pattern recognition, nearest neighbour rule,
reference set condensation, reference set reduction,
method of cutting hyperplanes

Artur SIERSZEŃ

REDUCTION OF REFERENCE SET WITH THE METHOD OF CUTTING HYPERPLANES

Reduction of this type may help to solve one of the greatest problems in pattern recognition, i.e. the compromise between the time of making a decision and its correctness. In the analysis of biomedical data, classification time is less important than certainty that classification is correct, i.e. that reliability of classification is accepted by the algorithm's operator. It is usually possible to reduce the number of wrong decisions, using a more complex recognition algorithm and, as a consequence, increasing classification time. However, with a large quantity of data, this time may be considerably reduced by condensation of a set. Condensation of a set presented in this article is incremental, i.e. formation of the condensed reference set begins from a set containing one element. In each step, the size of the set is increased with one object. This algorithm consists in dividing the feature space with hyperplanes determined with pairs of the mutually furthest points. The hyperplanes are orthogonal to segments linking pairs of the mutually furthest points and they go through their centre.

1. INTRODUCTION

Nowadays a tendency may be observed to describe studied problems (biomedical problems as well) in a very detailed way, using a large quantity of data. One of the most frequently used repositories is a data collection of University of California in Irvine (Machine Learning Repository, University of California, Irvine) [1]. During 1980s and 1990s sets contained 150 – 350 elements on average. Now this numbers has increased to app. 5000-7000 elements. However, the increase in computers' computation power did not compensate this fact; nor did it compensate the complexity level of computations used in data processing algorithms. It takes more time to obtain the end results in this circumstances (in case of biomedical data as well). This time may be shortened by the suitable reduction of the initial set. This article presents one of such methods.

The condensation rule of the reference set bases on the method of finding the mutually furthest points. This method consists in assigning one pair of the mutually furthest points (from different classes) to each point from the learning set on condition that in case of several furthest neighbours located at equal distances, the one with lowest number is always chosen. Because many objects may have the same pair of the mutually furthest points or a pair where a given point coincides with another, a pair with a lower number is chosen (i.e. a pair which was found more quickly). The problem of coinciding points belonging to the same classes and having the same properties was solved by omitting them (removing them from test sets). It did not adversely affect the error level of classification performed with the use of obtained condensed sets. A pair of the mutually furthest points is used to determine a hyperplane which divides the next subset. It goes through the centre of the segment connecting these points and it is orthogonal to it. The next subset to be divided is determined automatically by the algorithm. New subsets obtained through division are replaced with gravity centres; they are assigned to a specific class according to the size criterion, i.e. they are assigned to the largest class. The figure below (Fig. 1) illustrates the operation (the first two iterations) of the algorithm with the example of 2-dimensional feature space.

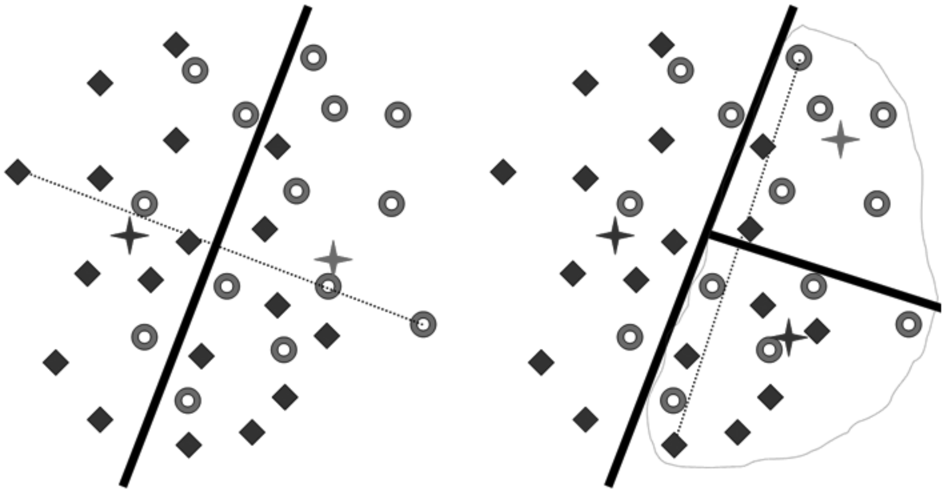

Fig. 1. The first two iterations of an exemplary operation of the presented algorithm (the first iteration – on the left, the second – on the right)

For the needs of the described condensation method, the modification of the algorithm of finding the mutually furthest points [2] was used. The operation of the algorithm is presented below with the use of a pseudocode.

T – the set of all testing objects, t – an element of T set;
$t = [a_1, a_2 \dots a_n]$; a – a property describing the point; n – a number of features

```
i00.  START
i01.  choose $t_k = t_0$ ($t_0$ = the first element of T set)
i02.  $t_z = t_k$
i03.  find $t_x$ element so that $|| t_k, t_x || = max$ if $t_x = t_z$ i05
i04.  $t_z = t_x$, $t_k = t_x$, go to i03.
i05.  END
```

The graphic interpretation of the algorithm of finding the mutually furthest points was presented in the figure (Fig. 2.). In each step of the algorithm, the distance to the furthest point from the other (another) class is determined (Fig. 2a. and Fig. 2b.). In case this distance is the same as the one computed in the previous cycle, the algorithm returns a pair of the mutually furthest points from different classes, which is essential for further computations (Fig. 2c.).

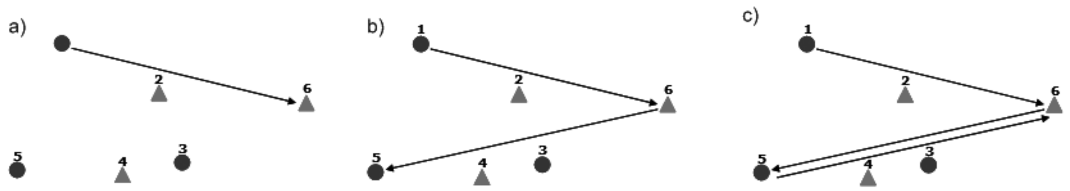

Fig. 2. The algorithm of finding the mutually furthest points (a – the first step, b – the second step, c – the third step)

2. REDUCTION OF REFERENCE SET WITH THE METHOD OF CUTTING HYPERPLANES

Below, a pseudocode is presented for the algorithm of condensing the reference set based on determining the cutting hyperplanes.

T – the learning set containing m objects; Z – the condensed set;
T^j – subsets of the learning set, j=1,2,...,i, after performing i^{th} iteration;
$T = \{t_1, t_2, \dots, t_m\}$; x, k – indexes of elements from the set T; m – number of elements in the primary reference set
$t = [a_1, a_2 \dots a_n]$; a – a feature describing an object; n – a number of features;

```
i00.  START; i=1; $T^i = T$; $Z=\Theta$ {i.e. null set}
i01.  Find a pair of the mutually furthest objects $t_j$ and $t_k$ in $T^i$ set
i02.  Construct a cutting hyperplane of $g(t)=0$ equation basing on $t_j$ and $t_k$ points
i03.  $T_{iA}= \{t \in T^i: g(t)>=0\}$; find a centre of gravity $z_{iA}$ of $T_{iA}$ set
i04.  $T_{iB}= \{t \in T^i: g(t)<0\}$; find a centre of gravity $z_{iB}$ of $T_{iB}$ set
i05.  Delete $T^i$, remember $T_{iA}$ as $T^i$ and $T_{iB}$ as $T^{i+1}$; next i=i+1
i06   Delete the gravity center of $T^i$
i07.  $Z= Z \cup \{z_{iA}, z_{iB}\}$
i08.  Estimate classification error for 1-NN rule working with the condensed set Z and remember it
i09.  Arrange $T^j$ sets, j=1,2,...i, so that $T^i$ is the largest set
i10.  If $T^i$ contains more than one objects, go to i01
i11.  END
```

Each time after a new reduced set has been determined (i.e. after each iteration), a classification error for 1-NN rule applied to the present condensed set is computed with the use of the leave-one-out method.

3. VERIFICATION OF THE ALGORITHM

The Cascades algorithm for the reduction of reference set was implemented in C++ in Microsoft Visual Studio .NET 2003 environment. This allowed the author to test the method in Windows environment with the use of a PC computer equipped with Intel Pentium processor 4 HT 3GHz and 512MB of operating memory.

The computation tests were conducted with the use of sets from the repository of the University of California in Irvine (Machine Learning Repository, University of California, Irvine) [1]. These tests are commonly used in literature. These are the following (tab. 1):

- PHONEME – data set created as a result of an analysis of separate syllables pronunciation (e.g. pa, ta, pan etc.); what was taken into account in this analysis was the type of a vowel pronunciation – nasal or oral;

- SATIMAGE – this data set was generated basing on the analysis of satellite pictures supported with other methods of observation (radar data, topographic maps, data concerning agriculture). Classes determine a kind of soil or a type of cultivation;
- WAVEFORM – artificially generated data set, where each of the classes is created as a result of a combining 2 out of 3 sinusoids; for each attribute in a class noise is generated.

Table 1. Parameters of the sets used during the tests

Name of the set	Number of classes	Number of features	Number of samples	Size of separate classes in the set					
				1	2	3	4	5	6
Phoneme	2	5	5404	3818	1586	-	-	-	-
Satimage	6	36	6435	1533	703	1358	626	707	1508
Waveform	3	21	5000	1657	1647	1696	-	-	-

All tests were repeated 25 times; the presented results (time of the algorithm's opera-tion) were calculated as the average during each iteration. Computing the error with the leave-one-out method was definitely most time-consuming. Therefore, the author decided that the error should be computed every second iteration. This permitted to accelerate computations significantly.

3.1. PHONEME TESTING SET

The chart presents the results of the algorithm's operation with the PHONEME testing set. It shows the relationship between the reference set's size and the classification error in the function of the number of iterations (Fig. 3.). The error level obtained for the 1-NN rule operating with the original reference set is presented (the horizontal line); the error was computed with the use of the leave-one-out method. Moreover, the total computation time is given (Fig. 4.) (in the aspect of the error level as well) in the function of an iteration. The computation time presented in the chart concerns the case of assessing classification quality (marked as Total computation time [s]) after each iteration and the case of its assessment after every second iteration (marked as Total computation time (2) [s]).

Additionally, exemplary iterations are shown with values of computation time and the error of classification performed with the use of the 1-NN method for the reference set obtained in the iteration in question (Table 2.).

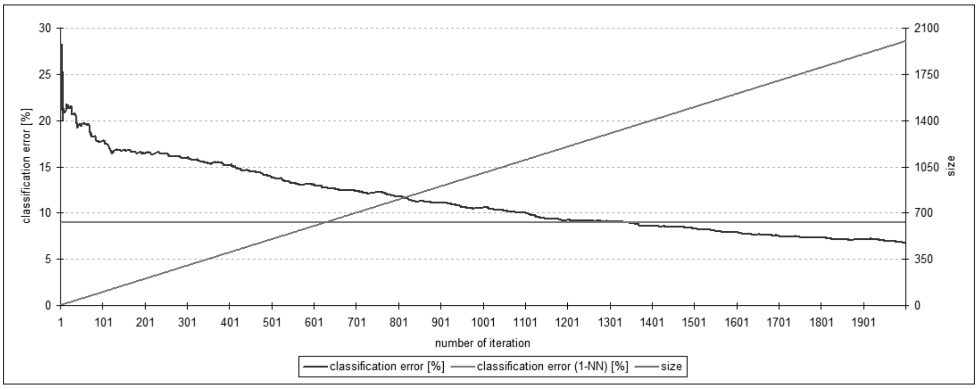

Fig. 3. The results of the algorithm obtained with the use of the PHONEME testing set
(relationship between the number of iterations and the size of the reference set or the classification error)

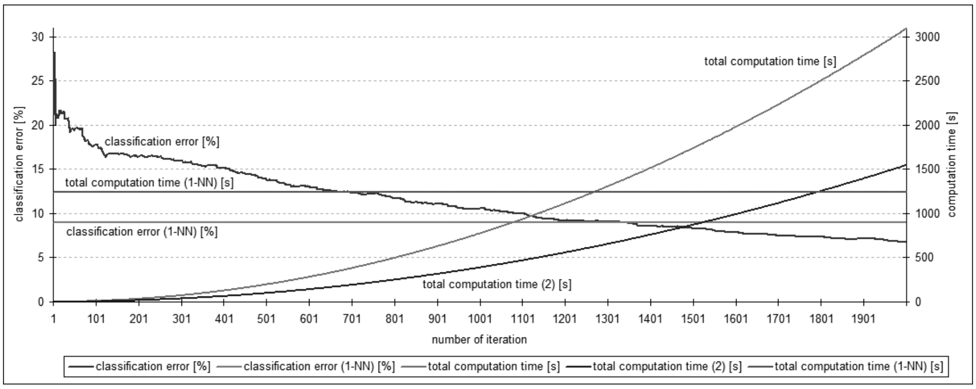

Fig. 4. The results of the algorithm obtained with the use of the PHONEME testing set
(the relationship between the number of iterations and the computation time or classification error)

Table 2. Time of obtaining the chosen results of the algorithm's operation with the use of the PHONEME testing set

	Time of computing classification error [ms]		Error [%]
the complete set	1237.8		8.97
	Time of computing classification error [ms]	Time of computing classification error (after every second iteration) [ms]	Error [%]
the reduced set after 1109 iterations	952.1	475.5	9.97
the reduced set after 1264 iterations	1236.6	618.5	9.11
the reduced set after 1342 iterations	1393.5	697.4	8.97
the reduced set after 1788 iterations	2476.2	1237.2	7.53

As a result of the tests, a better classification quality was obtained compared to the basic 1-NN method. Unfortunately, this result was achieved during a longer computation time. However, the application of an additional modification (computing the classification error every second iteration) enabled the author to obtain this result in the same time as with the use of the initial method. It should be emphasized that the author's algorithm obtained the result of the 1-NN method in twofold shorter time.

3.2. SATIMAGE TESTING SET

The chart presents the results of the algorithm's operation with the SATIMAGE testing set. It shows the relationship between the reference set's size and the classification error in the function of the number of iterations (Fig. 5.). The error level obtained for the 1-NN rule operating with the original reference set is presented (the horizontal line); the error was computed with the use of the leave-one-out method. Moreover, the total computation time is given (Fig. 6.) (in the aspect of the error level as well) in the function of an iteration. The computation time presented in the chart concerns the case of assessing classification quality (marked as Total computation time [s]) after each iteration and the case of its assessment after every second iteration (marked as Total computation time (2) [s]).

Additionally, exemplary iterations are shown with values of computation time and the error of classification performed with the use of the 1-NN method for the reference set obtained in the iteration in question (Table 3.).

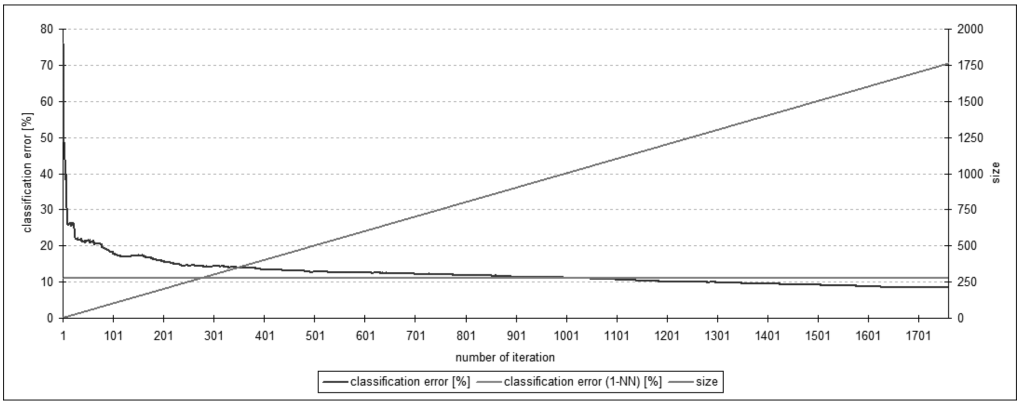

Fig. 5. The results of the algorithm obtained with the use of the SATIMAGE testing set
(relationship between the number of iterations and the size of the reference set or the classification error)

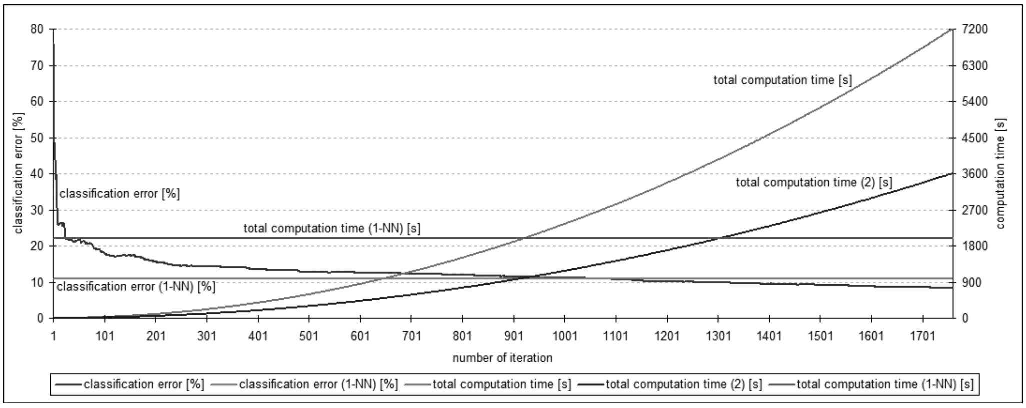

Fig. 6. The results of the algorithm obtained with the use of the SATIMAGE testing set
(the relationship between the number of iterations and the computation time or classification error)

Table 3. Time of obtaining the chosen results of the algorithm's operation with the use of the SATIMAGE testing set

	Time of computing classification error [ms]		Error [%]
the complete set	1982.3		10.96
	Time of computing classification error [ms]	Time of computing classification error (after every second iteration) [ms]	Error [%]
the reduced set after 919 iterations	1968.0	985.3	11.37
the reduced set after 1047 iterations	2568.6	1283.5	10.94
the reduced set after 1303 iterations	3966.5	1981.7	9.86
the reduced set after 1759 iterations	7211.4	3603.6	8.56

The performed tests revealed the possibility of obtaining better results, to be more specific a better classification quality compared to quality values obtained for the initial reference set. Similarly to the case of the PHONEME set, this result was obtained during the considerably longer computation time. Even the application of the additional modification (computing the error level every second iteration) did not enable the author to obtain these results in the same time as with the use of the initial method. However, it should be mentioned that the application of this modification allowed the author to obtain the result of the 1-NN method in 1/3 shorter time.

3.3. WAVEFORM TESTING SET

The chart presents the results of the algorithm's operation with the WAVEFORM testing set. It shows the relationship between the reference set's size and the classification error in the function of the number of iterations (Fig. 7.). The error level obtained for the 1-NN rule operating with the original reference set is presented (the horizontal line); the error was computed with the use of the leave-one-out method. Moreover, the total computation time is given (Fig. 8.) (in the aspect of the error level as well) in the function of an iteration. The computation time presented in the chart concerns the case of assessing classification quality (marked as Total computation time [s]) after each iteration and the case of its assessment after every second iteration (marked as Total computation time (2) [s]).

Additionally, exemplary iterations are shown with values of computation time and the error of classification performed with the use of the 1-NN method for the reference set obtained in the iteration in question (Table 4.).

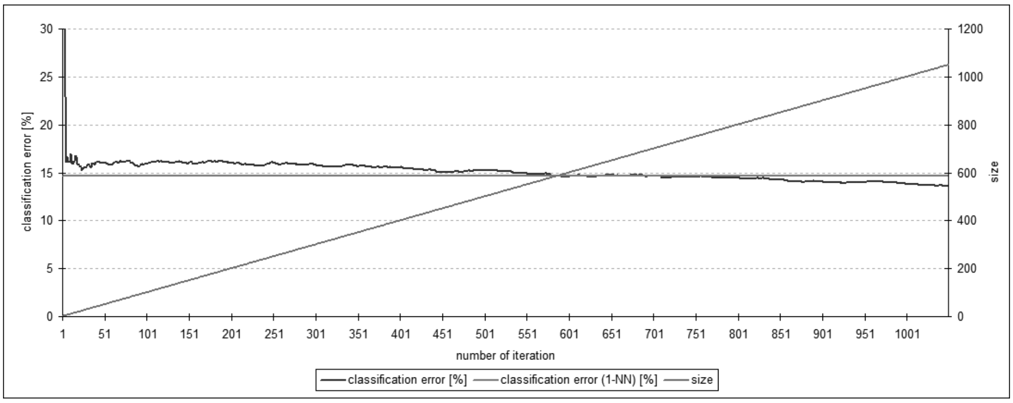

Fig. 7. The results of the algorithm obtained with the use of the WAVEFORM testing set
(relationship between the number of iterations and the size of the reference set or the classification error)

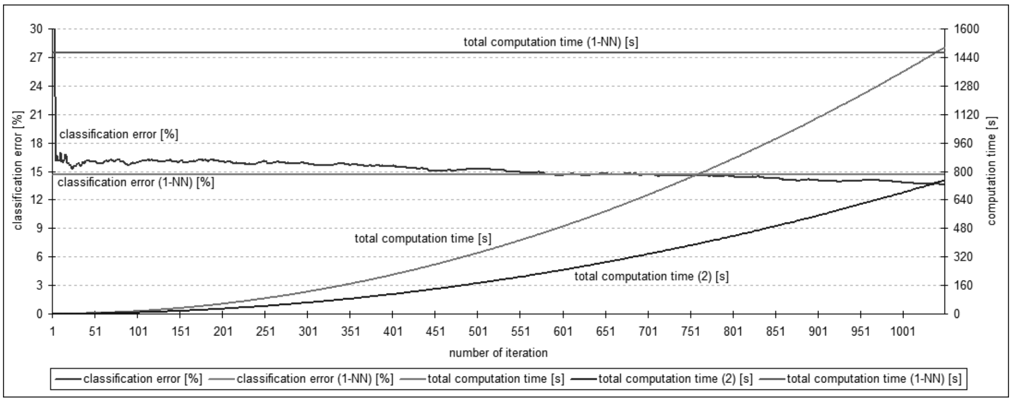

Fig. 8. The results of the algorithm obtained with the use of the WAVEFORM testing set
(the relationship between the number of iterations and the computation time or classification error)

Table 4. Time of obtaining the chosen results of the algorithm's operation with the use of the WAVEFORM testing set

	Time of computing classification error [ms]		Error [%]
the complete set	1467.2		14.67
	Time of computing classification error [ms]	Time of computing classification error (after every second iteration) [ms]	Error [%]
the reduced set after 90 iterations	11.9	6.0	15.64
the reduced set after 432 iterations	255.2	127.8	15.28
the reduced set after 580 iterations	458.5	229.6	14.68
the reduced set after 1051 iterations	1586.2	793.7	13.52

Very good results have been obtained for the WAVEFORM set. The author's condensation algorithm gave higher classification quality compared to the value obtained with the original reference set and, what is particularly important, the author's algorithm obtained the results in twofold shorter time in case of applying the additional modification (computing the error levels every second iteration).

4. CONCLUSIONS

The application of the author's algorithm, particularly with computing the error every second iteration, allowed to obtain a better result in a shorter time than the standard 1-NN method in case of medium sized sets (5000-6000 elements). Basing on computations performed for various sets, it was noticed that in each case the construction of the reference set enables one to find an optimum, in a particular moment, condensed set which may replace the original set; such condensed set gives lower classification error.

Medical data sets available to the author, e.g. PIMA set, which refers to recognizing symptoms of diabetes basing on criteria adopted by the World Health Organization, are too small (the number of samples amounts to 768) to be reduced; computation time is short anyway. However, if larger sets are created, the use of the presented method may be beneficial.

BIBLIOGRAPHY

[1] ASUNCION A., NEWMAN D.J., UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science, 2007.

[2] JÓŹWIK A., KIEŚ P.: Reference set size reduction for 1-NN rule based on finding mutually nearest and mutually furthest pairs of points. Computer Recognition Systems, Advances in Soft Computing, Vol. 30, pp.195-202 Springer Berlin/Heidelberg, 2005.