Some methods of replacing the nearest neighbor method

Tomasz Górecki, Maciej Luczak

Some methods of replacing the nearest neighbor method

Journal:	*Communications in Statistics - Simulation and Computation*
Manuscript ID:	LSSP-2008-0285.R2
Manuscript Type:	Original Paper
Date Submitted by the Author:	19-Sep-2009
Complete List of Authors:	Górecki, Tomasz; Adam Mickiewicz University, Faculty of Mathematics and Luczak, Maciej; Koszalin University of Technology, Department of Civil and Environmental Engineering
Keywords:	nearest neighbor method, classification, classifiers comparison
Abstract:	In this paper two classifiers, which generalize the nearest neighbor method, are introduced and studied. The first of them is based on calculating the distances to all objects from a learning sample. The second one additionally considers directions of the objects. Both of them have locally nonlinear classification borders. A number of real and artificial datasets and methods of error estimation are used.
Note: The following files were submitted by the author for peer review, but cannot be converted to PDF. You must view these files (e.g. movies) online.	
final.zip	

SOME METHODS OF REPLACING THE NEAREST NEIGHBOR METHOD

TOMASZ GÓRECKI[1,3] AND MACIEJ ŁUCZAK[2]

Abstract. In this paper two classifiers, which generalize the nearest neighbor method, are introduced and studied. The first of them is based on calculating the distances to all objects from a learning sample. The second one additionally considers directions of the objects. Both of them have locally nonlinear classification borders. A number of real and artificial datasets and methods of error estimation are used.

1. Introduction

The nearest neighbor method is very popular among researchers using classification methods. Information about distributions of data is not needed in this method. The result of classification only depends on the learning object with the shortest distance to the test object, but the value of the distance is not taken into consideration. Other objects of the learning sample have no influence on the classification. The classification in the generalized method of the nearest neighbor (which is called k-nearest neighbor method, kNN) depends on k objects of training set. However, only the order of distances, not values or their directions, is important (Duda et al. (2001)). kNN is nonlinear classifier but the decision boundaries of kNN are locally linear segments. However in general they have a complex shape that is not equivalent to a line in 2D or a hyperplane in higher dimensions. If a problem is nonlinear (**as most problems**) and its class boundaries cannot be approximated

Key words and phrases. nearest neighbor method.

[1] Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Umultowska 87, 61-614 Poznań, Poland

[2] Department of Civil and Environmental Engineering, Koszalin University of Technology, Śniadeckich 2, 75-453 Koszalin, Poland

[3] Corresponding author. Email: drizzt@amu.edu.pl

well with linear hyperplanes, then nonlinear classifiers are often more accurate than linear classifiers. Nonlinear classifiers are more powerful than linear **ones**. For some problems, there exists a nonlinear classifier with zero classification error, but no such linear classifier.

It seems that consideration of values of distances between a test object and all learning objects can have positive influence on the classification result. It might be interesting not only which object is nearer than another, but how much nearer it is. Also directions of objects can influence the classification process. In the paper we introduce and study two parametric families of classifiers, which fulfill, in a way, above assumptions. The first classifier considers distances of a test object to all objects in the training sample. Influence of the learning objects drops when the value of the distance to the test object rises. Moreover, the second studied classifier considers directions of objects in the training set. The idea of the methods derives from some mathematical maps in physics such as (gravitational, magnetic) field strength for the first method and fields of vectors (gravitational, magnetic force) for the second one. The classification borders of the classifiers are globally and locally nonlinear.

In the paper the methods are compared to the nearest neighbor method and the error of classification is regarded. The methods are also compared to each other, the number of wins is considered. Many real and artificial datasets are used. Classification errors are estimated by a few methods: leave-one-out cross-validation, 10-fold cross-validation, bootstrap sampling, test datasets. The results of the research are explained with a number of charts (bar, circle, contour), where differences between the classifiers are shown accurately.

In our paper first we present the main ideas in the view of introducing methods in machine learning (Section 2). Then we describe artificial and real datasets used in our researches (Section 3). In Section 4 we describe experimental setup and we present results of our experiments with artificial and real datasets. **We conclude with discussion in Section 5.**

2. Methods

Suppose that a training sample has been collected by sampling from a population P consisting of C subpopulations or classes $G_1, \dots, G_C$. The ith observation is a pair denoted by (x_i, y_i), where x_i is a d-dimensional feature vector and y_i is the label for recording class membership. The corresponding pair for an unclassified observation is denoted by (x, y). In this case x is observed but the class label y is unobserved. The object of classification is to construct a classification rule for predicting the membership of an unclassified feature vector $x \in P$. An automated classifier can be viewed as a method of estimating the posterior probability of membership in G_j. The classification rule assigns x to the group with the largest posterior probability estimate. We denote the posterior probability of membership in G_j by

$$p_j(x) = P(y = j|x).$$

Let us consider the Nearest Neighbor Classifier (1NN). A new object is assigned to the class to which the nearest object from the training sample belongs. First, for a test observation x, we find its nearest neighbor among the observations from the training sample:

$$k(x) = \arg\min_i \|x_i - x\|,$$

where $k(x)$ is the index of the nearest neighbor. After that, we observe the label of the object and classify it to the corresponding class:

$$d_{1\text{NN}}(x) = y_{k(x)}.$$

We **generally** use Euclidean norm $\|\cdot\|$ to compute distances in $\mathbb{R}^d$ space.

In this method there is only one training object (the nearest one) on which the classification depends.

Other methods might consider distances from a test observation to all training observations. We can construct such method, where all objects from a training sample are important for classification and the greater the distance between a test object and a training one is, the less important it is for the result of the classification.

For a given test observation x we compute a sum of some functions of distances for all observations from the jth class:

$$\rho_j(x) = \frac{1}{n_j} \sum_{x_i \colon y_i = j} \frac{1}{\|x_i - x\|^\alpha} \qquad \alpha \in \mathbb{R}^+.$$

Here, α is a nonnegative constant parameter, n_j is the number of elements in the class j. Then we assign the observation x to the class whose sum is the largest:

$$d_{\mathrm{SC}\alpha}(x) = \arg\max_j \rho_j(x).$$

The idea of the method derives from potential functions (in physics). Potential functions are scalar functions, so in this paper we call the method Scalar Classifier and denote by SC (SCα if the parameter α is fixed).

The above method depends on distances between observations but does not depend on directions. We construct a new classification method, in which directions of objects are as important for the classification result as distances.

Suppose, we are given a testing observation x. For each class j we construct a vector $v_j(x)$ which is a sum of vectors linking the object x with objects from the training sample. The greater distance to a training object is, the shorter the vector is:

$$v_j(x) = \frac{1}{n_j} \sum_{x_i \colon y_i = j} \frac{\mathrm{v}(x, x_i)}{\|x_i - x\|^\alpha} \qquad \alpha \in \mathbb{R}^+,$$

where α is a parameter and $\mathrm{v}(x, x_i)$ is a versor with the beginning at the point x and the direction outlined through the points x and x_i (from x to x_i), i.e.

$$\mathrm{v}(x, x_i) = \frac{x_i - x}{\|x_i - x\|}.$$

Note that the function of distance $\frac{1}{\|x_i - x\|^\alpha}$ is the same as for Scalar Classifier. Thus, we can write the equation in the following form:

$$v_j(x) = \frac{1}{n_j} \sum_{x_i:\, y_i = j} \frac{x_i - x}{\|x_i - x\|^{\alpha+1}} \qquad \alpha \in \mathbb{R}^+.$$

We assign the observation x to **the** class whose vector $v_j(x)$ is the longest:

$$d_{\mathrm{VC}\alpha}(x) = \arg\max_j \|v_j(x)\|.$$

The idea of the above method derives from field of vectors (in physics). Since $v_j(x)$ is a vector we call this classification method Vector Classifier and denote by VC (VCα if the parameter α is fixed).

In this way constructed classifier considers not only distances between objects but also their positions (Fig. 1). Note, if two training objects are situated on a line with the same distance from a testing object but with opposite directions then the sum of them is a zero vector and it has no influence on classification.

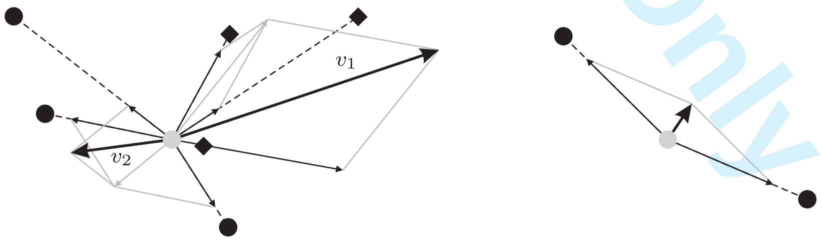

FIGURE 1. Vector Classifier

If both training and test sample include the same observation then the denominator in equations for SC and VC is zero. Then we make a standard assumption $\frac{1}{0} = \infty$ and perform all computations on the extended real number line.

The result of classification by SC and VC is a label of a class. The use of data allows us to compute posterior probability directly, using simply the Bayesian rule. Usually the **prior** probability π_i of each class are known. Under the assumption that the **priors are** unknown, we may assume uniform **priors**. In this case using Bayes' rule, we can form posterior probabilities in the following way (for SC and VC, respectively):

$$p_j(x) = \frac{\rho_j(x)}{\sum\limits_{k=1}^{C} \rho_k(x)}, \qquad p_j(x) = \frac{\|v_j(x)\|}{\sum\limits_{k=1}^{C} \|v_k(x)\|} \qquad j = 1, 2, \dots, C.$$

If there are infinities in above equations, we make a nonstandard assumption $\frac{\infty}{\infty} = 1$ (we assume also that there is no observation which belongs to different classes at the same time).

All the three classification methods (1NN, SC, VC) have a number of the same properties: Classification result is not depended on isometries of $\mathbb{R}^d$ space (translations, rotations, symmetries) and scaling (for SC and VC it arises from equation $\sum \frac{1}{\|tx_i - tx\|^\alpha} = \frac{1}{|t|^\alpha} \sum \frac{1}{\|x_i - x\|^\alpha}$, $t \in \mathbb{R}$); Each observation from the training sample is classified correctly.

Methods SC and VC depend on a parameter α. The greater the parameter is, the less observations influence the classification. Particulary, if $\alpha \to \infty$ then influence of further points is more and more slight, and finally the classification depends only on the nearest observations in each class. This means that, for **large values of parameter** α methods SC and VC are equivalent to 1NN classifier. More strictly, for any pair of training and test sample there is a parameter $\alpha \in \mathbb{R}^+$ that methods SC and VC give the same classification result as 1NN classifier (Fig. 2–4).

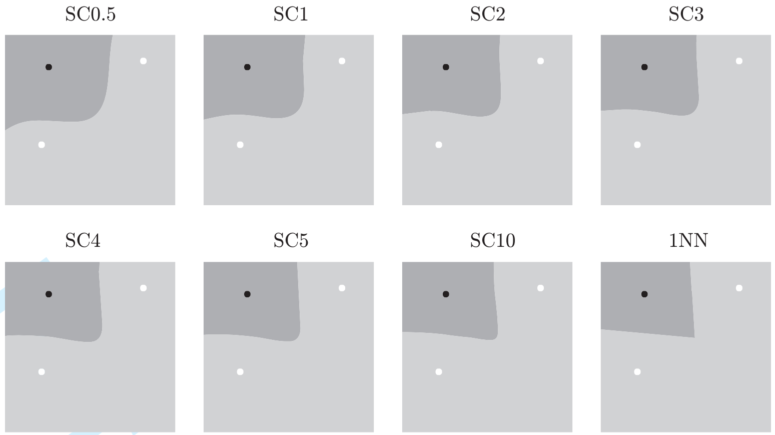

FIGURE 2. Variability of classification border of Scalar Classifier depending on parameter α

If the parameter α approaches zero, methods SC and VC have different behavior. If $\alpha = 0$, then the function of distance ρ_j is equal to 1. Therefore, for SC, posterior probabilities are the same and equal to $\frac{1}{C}$, but the classification error does not tend to $\frac{C-1}{C}$, it is far smaller (Fig. 3).

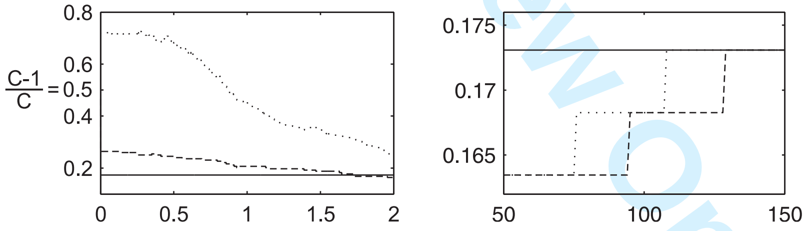

FIGURE 3. Behavior of classification error for very small and very **large** values of parameter α. *Sonar* set (2 classes), leave-one-out cross-validation error estimation, — 1NN, – – SC, · · · VC, y-axis: error rate, x-axis: α parameter

However for VC, v_j is a sum of versors and its length depends only on directions of training observations. For small values of α behavior of this classifier is a bit chaotic, the error rises fast. If $\alpha \to 0$, then the classification error tends to an error for $\alpha = 0$ (Fig. 3).

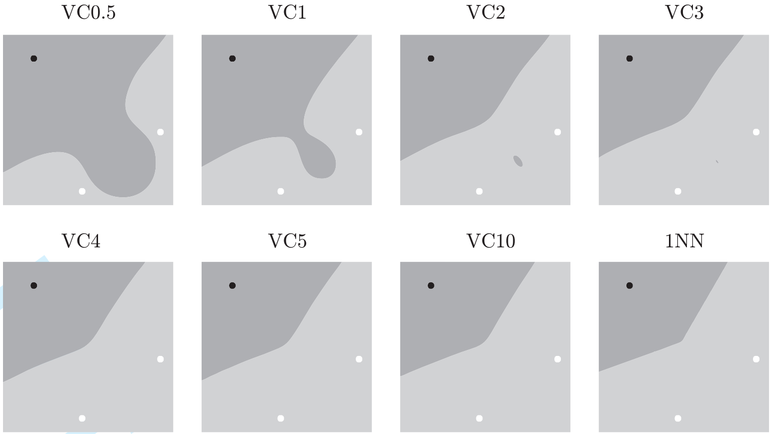

Figure 4. Variability of classification border of Vector Classifier depending on parameter α

In practice, we select an appropriate value of the parameter α by cross-validation method.

Scalar Classifier seems to be a smoother version of Nearest Neighbor Classifier. Vector Classifier sometimes generates ”islands” in classification area between points of the same class (Fig. 5). Figures 2, 4, 5 clearly show that Scalar and Vector methods have locally nonlinear classification borders.

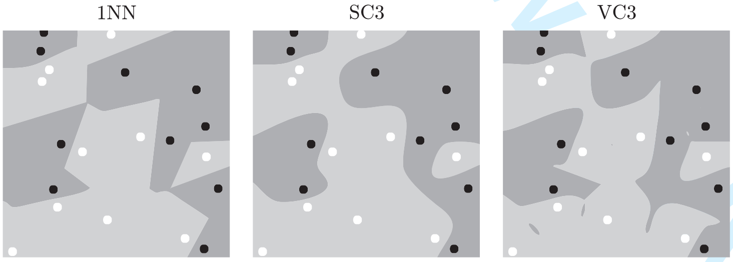

Figure 5. Comparison of classification areas of discussed classifiers

3. Datasets

3.1. **Artificial datasets.** To test our methods we carried out experiments with 8 artificial datasets. Information about 6 first used artificial datasets are presented in Table 1. First, second and third dataset comes from (Fukunaga, 1990).

Table 1. Information about artificial datasets (N – normal distribution, U – uniform distribution, C – Cauchy distribution).

Name	I class distribution	II class distribution
1	$N_5([0_5]', I_5)$	$N_5([2, 0_4]', I_5)$
2	$N_8([0_8]', I_8)$	$N_8([0_8]', 4I_8)$
3	$N_8([0_8]', I_8)$	$N_8([\mu, \Sigma)$
4	$N_2([0_2]', I_2)$	$U_2([-6, 6] \times [-6, 6])$
5	$N_2([0_2]', I_2)$	$C_2([2, 2]', [1, 1]')$
6	$U_2([0, 0.9] \times [0, 1])$	$U_2([0, 0.8] \times [0, 1])$

The seventh dataset is 2-dimensional with 2 classes. The first class is a circle in the middle of a square, the second class is the square without the circle. The side of the square is equal to 500 and radius of circle is equal to $\frac{500}{\sqrt{2\pi}}$ so that the distribution is uniform, and the support of probability is a circle (the first class) and a square without a circle (the second class).

The last dataset is 3-class *waveforms* data and was taken from (Breiman et al. (1984)). Each class consists of a random combination of two of waveforms $h_1(t)$, $h_2(t)$ and $h_3(t)$ sampled at the integers with noise added. The measurement vectors are 21 dimensional. To generate data we first independently generate a uniform random number u and 21 random numbers $\varepsilon_1, \ldots, \varepsilon_{21}$ normally distributed with mean 0 and variance 1. Then set

$$x_{1i} = uh_1(i) + (1-u)h_2(i) + \varepsilon_i, \;\; i = 1, \ldots, 21 \text{ for I class},$$

$$x_{2i} = uh_1(i) + (1-u)h_3(i) + \varepsilon_i, \;\; i = 1, \ldots, 21 \text{ for II class},$$

$$x_{3i} = uh_2(i) + (1-u)h_3(i) + \varepsilon_i, \;\; i = 1, \ldots, 21 \text{ for III class}.$$

Some examples of used datasets are presented in Figure 6.

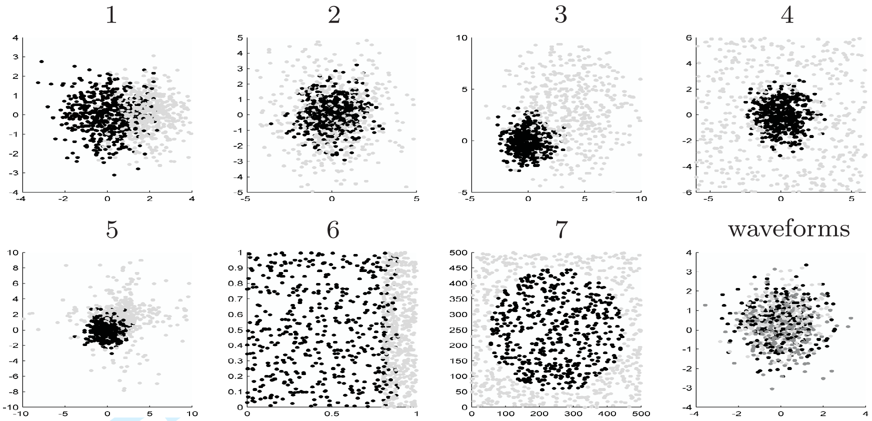

Figure 6. Artificial datasets (only two first features).

3.2. **Real datasets.** We performed experiments also on 10 real datasets. Information about used datasets are presented in Table 2.

Table 2. Information about used real datasets.

Name	Number of features	Number of classes	Number of instances in classes	Number of all instaces
beetles	2	3	21,21,22	64
fish	4	3	12,12,12	36
football	6	3	30,30,30	90
glass	9	6	70,76,17,13,9,29	214
ionosphere	34	2	225,126	351
iris	4	3	50,50,50	150
sonar	60	2	111,97	208
thyroid	5	3	150,35,30	215
turtles	6	2	24,24	48
wine	13	3	59,71,48	178

Datasets *glass*, *ionosphere*, *iris*, *sonar*, *thyroid* and *wine* originate from Merz and Murphy (1998). The dataset *beetles* comes from Seber (1984), *fish* from Hawkins and Rasmussen (1978), *football* from Gleim (1984) and *turtles* from Statistica (2001) program repository.

4. Results

4.1. **Experimental setup.** In case of artificial datasets we did experiments for various sizes of learning sample. We carried out for learning samples:

$N = 5, 10, 20, 50, 100, 200, 500, 1000$ observations. Depending on this size, we fixed the size of the test sample as $5CN$ (C – number of classes). In the test samples, the sizes of classes were equal. For the purpose of the assessing the classification error, repeated the experiment the appropriate number of times, which depended on the learning sample size in the following way: $500000/N$. In case of real datasets we used bootstrap, leave-one-out and 10-fold cross-validation methods to estimate classification error rate. Number of bootstrap samples was equal to 1000. In case of 10-fold CV we regarded 1000 repetitions and final result was a mean error rate. We did experiments for the following α parameters in vector and scalar method: $0.5, 1, 2, 3, 4, 5, 10$. We also carried out the number of wins of SC and VC method (real datasets – bootstrap samples, artificial datasets – test samples). Especially, we consider the difference between percentage of wins of VS and SC method: $\frac{n_{\mathrm{VC}}}{n} - \frac{n_{\mathrm{SC}}}{n}$, where n_{VC} and n_{SC} – numbers of wins of VC and SC method respectively, and n – number of elements in a sample.

4.2. **Datasets results.** Generally, researches showed that SC and VC methods are better than 1NN method for many values of the parameter α and on almost all data sets. Tab. 3 and Tab. 4 present overall results. For each method of error estimation and each classifier we choose the best value of α, this means the α for that the classifier has the lowest error rate in the method.

Table 3. Classification error rate for real datasets. For each method of error estimation (CV – leave-one-out, 10CV – 10 fold cross validation, boot – bootstrap) and for each dataset the best value of α parameter was chosen

		beetles	fish	football	glass	ionosphere	iris	sonar	thyroid	turtles	wine
CV	1NN	6.76	55.56	40,00	26.64	13.39	4,00	17.31	5.12	14.58	23.03
	SC	5.41	44.44	35.56	29.91	15.95	3.33	14.9	3.72	12.5	23.03
	VC	5.41	52.78	34.44	30.37	16.52	3.33	14.9	4.65	14.58	23.6
10CV	1NN	6.57	53.38	39.85	26.9	13.51	4.1	17.67	5.31	14.83	23.72
	SC	5.52	42.76	35.09	30.02	16.26	3.64	15.08	4.49	13.64	23.31
	VC	5.51	52.23	36.43	30.41	16.64	3.5	15.59	4.47	14.82	24.33
boot	1NN	6.48	53.04	41.93	30.05	14.46	4.49	19.06	6.49	18.03	27.07
	SC	6.09	45.83	35.65	32.09	16.56	3.85	17.31	5.98	17.88	26.34
	VC	6.21	49.27	39.24	32.27	17.13	3.94	18.14	6.11	18.65	26.92

Table 4. Classification error rate for artificial datasets. For each training sample size and for each dataset, the best value of α parameter was chosen

		5	10	20	50	100	200	500	1000
1	1NN	35.01	26.67	28.71	27.39	25.03	24.69	23.76	23.66
	SC	31.43	19.57	21.04	20.11	16.44	15.98	16.19	16.07
	VC	33.31	24.16	26.32	24.87	21.56	20.52	19.38	18.98
2	1NN	42.74	39.9	37.87	35.08	30.86	28.5	24.97	22.61
	SC	43.46	41.53	40.56	39.19	36.23	34.34	30.6	27.34
	VC	43.64	41.96	41.25	40.32	38.03	36.8	34.2	31.8
3	1NN	20.27	10.73	10.62	6.16	6.18	5.36	4.52	4.25
	SC	19.38	10,00	10.31	5.71	5.98	4.88	3.98	3.66
	VC	19.66	10.32	10.64	6.04	6.29	5.21	4.3	3.97
4	1NN	30.96	25.37	22.9	16.36	15.07	14.07	13.9	14.01
	SC	31.11	25.52	20.46	16.18	14.23	12.64	11.88	11.52
	VC	31.19	25.74	20.79	16.51	15.1	14.07	13.94	14.01
5	1NN	24.98	17.74	21.2	15.25	16.05	14.28	14.08	13.87
	SC	24.91	17.28	22.39	15.14	16.72	14.53	13.89	13.07
	VC	26.14	18.21	22.39	15.65	17.06	14.77	14.29	13.98
6	1NN	31.78	17.17	16.37	13.91	13.5	10.7	10.33	9.97
	SC	30.58	14.63	14.93	12.78	12.6	9.75	9.11	8.57
	VC	31.24	16.83	16.34	13.94	13.57	10.79	10.37	10,00
7	1NN	37.71	30.52	22.24	12.98	8.2	5.66	3.59	2.57
	SC	37.15	30.03	21.96	12.65	7.74	5.22	3.09	2.1
	VC	37.06	30.32	22.22	13.06	8.28	5.72	3.62	2.59
Waveforms	1NN	42.31	33.34	30.35	28.04	24.94	24.63	22.81	22.75
	SC	40.77	29.61	25.46	22.65	18.86	18.26	16.77	16.34
	VC	41.28	31.28	27.67	24.59	20.63	19.88	17.5	16.94

There are few exceptions — the sets on which SC and VC methods have greater classification error than 1NN method for all values of parameter α (Fig. 7). In those cases the error decreases as the value of α increases.

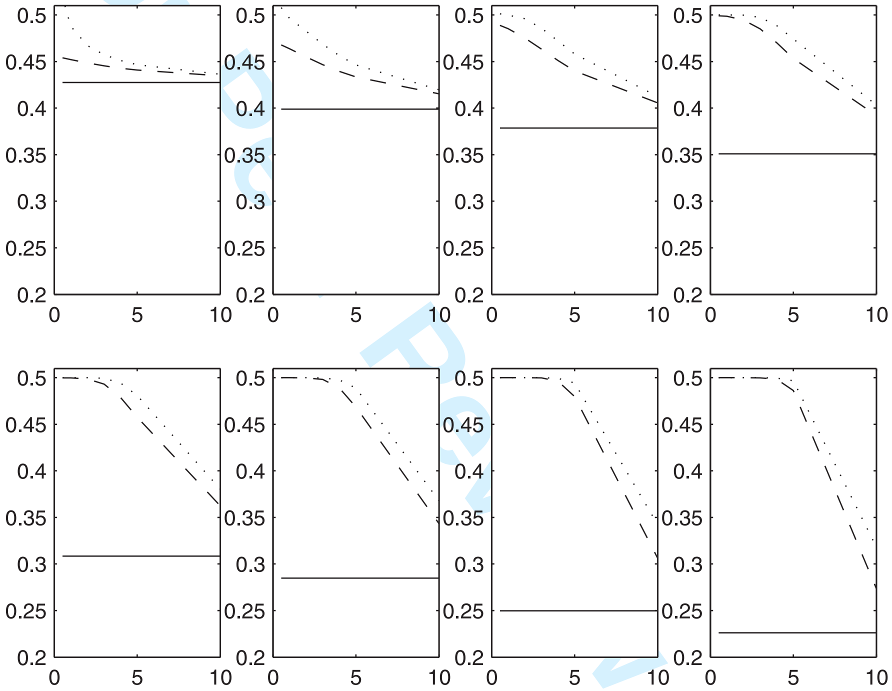

Figure 7. Shape of error lines for artificial dataset 2 (in the order of training sample size), — 1NN, −− SC, ··· VC, y-axis: error rate, x-axis: α parameter

On data sets, where SC and VC are better than 1NN, we can see a very distinctive shape of the error line. As the parameter α increases the error of classification drops to a minimum lower than 1NN-error, and then asymptotically increases to the 1NN-error value (Fig. 8).

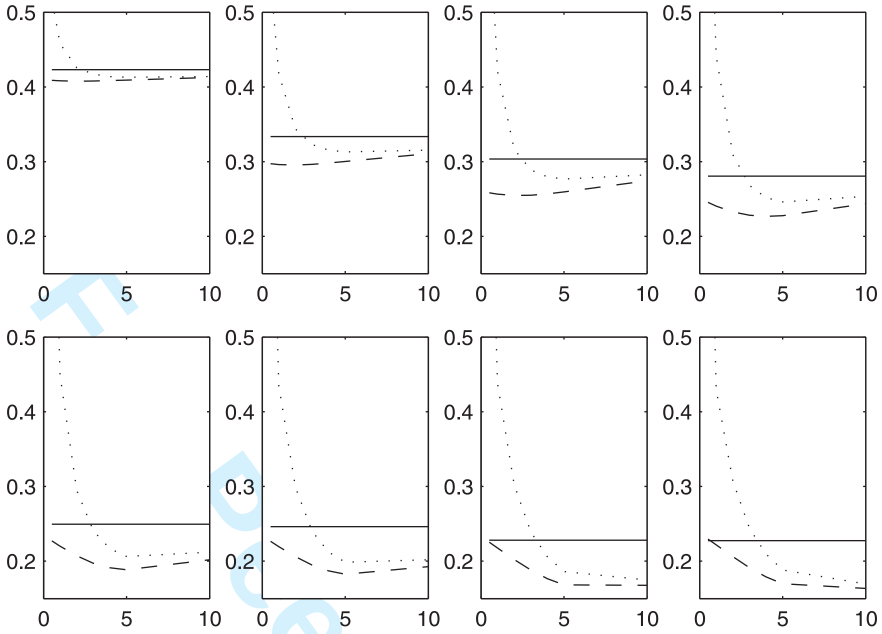

FIGURE 8. Shape of error lines for dataset waveforms (in the order of training sample size), — 1NN, −− SC, ··· VC, y-axis: error rate, x-axis: α parameter

On the real data sets, all methods of error estimation follow the shape of the error line. The value of parameter α, for which the error is minimal, is determined identically by all error estimators (Fig. 9).

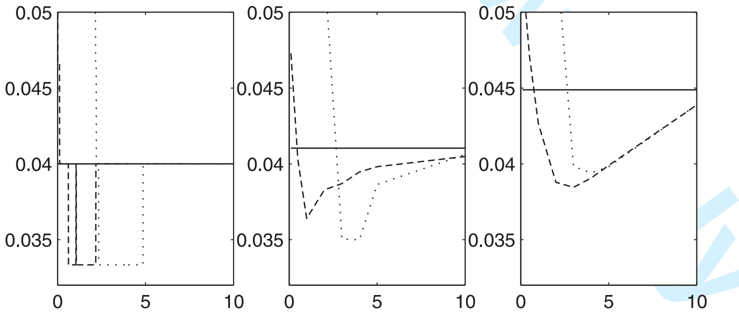

FIGURE 9. Shape of error lines for dataset *iris* (left figure: CV, middle: 10CV, right: boot), — 1NN, −− SC, ··· VC, y-axis: error rate, x-axis: α parameter

It is confirmed that the greater the parameter α is, the more similar to 1NN method SC and VC methods are. If α tends to zero, the SC method error settles on a level that is (much) lower than value of the error for $\alpha = 0$. The error of VC method rises very fast for small values of the parameter α and seems to tend to the error level for $\alpha = 0$ (Fig. 3).

For fixed α SC error is rather lower than VC error. We can see that for many artificial data sets. It seems that the smaller a data set is, the better VC method is in comparison to SC method. For very small data sets, it sometimes happens that VC method (for some fixed α) is better than SC method independently of the value of its parameter α (Fig. 10, top-left).

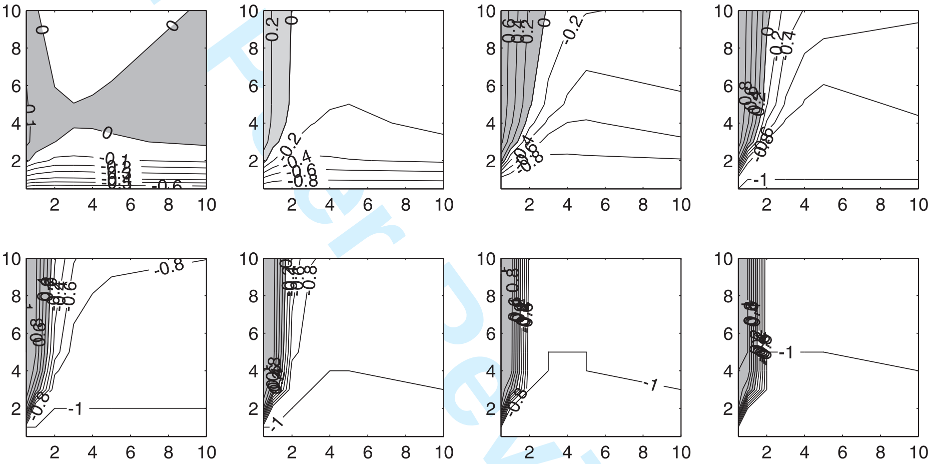

FIGURE 10. Difference between the percentage of VC and SC method wins for dataset 7 (in the order of training sample size), x-axis: α parameter of SC, y-axis: α parameter of VC. Gray area means that VC is better than SC.

On a few real datasets, the error estimators sometimes also show that VC method is better than SC (Fig. 11).

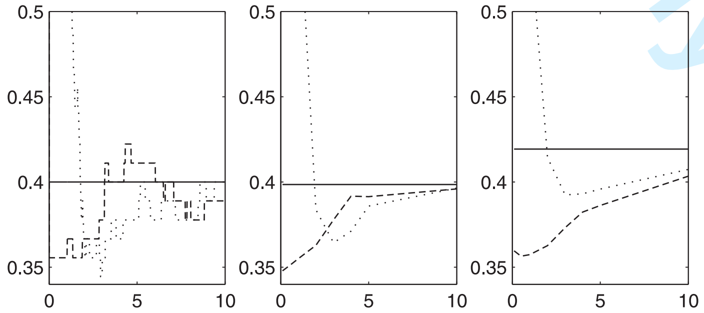

FIGURE 11. Shape of error lines for dataset *football* (left figure: CV, middle: 10CV, right: boot), — 1NN, – – SC, · · · VC, y-axis: error rate, x-axis: α parameter

The comparison of the numbers of wins (contour figures) shows that only once VC method is better than SC method (Fig. 10) for all values of the parameter α but it is very often better on a large subset of the parameters (Fig. 12).

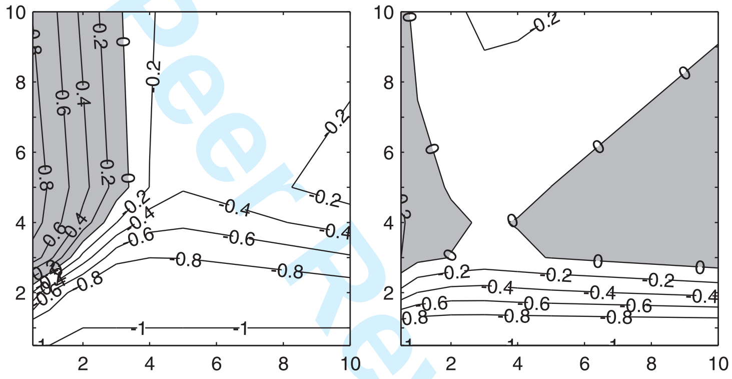

FIGURE 12. Difference between the percentage of VC and SC method wins for dataset *beetles* (left figure) and *iris* (right figure) (gray area means that VC is better than SC), x-axis: α parameter of SC, y-axis: α parameter of VC

As the number of elements of a dataset increases, the error of classification stabilizes — low variability, the variance decreases. VC method has higher level of variability, for small values of the parameter α, than SC method. As α increases, the variances of both methods decrease and tend to the level of variance of 1NN method. Often, not only error of classification but also the variances of SC and VC methods are lower than the variance of 1NN method (Fig. 13).

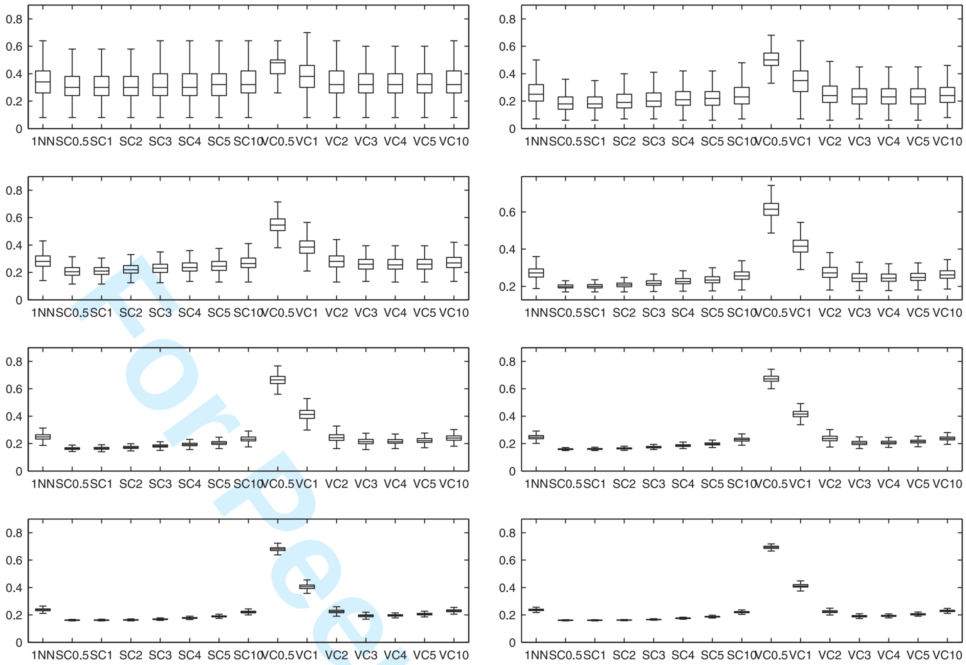

FIGURE 13. Boxplots of error rate for dataset 1 (in the order of training sample size), x-axis: methods, y-axis: error rate

Comparing all three classifiers by the numbers of wins, we see that SC and VC methods play a big part in classification (Figs. 14, 15). Fig. 14 shows participation of SC, VC, and 1NN methods for each real dataset. The numbers of wins are summed for all α parameters for each SC and VC methods. In Fig. 15, the numbers of wins for each α parameter for all real datasets are summed.

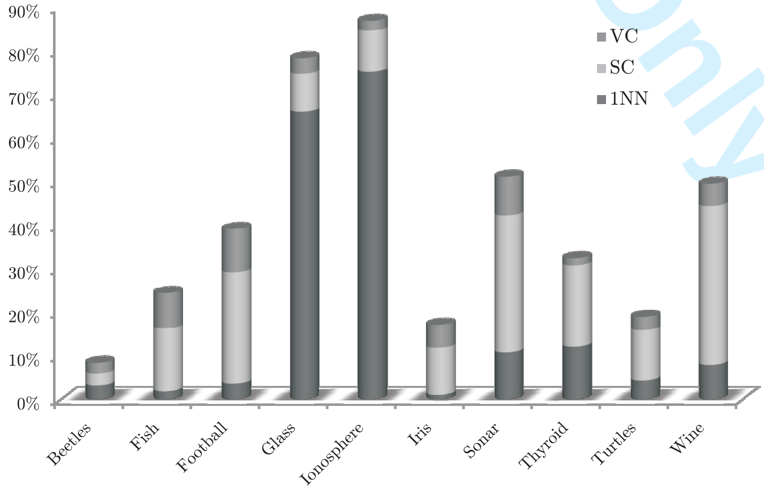

FIGURE 14. Cumulative percentage of wins for real datasets

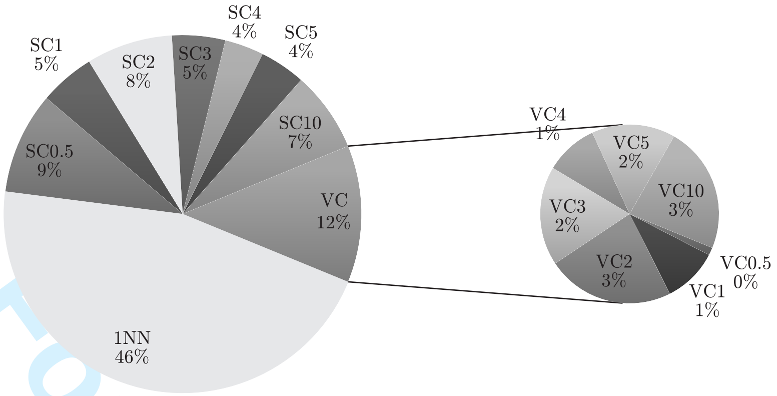

FIGURE 15. Percentage of wins for all real datasets

For example, SC and VC methods dominate (Fig. 14) on the dataset *fish* and *football*. We can see the significant participation of VC method on that dataset. Often, we cannot point which method is better, they tie. 1NN method just dominates on two real dataset: *glass* and *ionosphere* (Figs. 14).

SC and VC methods play a main part on artificial datasets as well. We can observe changing of classifiers participation as the number of elements in a dataset rises (Fig. 16). Especially, some SC classifiers are better on rather larger training datasets.

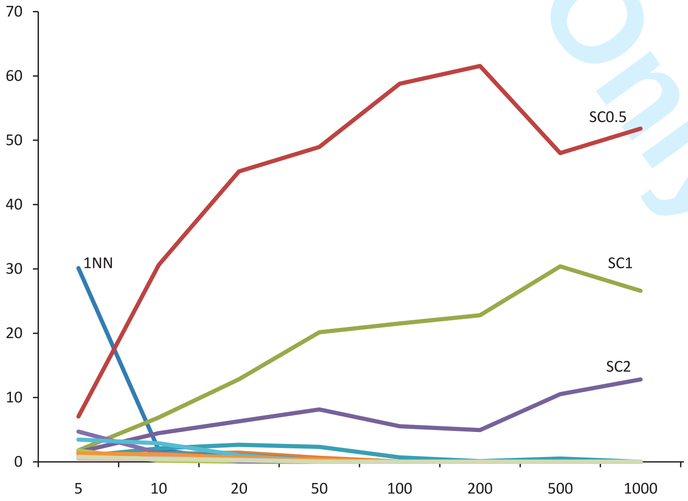

FIGURE 16. Percentage of wins for dataset 1, x-axis: training sample size, y-axis: percentage of wins

There is a significant participation of VC method on some datasets (Fig. 17). We can see that clearly on smaller training datasets.

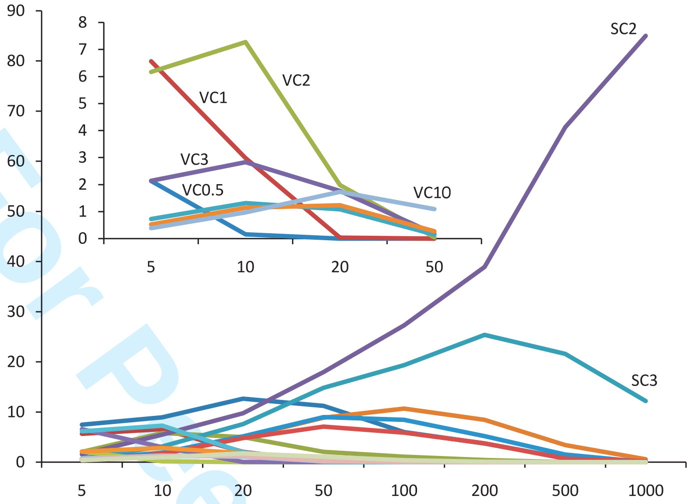

FIGURE 17. Percentage of wins for dataset 7, x-axis: training sample size, y-axis: percentage of wins

5. CONCLUSIONS

Our research showed that the introduced classifiers are much better than 1NN method on many datasets. For most datasets, the classification error of the classifiers (for some fixed value of the parameter α) is much lower than for 1NN method. The comparison of SC and VC methods showed that the first one is better on most datasets. However, VC method is sometimes as good as SC or even better, especially on very small datasets. It seems that these methods can replace 1NN classifier in many cases, since the implementation of them is not very difficult. As the single classification method SC seems to be a better choice. Since VC method is sometimes better than SC in some special cases, it may be successfully used as a component method in combining classification methods such as, for example, stacked regression (Breiman (1996)).

REFERENCES

[1] Breiman L., Friedman J.H., Olshen R.A., Stone C.J. (1984). *Classification and Regression Trees.* Boca Raton: Chapman & Hall.

[2] Breiman L. (1996), Stacked Regression. Machine Learning 24:49–64.

[3] Duda R., Hart P., Stork D. (2001), *Pattern Classification.* Wiley.

[4] Fukunaga K., (1990). *Introduction to Statistical Pattern Recognition.* San Diego: Academic Press.

[5] Gleim G. (1984). The Profiling of Professional Football Players. Clinical Sport Medicine 3(1):185–97.

[6] Hawkins A., Rasmussen K. (1978). The Calls of Gadoid Fish. Journal of the Marine Biology Association 58:881–911.

[7] Merz C.J., Murphy P.M. (1998). UCI repository of machine learning databases. Machine readable data repository http://www.ics.uci.edu/∼mlearn/MLRepository.html. Irvine, CA: University of California, Department of Information and Computer Science.

[8] Seber G.A.F. (1984). *Multivariate Observations.* New York: John Wiley & Sons.

[9] StatSoft Inc. (2001). STATISTICA (data analysis software system), version 6. www.statsoft.com.