_k\left(x^*, x^{(j)}\right) = P\{\xi_k = 1\} \tag{A.5}$$

and

$$p\left(x^*, x^{(j)}\right) = \frac{1}{N}\sum_{k=1}^{N} p_k\left(x^*, x^{(j)}\right). \tag{A.6}$$

Also let

$$p_k\left(x^*, x^{(l)}\right) = P\{\zeta_k = 1\} \tag{A.7}$$

and

$$p\left(x^*, x^{(l)}\right) = \frac{1}{N}\sum_{k=1}^{N} p_k\left(x^*, x^{(l)}\right). \tag{A.8}$$

Clearly, the relative clustering condition implies that there exists some $\delta > 0$ (in particular, $\delta = (p(x^*, x^{(j)}) + p(x^*, x^{(l)}))/2)$, such that $p(x^*, x^{(k)}) < \delta - \epsilon/2$ for all $x^{(k)} \in X'$ and $p(x^*, x^{(k)}) > \delta + \epsilon/2$ for all $x^{(k)} \in X^-$. The probability that the entire set X^- is at a Hamming distance no smaller than δN from x^* satisfies

$$P\left\{d_h(x^*, X^-) \geq \delta N\right\} = 1 - P\left\{d_h\left(x^*, x^{(j)}\right) < \delta N\right\}. \tag{A.9}$$

Employing Chernoff's bound (Kearns & Vazirani, 1994),

$$P\{S < (p - \gamma)N\} \leq e^{-2\gamma^2 N}, \tag{A.10}$$

with $S = \sum_{k=1}^{N} \chi_k$, where χ_k are Bernoulli trial (coin tossing) variables, $p = E(\chi_k) = P_r(\chi_k = 1)$ and $0 < \gamma \leq 1$, we obtain

$$P\left\{d_h\left(x^*, x^{(j)}\right) < \delta N\right\} < e^{-2(p(x^*, x^{(j)}) - \delta)^2 N}, \tag{A.11}$$

yielding, since $p(x^*, x^{(j)}) - \delta > \epsilon/2$,

$$P\left\{d_h\left(x^*, x^{(j)}\right) < \delta N\right\} < e^{-\frac{1}{2}\epsilon^2 N}. \tag{A.12}$$

The requirement,

$$P\left\{d_h\left(x^*, x^{(j)}\right) < \delta N\right\} < P, \tag{A.13}$$

is satisfied by

$$N > \frac{2 \ln 1/P}{\epsilon^2}, \tag{A.14}$$

which will give

$$P\left\{d_h(x^*, X^-) \geq \delta N\right\} > 1 - P. \tag{A.15}$$

The probability that the entire set X' is at a distance no greater than δN from x^* satisfies

$$P\left\{d_h\left(x^*, X'\right) \leq \delta N\right\} = 1 - P\left\{d_h\left(x^*, x^{(l)}\right) > \delta N\right\}. \tag{A.16}$$

Employing the other side of Chernoff's bound (Kearns & Vazirani 1994),

$$P\{S > (p + \gamma)N\} \leq e^{-2\gamma^2 N}, \tag{A.17}$$

and following arguments similar to those used above, we obtain

$$P\left\{d_h\left(x^*, X'\right) \leq \delta N\right\} > 1 - P \tag{A.18}$$

if

$$N > \frac{2\ln(1/P)}{\epsilon^2}. \tag{A.19}$$

From equation A.15, we have, with probability greater than $1 - P$,

$$d_h\left(x^*, x^{(k)}\right) = \frac{1}{4}\left\|x^* - x^{(j)}\right\|^2 \geq \delta N, \quad \text{for all } x^{(k)} \in X^-, \tag{A.20}$$

yielding, with probability greater than $1 - P$,

$$\left(x^*, x^{(k)}\right) - 0.5N \leq -\delta N \quad \text{for all } x^{(k)} \in X^-. \tag{A.21}$$

Similarly, it follows from equation A.18 that with probability greater than $1 - P$,

$$d_h\left(x^*, x^{(k)}\right) = \frac{1}{4}\left\|x^* - x^{(k)}\right\|^2 \leq \delta N, \quad \text{for all } x^{(k)} \in X', \tag{A.22}$$

yielding, with probability greater than $1 - P$,

$$\left(x^*, x^{(k)}\right) - 0.5N \geq \delta N \quad \text{for all } x^{(k)} \in X'. \tag{A.23}$$

Since the events A.21 and A.23 are independent, the probability that both occur is $(1 - P)^2$.

A.2 Proof of Theorem 2. Vapnik and Chervonenkis (1971) showed that for any distribution and a VC-dimension d,

$$|P(\theta) - \hat{P}(\theta)| < \sqrt{\frac{d\log(M/d) + \log(1/\delta)}{M}}, \tag{A.24}$$

with probability greater than δ. Baum and Hausler (1989) showed that the VC-dimension of a network of K linear threshold units and W weights is bounded as

$$d \leq 2W\log_2(eK), \tag{A.25}$$

where e is the base of the natural logarithm. Since, by theorem 1, the classifier has N units and $O(N^2)$ weights, where N is independent of M, the bound in equation A.25 is independent of M and the bound in equation A.24 vanishes as $M \to \infty$.

Acknowledgments

I thank Ran El-Yaniv, Mark Zlochin, and the referees for their very helpful comments.

References

Baram, Y. (1998). Partial classification: The benefit of deferred decision. *IEEE Trans. on Pattern Analysis and Machine Intelligence, 20*(8), 769–776.

Baram, Y. (2000). A geometric approach to consistent classification. *Pattern Recognition, 33*, 177–184.

Bartlett, P., & Ben-David, S. (1999). Hardness results for neural network approximation problems. Paper presented at EuroCOLT 99.

Baum, E. B., & Hausler, D. (1989). What size net gives valid generalization? *Neural Computation, 1*, 151–160.

Frean, M. (1990). The upstart algorithm: A method for constructing and training feedforward neural networks. *Neural Computation, 2*(2), 198–209.

Gallant, S. I. (1986). Optimal linear discriminants. In *Proc. Eighth International Conference on Pattern Recognition* (pp. 28–31). Paris.

Gallant, S. I., & Smith, D. (1987). Random cells: An idea whose time has come and gone ... and come again? In *Proc. of the IEEE First Int. Conf. on Neural Networks* (pp. 671–678). San Diego, CA.

Gori, M., & Scarselli, F. (1998). Are multilayer perceptrons adequate for pattern recognition and verification. *IEEE Trans. on Pattern Analysis and Machine Intelligence, 20*, 1121–1132.

Joachim, T. (1998). Available online at www-ai.informatik.uni-dortmund.de /FORSCHUNG /VERFAHREN/SVM_LIGHT/svm_light.eng.html.

Kearns, M. J., & Vazirani, U. V. (1994). *An introduction to computational learning theory*. Cambridge, MA: MIT Press.

Kirkpatrick, S., Gelatt Jr., C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*, pp. 671–680.

Marchand, M., Golea, M., & Rujan, P. (1990). A convergence theorem for sequential learning in two-layer perceptions. *Europhysics, 11*(6), 487–492.

Mézard, M., & Nadal, J. P. (1989). Learning in feedforward layered networks: The tiling algorithm. *J. of Physics, A 22*, 2191–2204.

Minsky, M. L., & Papert, S. A. (1988). *Perceptrons*. Cambridge, MA: MIT Press.

Motwani, R., & Raghavan, P. (1995). *Randomized algorithms*. Cambridge: Cambridge University Press.

Pitt, L., & Warmuth, M. (1990). Prediction preserving reducibility. *J. of Computer and System Sciences, 41*, 430–467.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review, 65*, 386–408.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer-Verlag.

Vapnik, V. N., & Chervonenkis, A. Y. (1971). On the uniform convergence of the relative frequencies of events to their probabilities. *Theory of Probability and Its Applications, 16*, 264–280.

UCI. (1999). Machine learning databases. Available online at: www.ics.uci.edu/ AI/ ML/Machine-Learning.html.

Received December 15, 1999; accepted January 15, 2001.