ONE-DIMENSIONAL KOHONEN LVQ NETS FOR MULTIDIMENSIONAL PATTERN RECOGNITION

EWA SKUBALSKA-RAFAJŁOWICZ*

A new neural network based pattern recognition algorithm is proposed. The method consists in preprocessing the multidimensional data, using a space-filling curve based transformation into the unit interval, and employing Kohonen's vector quantization algorithms (of SOM and LVQ types) in one dimension. The space-filling based transformation preserves the theoretical Bayes risk. Experiments show that such an approach can produce good or even better error rates than the classical LVQ performed in a multidimensional space.

Keywords: space-filling curve, pattern recognition, learning vector quantization, reduction of dimension

1. Introduction

We consider a classification problem with M classes and a training sequence L_n consisting of n d-dimensional feature vectors and their known class labels. It is known that any nonsingular transformation of the feature space including both linear and nonlinear mappings retains critical properties of the data such as the overlap or class separability among distributions (Fukunaga, 1972). These transformations do not reduce the dimensionality of the feature space. Fortunately, it is possible to obtain a class of mappings from a multidimensional bounded space into the unit interval, which preserves crucial (from the classification point of view) properties of the space. This class of mappings is based on space-filling curves.

A space-filling curve (SFC) is a continuous function which passes through every point in the unit hypercube. Examples were given by Peano (1890), Hilbert (1891) and Sierpiński (1912), among others (Sagan, 1994).

An SFC performs a transformation between an interval and a d-dimensional bounded space, resulting in a substantial compression of the information and retaining some of the spatially associative properties of the space. The key idea is to apply a quasi-inverse of the SFC in order to transform multidimensional learning data to one dimension.

The transformation has the property that the points which are close to one another after transformation must be so in the original d-dimensional space. Essential

* Technical University of Wrocław, Institute of Engineering Cybernetics, ul. Wybrzeże S. Wyspiańskiego 27, 50–370 Wrocław, Poland, e-mail: raf@immt.pwr.wroc.pl

properties of the classical SFCs can be found in (Milne, 1980; Sagan, 1994; Skubalska-Rafajłowicz, 1997a; 1997b; 1999; Skubalska-Rafajłowicz and Krzyżak, 1996). The space-filling curve to be discussed here is a Lipschitz continuous (of the order $1/d$) and Lebesgue measure-preserving transformation from the unit interval onto the multidimensional cube, i.e. it is a continuous curve which passes through every point in the unit hypercube.

Applications of space-filling curves include image scanning and coding (Quweider and Salaeri, 1995; Stevens *et al.*, 1983), computer display of real-valued functions with multidimensional bounded domains (Patrick *et al.*, 1968), image processing (Lamarque and Robert, 1996), combinatorial optimization (Bartholdi and Platzman, 1988; Platzman and Bartholdi, 1989) and pattern recognition (Patrick *et al.*, 1968).

The idea of using space-filling curves for statistical pattern recognition dates back to the paper of Patrick *et al.* (1968). Unfortunately, this idea was not further developed, because the space-filling curve based transformation cannot be one-to-one (Patrick, 1972, p.355).

Fortunately, a detailed examination of other possible properties of the space-filling curves confirms the possibility of using such a type of transformations in pattern recognition without spoiling the Bayes error (Skubalska-Rafajłowicz, 1997a; 1997b; 1999).

The LVQ algorithm with data in one dimension allows us to obtain a simple neural network structure with a small number of weights and nice asymptotic properties in the Bayesian framework.

2. Classification of Multidimensional Patterns and the Bayes Risk

We confine ourselves to the statistical pattern recognition problem (Fukunaga, 1972; Kulkarni *et al.*, 1998). Let $L_n = \{(X_k, Y_k),\ k = 1, 2, \ldots, n\}$ be the learning sequence which consists of independent, identically distributed (i.i.d.) pairs (X_k, Y_k) of random vectors, where $X_k \in \mathbb{R}^d$ is the k-th pattern (feature vector), while $Y_k \in \{1, 2, \ldots, M\}$ indicates its correct classification to one of M classes. Further, (X, Y) denotes a generic pair which is independent of L_n and has the same probability distribution as (X_k, Y_k). The problem is to classify X, using only L_n, i.e., to decide against Y when the distribution of (X, Y) is unknown.

Let us denote by $p_i = P\{Y = i\}$ the *a-priori* probability of class i and by $p_i(x) = P\{Y = i | X = x\}$, $x \in \mathbb{R}^d$, $i = 1, 2, \ldots, M$ the *a-posteriori* class probabilities. Let $f_i(x)$ be the density of class i.

The Bayes classification rule, denoted by $g^*: \mathbb{R}^d \to \{1, 2, \ldots, M\}$, minimizes the probability of error by deciding

$$g^*(X) = i, \quad \text{if} \quad p_i(X) = \max_{1 \le j \le M} p_j(X).$$

The Bayes risk J^* is defined by $P\{g^*(X) \neq Y\} = \inf_g P\{g(X) \neq Y\}$, where the infimum is taken over all $g\colon \mathbb{R}^d \to \{1, 2, \ldots, M\}$.

We confine our attention to patterns with bounded components and, without further loss of generality, we assume that X, $X_k \in I_d$, where I_d denotes the d-dimensional unit cube, i.e., $I_d = [0,1]^d$.

Further we consider only the case $M = 2$, since a general case $(M > 2)$ reduces easily to a sequence of dichotomies if the patterns are one-dimensional.

3. Some Properties of the Space-Filling Curves Based Data Transformation

Let $\Phi : I_1 \to I_d$ be a space-filling curve, i.e., a continuous mapping from the $[0,1]$ interval onto the d-dimensional unit cube $[0,1] \times [0,1] \times \cdots \times [0,1]$ and let Ψ be the pseudoinverse of Φ (it is known that Φ cannot be one-to-one).

It is not possible to define $\Phi(t)$ in a direct form. The mapping is regarded as the limit of a sequence of continuous functions (Sagan, 1994) which fill the space more and more densely. The construction of the space-filling curve can be based on the recursive partitioning of the multidimensional space into equal parts (elementary regions). The one-to-one correspondence is defined between elementary regions and subintervals of $[0,1]$ in such a way that neighboring subintervals have the corresponding elementary regions that are neighboring. The broken-line curve with nodal points in elementary regions forms some approximation of the space-filling curve a continuous mapping from the unit interval into the multidimensional cube). When the resolution of the decomposition approaches zero, the sequence of the broken-line curves converges uniformly to a continuous function which maps the unit interval onto the multidimensional unit cube (Milne, 1980; Sagan, 1994). For other constructions, see e.g. (Milne, 1980; Skubalska-Rafajłowicz, 1994). Figures 1–3 show the first three approximations of the Hilbert, Sierpiński and Peano space-filling curves in 2-D, respectively.

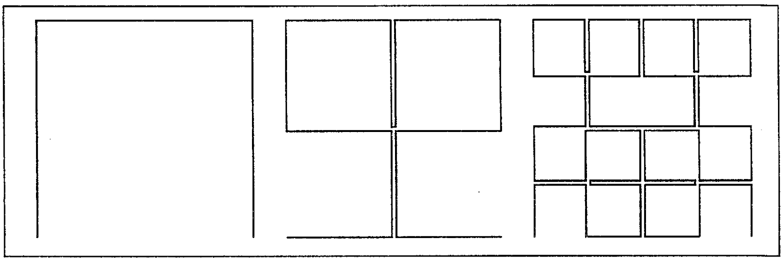

Fig. 1. Approximations of the Hilbert space-filling curve in 2-D.

The idea of the square-filling curves can be traced back to Peano (Milne, 1980; Sagan, 1994). For a long time, researchers have confined their attention to planar curves only, although the existence of multidimensional SFCs has been established (e.g., Steinhaus in 1936 (Sagan, 1994)). Milne (1980) showed how the Peano curve can be generalized to the multidimensional case. He proved that his curve preserves the Lebesgue measure and that the corresponding map is Lipschitz continuous with

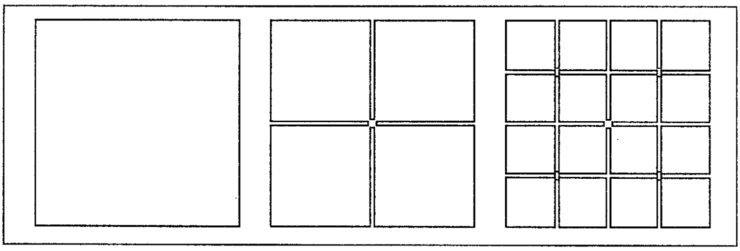

Fig. 2. Approximations of the Sierpiński (Knopp-Sierpiński) space-filling curve in 2-D.

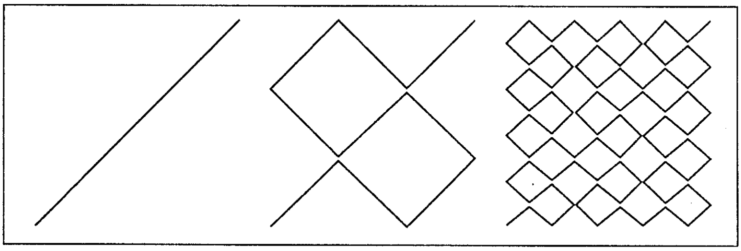

Fig. 3. Approximations of the Peano space-filling curve in 2-D.

the exponent $1/d$. In fact, all classical SFCs are Lebesgue measure-preserving (Sagan, 1994), i.e., subcurves of equal length fill multidimensional regions of equal volume.

Any space-filling curve satisfying the conditions C1–C4 given below is suitable for our purposes:

C1) Φ is a Lipschitz continuous mapping, i.e., $||\Phi(t_1) - \Phi(t_2)|| \leq L_d|t_1 - t_2|^{1/d}$, $t_1, t_2 \in I_1$, where $||\cdot||$ denotes the norm in I_d, $L_d > 0$ is some constant which depends on d.

We remark that for $d \geq 2$ the Lipschitz exponent $1/d < 1$, which implies that Φ is continuous, but is not differentiable (Sagan, 1994). We write $\Phi^{-1}(B) = \{t \in I_1 : \Phi(t) \in B\}$ for $B \subseteq I_d$. Denote by μ_d the Lebesgue measure in I_d, which is treated here as a dimensionless quantity.

C2) Φ preserves the Lebesgue measure, i.e., for every Borel set $B \subseteq I_d$ we have

$$\mu_d(B) = \mu_1\left(\Phi^{-1}(B)\right). \tag{1}$$

Note that construction of Φ depends on d, but this is not reflected in the notation. This convention is retained below in the definition of the pseudoinverse of Φ.

It is known that Φ cannot be one-to-one (Steele, 1989), thus it is not invertible, but it can be shown that it is a.e. one-to-one (Platzman and Bartholdi, 1989; Skubalska-Rafajłowicz, 1994). (We write a.e. if some property holds almost everywhere with respect to the Lebesgue measure.)

C3) Φ is a.e. one-to-one and there exists $\Psi : I_d \to I_1$, which is an a.e. inverse of Φ.

C4) Ψ preserves the Lebesgue measure in the sense that for every Borel set $A \subset I_1$ we have

$$\mu_1 \left(A \cap \Psi(I_d)\right) = \mu_d \left(\Psi^{-1}\left(A \cap \Psi(I_d)\right)\right). \tag{2}$$

In (Milne, 1980; Sagan, 1994), or (condition C4) in (Platzman and Bartholdi, 1989; Skubalska-Rafajłowicz, 1994) the results are formulated for the Peano, Hilbert and Sierpiński curves, which imply that conditions C1–C4 hold for them.

The next properties follow immediately from C1–C2.

C5) If $f(x)$ is a continuous function defined in I_d, then $g(t) \overset{\text{def}}{=} f(\Phi(t))$, $t \in I_1$ is also continuous. Furthermore, if the Hölder inequality holds for $f(x)$ with an exponent ν, $0 < \nu < 1$, then $g(t)$ remains a Lipschitz continuous function with the exponent ν/d.

C6) For every measurable function $f\colon\ I_d \to I_1$ the following integrals, interpreted in the Lebesgue sense, are equal:

$$\int_{B \subseteq I_d} f(x)\, \mathrm{d}x = \int_{\Psi(B) \subseteq I_1} f(\Phi(t))\, \mathrm{d}t. \tag{3}$$

The idea of the space-filling curve based transformation of multidimensional data is sketched in Fig. 4 in the case of two-dimensional observations.

Before describing the learning procedure, we point out an important property of the space-filling curve transformation. Our aim in this section is to underline that applying space-filling based transformations we keep the same level of the Bayes risk.

Lemma 1. *Let*

$$g^{\star}(x) = \begin{cases} 0 & \text{if } p_1 f_1(x) - p_0 f_0(x) \le 0, \\ 1 & \text{otherwise} \end{cases}$$

be a Bayes rule with risk $J^{\star}$. Then the decision rule $G^{\star}(t) \overset{def}{=} g^{\star}(\Phi(t))$, $t \in I_1$ is the Bayes rule for the classification problem with the same a-priori *probabilities p_0, p_1 and the class densities $f_0(\Phi(t))$ and $f_1(\Phi(t))$, $t \in I_1$. Furthermore, the Bayes risk of the transformed problem also equals $J^{\star}$.*

The proof follows immediately from C2, C4 and C6 (Skubalska-Rafajłowicz, 1999).

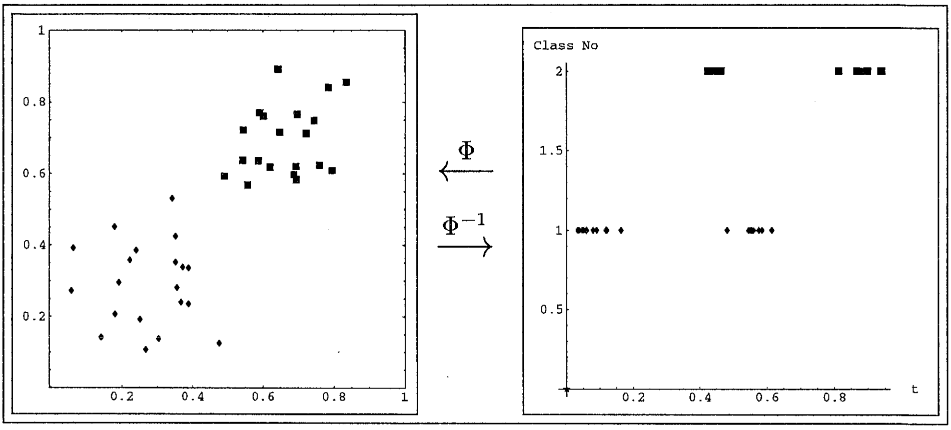

Fig. 4. Sketch of the idea of transforming patterns by a space-filling curve. Left panel: the original learning sequence with patterns from two classes (marked by diamonds and squares). The quasi-inverse of the Peano curve is used to transform the patterns to a one-dimensional form, which is shown in the right panel, where class labels 1 and 2 are also added to increase the readability. The arrows between the panels remind us that the transformations are realized by the Φ and Φ^{-1} mappings.

4. One-Dimensional Learning Vector Quantization

The Learning Vector Quantization (LVQ) proposed by Kohonen (Kangas *et al.*, 1990; Kohonen, 1990; 1995) is a supervised learning algorithm for training vector templates (prototypes, reference vectors) known as a codebook. A finite number of codebook vectors, each labeled with a class identifier, are chosen in the input space. During classification the distances between an input vector and all codebook vectors are computed. The input vector is assigned to the class corresponding to the nearest codebook vector (i.e. using 1-NN rule (Cover and Hart, 1967)). The learning process modifies only the location of the codebook vectors. Each training vector is compared with all prototypes and the nearest one is selected. If the class label of the winning codebook vector has the same class membership as the training one, then the selected prototype is moved in the direction of the training sample, otherwise it is moved in the opposite direction. Thus, the algorithm known as LVQ1 (Kohonen, 1990; 1995), performs both quantization and classification tasks. Unfortunately, the boundaries between the classes do not approximate the optimal Bayes boundary. Also the convergence of LVQ2 (Kohonen, 1995) is still an open problem (Diamantini and Spalvieri, 1998).

In the approach proposed here each d-dimensional X input vector may be thought of as a point $\Psi(X)$ on a unit interval I_1. Thus the prototypes simply form a set of numbers from I_1 labeled by its class-membership:

$$V = ((v_1,c_1),(v_2,c_2),\ldots,(v_N,c_N)), \quad v_i \in I_1, \quad c_i \in \{0,1\}, \quad i=1,2,\ldots,N.$$

The learning algorithm LVQ1 with an optimized learning rate, OLVQ1 (Kohonen, 1995), can be formulated in the following form:

ALGORITHM OLSQ1

Step 1. Transform the observations (X_j, Y_j), to (t_j, Y_j), $t_j = \Psi(X_j)$, $j = 1, 2, \dots, n$. Initialize all of the reference numbers $V = ((v_1(0), c_1), (v_2(0), c_2), \dots, (v_N(0), c_N))$ and individual learning rates α_i, $i = 1, \dots, N$.

Step 2. Generate randomly $t = t_i = \Psi(X_i)$ $c = Y_i$. Find v_k such that $|v_k(l) - t| \leq |v_i(l) - t|$, $i = 1, \dots, N$.

Step 3. Adjust the position of the winning v_k and its individual learning rate:

$$\begin{aligned} v_k(l+1) &= v_k(l) + s(l)\alpha_k(l)(t - v_k(l)), \\ \alpha_k(l+1) &= \alpha_k(l)/(1 + s(l)\alpha_k(l)), \end{aligned}$$

where $s(l) = 1$ if the classification is correct, i.e., $c = c_k$, and $s(l) = -1$ otherwise.

ALGORITHM OLSQ*m*

To refine the position of V, we have used another modification of LVQ2, namely OLSQm, which deals only with incorrectly classifying prototypes, i.e., $v_k(l+1) = v_k(l) - \alpha_k(l)(t - v_k(l))$, if $c_k \neq c$, otherwise do nothing.

Initialization of the codebook V can be performed in many different ways (Diamantini and Spalvieri, 1998; Kohonen, 1995). The prototypes can be randomly chosen from the training set. A better solution is to organize the codebook vectors by SOM or K-means, (for each class separately or for all training sequences (Diamantini and Spalvieri, 1998)), but the reference vectors can also be initialized by other algorithms, e.g., by finding a group of vectors which satisfy the K-NN riterion (Kohonen, 1995; Skubalska-Rafajłowicz and Krzyżak, 1996).

4.1. Compression of the Codebook Procedure

V can be sorted, without further loss of generality, in such a way that $v_i \leq v_{i+1}$. Thus, in one-dimensional problems, the boundaries between decision regions are given by points $(v_i + v_{i+1})/2$, $c_i \neq c_{i+1}$, $i = 1, 2, \dots, N-1$. The codebook which contains N reference positions produces a decision rule which consists of at most N subintervals. Moreover, each reference position v_i with neighbors in the same class, i.e., such that $c_i = c_{i-1} = c_{i+1}$, can be removed from V. It is easy to see that such a compression of V does not change the classification rule induced by V.

5. Examples

The data in the examples presented here have been transformed to one dimension using the pseudoinverse of the Peano space-filling curve (Milne, 1980; Sagan, 1994).

In order to estimate the error rates of the classifier, the resubstitution and leave-one-out method (Fukunaga, 1972) (averaged on 10 quantizers) were used for the real data example. In the simulated situation 5 independent training series were performed for each choice of the codebook length and the starting value of the learning parameter in the OLSQ procedure. For each of these, an additional independent test sample consisting of 2000 observations was generated.

The relative number of prototypes per class is chosen consistently with the ratio between the different *a-priori* probabilities of classes for both the real and simulated data.

Example 1. (Real data example—Iris data) The Iris data are measurements of three species of iris: iris setosa, iris versicolor and iris virginica, consisting of 50 samples each, with 4 measurements (Andrews and Hertzberg, 1985; Friedman, 1994). The measurements are the septal length and width, and the petal length and width.

The best error rate obtained with the k-NN classifier (in 1-D) is about 2.66% (Skubalska-Rafajłowicz and Krzyżak, 1996) and 5.33% with the k-NN classifier (Cover and Hart, 1967) based on the original data (in 4 dimensions) (Friedman, 1994). The error rates obtained with OLSQ1 and OLSQm (used in the sequel) are given in Table 1.

Table 1. Averaged classification errors obtained by the OLSQ Algorithm with the Peano curve applied to the iris benchmark.

	% of misclassification		
No. of prototypes in one dimension	resubstitution method	leave-one-out method	best vs. worst result in the l-o-o method
12	4.0%	4.0%	4.0%–4.0%
30	3.1%	4.5%	2.66%–5.3%
60	2.4%	3.9%	2.0%–5.3%

Furthermore, it turns out that the number of prototypes can be reduced to at most 16 (with $N = 60$) using the compression procedure. For comparison, with the classical LVQ (in 4-D) the best (average) result obtained was the 5% error rate for the codebook size of 10 (multidimensional) prototypes (Blayo *et al.*, 1995).

Example 2. (Simulated data—two triangles) The data were generated using a uniform distribution over the square I_2. There are $d = 2$ input variables and two classes with the prior probability 1/2. The classes are defined by $x_1 + x_2 \leq 1 \Rightarrow$ class 0, otherwise $\Rightarrow$ class 1.

The diagonal, which discriminates the classes, forms a Cantor-type set in I_1 after transformation by the Peano curve. The example is rather sophisticated and was aimed at examining an extremely difficult problem with a known optimal solution. It should be mentioned that application of the Sierpiński curve leads to a very simple classification problem on I_1, with one discriminating point only. The theoretical Bayes error is 0%. Results of experiments (averaged error rates) are shown in Figs. 5 and 6.

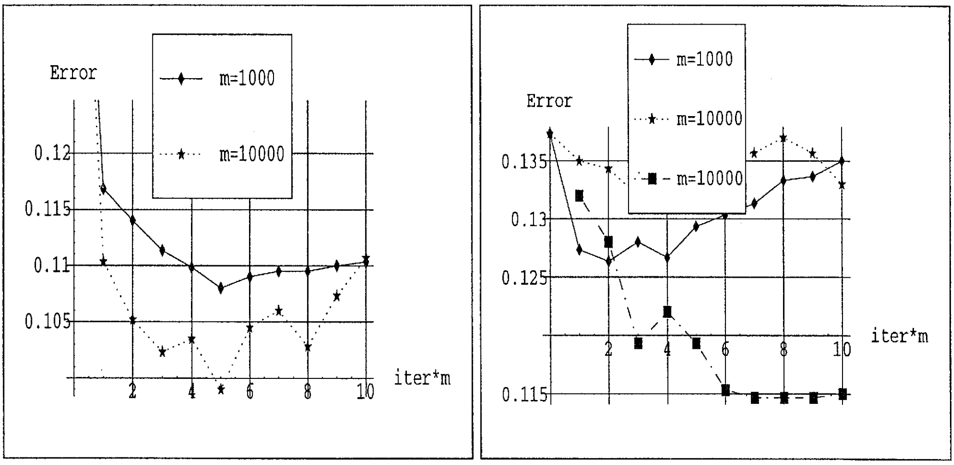

Fig. 5. Average rate of misclassifications obtained by the OLSQ algorithm (with $\alpha = 0.1$) vs. the number of iterations obtained for Example 2 and $N = 18$. Left panel: initialization from the learning sequence, OLSQ1 method; right panel: initialization close to a local optimum, OLSQ1 followed by OLSQm (squares).

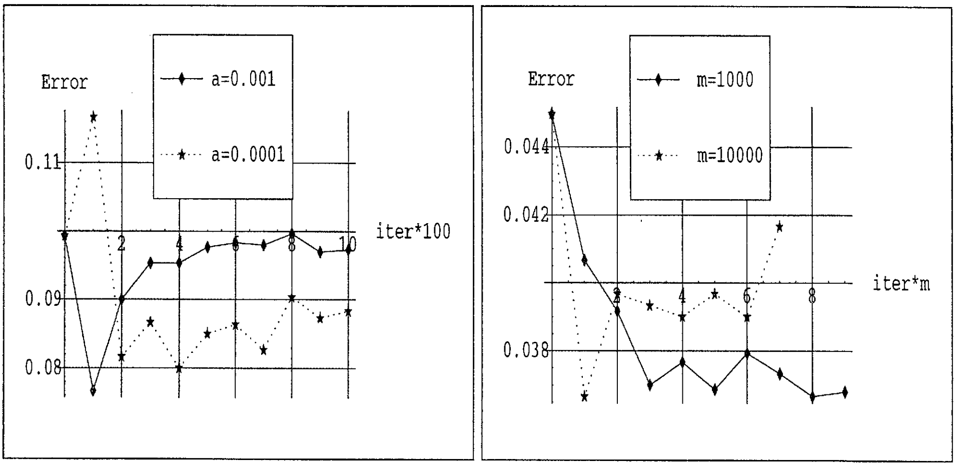

Fig. 6. Average rate of misclassifications obtained by the OLSQ1 algorithm vs. the number of iterations obtained for Example 2. Left panel: $N = 18$, initialization by SOM, OLSQ1 with a different starting value of the learning rate α; right panel: $N = 162$, initialization from the learning sequence, OLSQ1 with $\alpha = 0.1$.

Example 3. (Simulated data—a multidimensional ball) The data were generated using a standard normal distribution. Here there are $d = 10$ input variables (i.i.d. standard normal) and two classes with the prior probability $1/2$. The classes are defined by

$$\sum_{i=1}^{10} x_i^2 \leq 9.8 \Rightarrow \text{class}\,0, \quad \text{otherwise} \Rightarrow \text{class}\,1.$$

This example was taken from (Friedman, 1994) and is seen as more favorable to the k-NN procedure. The theoretical (Bayes) error rate is 0.0%, but the problem is rather difficult to deal with.

The error rate obtained by the k-NN procedure in $d = 10$ dimensions and based on 500 observations is about 0.34. The best result reported by Friedman (1994) is about 0.26 and was obtained with the machette method, a procedure which is hybrid between the regular k-nearest neighbor method and the tree-structured recursive partitioning technique, cf. (Friedman, 1994) for details. A similar result, i.e., 0.26 with standard deviation 0.01, can be attained using the LSQ algorithm with 40 neurons (20 prototypes per class).

The number of neurons can be reduced to about 25 using the compression procedure. Increasing the codebook size does not improve the performance of the classifier.

6. Concluding Remarks

The pattern recognition algorithm based on transforming data by SFCs and applying the LVQ-type algorithm has been discussed. Essential features of the proposed approach are: easy learning, fast recognition of testing patterns and a large degree of compression of the codebook. Data preprocessing of one pattern based on a space-filling curve can be performed in $O(d)$ time, i.e. proportionally to the dimension of the data (the number of features). Many interesting problems which seem to be worth of further investigations are left beyond the scope of this paper. In particular, one may consider application of the preprocessing of multidimensional data, using a space-filling curve based transformation into the unit interval, to the analysis of properties LVQ-based classification algorithms, which is easier to perform in one dimension.

References

Andrews D.F. and Hertzberg A.M. (1985): *Data. A Collection of Problems for Many Fields for Student and Research Worker.* — Berlin: Springer-Verlag.

Bartholdi J.J. and Platzman L.K. (1988): *Heuristics based on space-filling curves for combinatorial problems in Euclidean space.* — Management Sci., Vol.34, No.3, pp.291–305.

Blayo F., Cheneval Y., Guérin-Dugué A., Chentouf R., Aviles-Cruz A., Madrenas J., Moreno M., Voz J.L. (1995): *Enhanced Learning for Evolutive Neural Architecture.* — ESPRIT Basic Research Report 6891, UCL – Univ. Cath. Louvain, `http://www.dice.ucl.ac.be/neural-nets/Research/Project/ELENA`

Cover T.M. and Hart P.E. (1967): *Nearest neighbor pattern classification.* — IEEE Trans. Inform. Theory, Vol.13, No.1, pp.21–27.

Diamantini C. and Spalvieri A. (1998): *Quantizing for minimum average misclassification risk.* — IEEE Trans. Neural Netw., Vol.9, Vol.1, pp.174–182.

Friedman J.H. (1994): *Flexible Metric Nearest Neighbor Classification.* — Techn. Rep. Stanford University.

Fukunaga K. (1972): *Introduction to Statistical Pattern Recognition.* — Academic Press: New York.

Hilbert D. (1891): *Ueber die stetige abbildung einer linie auf ein flaechenschtueck.* — Math. Ann., Vol.38, pp.459–469.

Kangas J., Kohonen T. and Laaksonen J. (1990): *Variants of self-organizing maps.* — IEEE Trans. Neural Netw., Vol.1, No.1, pp.93–99.

Kohonen T. (1990): *Self-organizing maps.* — Proc. IEEE, Vol.78, No.9, pp.1464–1480.

Kohonen T. (1995): *Self-Organizing Maps.* — Berlin: Springer Verlag.

Kulkarni S.R., Lugosi G. and Venkatesh S. (1998): *Learning pattern classification — A survey.* — IEEE Trans. Inform. Theory, Vol.44, No.6, pp.2178-2206.

Lamarque C.H. and Robert F. (1996): *Image analysis using space-filling curves and 1D wavelet basis.* — Pattern Recogn., Vol.29, No.8, pp.1309–1322.

Milne S.C. (1980): *Peano curves and smoothness of functions.* — Adv. Math., Vol.35, No.1, pp.129–157.

Patrick E.A. (1972): *Fundamentals of Pattern Recognition.* — Englewood Cliffs, NJ: Prentice-Hall.

Patrick E.D., Anderson D.R. and Bechtel F.K. (1968): *Mapping multidimensional space to one dimension for computer output display.* — IEEE Trans. Comput., Vol.C-17, No.10, pp.949–953.

Peano G. (1890): *Sur une courbe qui remplit toute une aire plane.* — Math. Ann., Vol.36, pp.157–160.

Platzman L.K. and Bartholdi J.J. (1989): *Space-filling curves and the planar traveling sales man problem.* — J. ACM, Vol.36, No.4, pp.719–737.

Quweider M.K. and Salaeri E. (1995): *Peano scanning partial distance search for vector quantization.* — IEEE Signal Process. Lett., Vol.2, No.9, pp.170–171.

Sagan H. (1994): *Space-Filling Curves.* — Berlin: Springer-Verlag.

Sierpiński W. (1912): *Sur une nouvelle courbe continue qui remplit toute une aire plane.* — Bulletin de l'Acad. des Sciences de Cracovie A., pp.463–478.

Skubalska-Rafajłowicz E. (1994): *The closed curve filling multidimensional cube.* — Techn. Rep. No.46/94, ICT, Technical University of Wrocław, Poland.

Skubalska-Rafajłowicz E. (1997a): *Applications of the space-filling curves with data-driven measure-preserving property.* — Nonlinear Anal. Theory Methods Appl., Vol.30, No.3, pp.1305–1310.

Skubalska-Rafajłowicz E. (1997b): *Space-filling curves and Kohonen's 1-D SOM as a method of a vector quantization with known asymptotic level density.* — Proc. 3rd Conf. *Neural Networks and Their Applications*, Kule, Poland, pp.161–166.

Skubalska-Rafajłowicz E. (1999): *Pattern recognition algorithm based on space-filling curves and orthogonal expansion.* — (submitted).

Skubalska-Rafajłowicz E. and Krzyżak A. (1996): *Fast k-NN classification rule using metric on space-filling curves.* — Proc. 13th Int. Conf. *Pattern Recognition*, Vienna, Austria, Vol.2, pp.221–225.

Steele J.M. (1989): *Efficacy of space-filling heuristics in Euclidean combinatorial optimization.* — Opers. Res. Lett., Vol.8, No.5, pp.237–239.

Stevens R.J., Lehar A.F. and Preston F.H. (1983): *Manipulation and presentation of multidimensional image data using the Peano scan.* — IEEE Trans. Pattern Anal. Mach. Intell., Vol.5, No.5, pp.520–525.