ELSEVIER

Pattern Recognition Letters 20 (1999) 1149–1156

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

Nearest neighbor classifier: Simultaneous editing and feature selection

Ludmila I. Kuncheva [a,*], Lakhmi C. Jain [b]

[a] School of Mathematics, University of Wales, Bangor, Bangor, Gwynedd, LL57 1UT, UK
[b] University of South Australia, Adelaide, Mawson Lakes, SA, 5095, Australia

Abstract

Nearest neighbor classifiers demand significant computational resources (time and memory). Editing of the reference set and feature selection are two different approaches to this problem. Here we encode the two approaches within the same genetic algorithm (GA) and simultaneously select features and reference cases. Two data sets were used: the SATIMAGE data and a generated data set. The GA was found to be an expedient solution compared to editing followed by feature selection, feature selection followed by editing, and the individual results from feature selection and editing. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Editing for the nearest neighbor classifier (1-nn); Feature selection; Genetic algorithms (GAs)

1. Introduction

The nearest neighbor classifier (1-nn) (Dasarathy, 1990; Duda and Hart, 1973) is intuitive and accurate. According to 1-nn, an input is assigned to the class of its nearest neighbor from a stored labeled reference set. The main problem using 1-nn is the significant time and memory resources that are required. With the developments of modern computational technology (faster hardware, higher memory capacity), this problem is likely to become less severe. On the other hand, however, larger data sets are being collected, stored and processed (e.g., in medical imaging) and the 1-nn computational demand is still a problem.

Let $X = \{X_1, \ldots, X_n\}$ be the set of features describing objects as n-dimensional vectors $\boldsymbol{x} = [x_1, \ldots, x_n]^{\mathrm{T}}$ in $\mathbb{R}^n$ and let $Z = \{\boldsymbol{z}_1 \ldots, \boldsymbol{z}_N\}$, $\boldsymbol{z}_j \in \mathbb{R}^n$, be the data set. Associated with each $\boldsymbol{z}_j$, $j = 1, \ldots, N$, is a class label from the set $C = \{1, \ldots, c\}$.

Two ways of reducing the operational time of 1-nn are to structure the reference set properly or to use fast search methods. Here we focus on two other ways: *editing* and *feature selection*, and their simultaneous application. Instead of Z we use a subset $S_1 \subseteq Z$, and instead of X we use $S_2 \subseteq X$. Fig. 1 shows the reduction of Z, both row-wise and column-wise.

To find a reduced set we use a *genetic algorithm* (GA). Section 2 outlines editing and feature selection, Section 3 describes our GA, Section 4 contains the experimental results and Section 5 offers some conclusions.

* Corresponding author.
E-mail addresses: l.i.kuncheva@bangor.ac.uk (L.I. Kuncheva), Lakhmi.Jain@unisa.edu.au (LC. Jain)

0167-8655/99/$ - see front matter © 1999 Elsevier Science B.V. All rights reserved.
PII: S0167-8655(99)00082-3

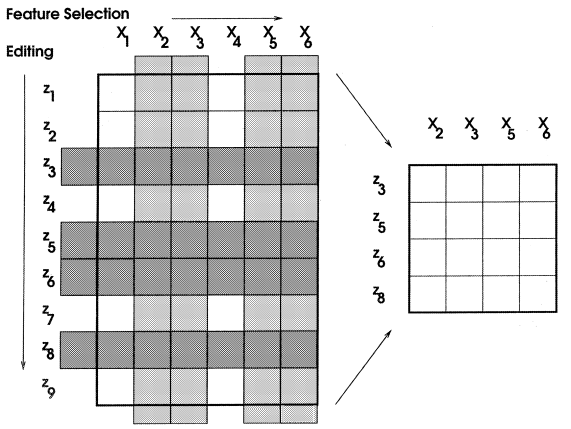

Fig. 1. Editing and feature selection for the 1-nn classifier.

2. Feature selection and editing

In this study we use two basic methods for editing: *Hart's condensed nearest neighbor rule* (Hart, 1968) and *Wilson's method* (Wilson, 1972). For feature selection we use the *Sequential Forward Selection* (SFS) method (Stearns, 1976). The three algorithms are outlined in Figs. 2–4. Hart's algorithm guarantees zero resubstitution errors on Z using S_1 as the reference set (S_1 is called a *consistent* subset of Z). Hart's method tends to retain objects along the classification boundaries. The resultant set S_1 depends on the ordering of the elements of Z. In contrast to this, Wilson's method rules out from Z objects which are misclassified (presumably the boundary objects!), retaining those which are likely to belong to their own Bayes-optimal classification region. The selected subset does not depend on the order of the elements in Z. None of these two methods has any restriction on the cardinality of the resulting set S_1.

Because of the peaking effect in feature selection, it is possible to find a subset $S_2 \subset X$ such that the classification accuracy of 1-nn ($P_{1\text{-nn}}$) using only S_2, is higher than that using all of X. SFS is a simple and yet effective algorithm, although not guaranteeing optimality of the solution, nor even close sub-optimality. Ties in the classification accuracy are possible to occur because $P_{1\text{-nn}}$ is calculated as the "apparent error rate" (i.e., the proportion of correctly classified elements of Z) and is, therefore, discrete.

Although both data editing and feature selection aim at data reduction while trying to keep the classification accuracy as high as possible, their semantics and therefore search strategies are different. For example, SFS starts with the best *single feature* and adds one feature at a time. This strategy is not applicable for data editing: First, there cannot be any "best data point" to start with. Second, feature selection searches through a

Hart's condensing algorithm

1. Set GRABBAG=$Z - \{\mathbf{z}_1\}$, STORE=$\{\mathbf{z}_1\}$.
2. Classify the elements of GRABBAG using 1-nn and STORE as the reference set. If a misclassification occurs, move the misclassified element of GRABBAG in STORE and continue.
3. Repeat previous step until one of the two stop conditions is met
   (a) GRABBAG is empty.
   (b) No misclassification occurs when classifying the whole of GRABBAG.
4. Return S_1 = STORE.

Fig. 2. Hart's condensing algorithm.

Wilson's editing algorithm

1. Run k-nn with leave-one-out on Z and mark for deletion all misclassified object. (The value for k recommended by Wilson in [15] is 3)
2. Delete the marked objects and return the remaining set as S_1.

Fig. 3. Wilson's editing algorithm.

SFS algorithm

1. Set $Y = \emptyset$.
2. For $i = 1 : n$, (Number of features)
   (a) For any $X_j \in X - Y$ calculate the leave-one-out 1-nn classification accuracy of Z using $Y \cup \{X_j\}$ as the feature set.
   (b) Assign as the new Y the set $Y \cup X_k$ with the highest accuracy among the tested feature subsets at step i.
   (c) Store the result found at step i: the feature subset and the value of the P_{1-nn}.
3. End i.
4. Return as S_2 the feature subset with the highest accuracy, and if a tie occurs, choose the set with smaller cardinality.

Fig. 4. Sequential Forward (feature) Selection algorithm.

smaller set (2^n, the cardinality of the power set of X) than editing (2^N, the cardinality of the power set of Z; usually $N > n$). Therefore, methods for feature selection, like the groups of sequential forward and backward selection (SFS, SBS) (Stearns, 1976) and the extentions thereof (Pudil et al., 1994) are not applicable for editing, nor are editing search strategies applicable for feature selection.

The opportunity to combine the two approaches lies in the fact that they use the same criterion function: the classification accuracy. Let $\mathscr{P}(Z)$ be the power set of Z (the data set) and $\mathscr{P}(X)$ be the power set of X (the feature set). We consider the Cartesian product of the two power sets and define $P_{1\text{-nn}}$ as a real-valued function

$$P_{1\text{-nn}} : \mathscr{P}(Z) \times \mathscr{P}(X) :\rightarrow [0, 1].$$

Let $S_1 \subseteq Z$ and $S_2 \subseteq X$. Then $P_{1\text{-nn}}(S_1, S_2)$ is the classification accuracy of the nearest neighbor classifier using S_1 as the reference set and S_2 as the feature set. The problem now is *how* to search in the combined space.

3. Genetic algorithms for simultaneous editing and feature selection

There are controversies about applying Genetic Algorithms (GAs) for feature selection. Some authors find them very useful (Chtioui et al., 1998; Leardi, 1994, 1996; Sahiner et al., 1996; Siedlecki

and Sklansky, 1989), while others are skeptical (Pudil et al., 1994) and warn that the results are often not as good as expected, compared with other (simpler!) feature selection algorithms (Jain and Zongker, 1997).

Using GAs for data editing is suggested in (Kuncheva, 1995, 1997). Although GAs do not guarantee optimality of the solution, it was found that they are an expedient editing technique (Kuncheva and Bezdek, 1998).

In this study we propose to use a genetic algorithm for simultaneous editing and feature selection. Three advantages of this idea are:

- The encoding of the problem is straightforward.
- Unlike many feature selection algorithms, GAs do not assume monotonicity of the criterion (fitness) function.
- Unlike many editing techniques, GAs have a number of tuning parameters.

The usual problem with GAs is the long time to get to a good solution. Notice that, in terms of real application of the 1-nn classifier, the GA is applied in the *training* phase (off-line), and the *operational time* of 1-nn depends on the GA result.

The encoding of the problem is straightforward. We believe that GAs are most suitable for binary search problems, such as editing and feature selection. The search space in our "joint" problem consists of $2^{(N+n)}$ elements. Each chromosome S is a binary string consisting of $N+n$ bits and representing 2 sets: $S_1 \subset Z$ and $S_2 \subset X$. The first N bits are used for S_1, and the last n bits for S_2. The ith bit has value 1 when the respective element of Z/X is included in S_1/S_2, and 0 otherwise. For example, the chromosome corresponding to the reduced set Z in Fig. 1 (data points z_3, z_5, z_6, z_8 and features X_2, X_3, X_5, X_6) is

$$S = [0\,0\,1\,0\,1\,1\,0\,1\,0\,0\,1\,1\,0\,1\,1].$$

As the fitness function we use $P_{\text{1-nn}}$ and a penalty term as a soft constraint on the total cardinality of S_1 and S_2.

$$F(S) = \hat{P}_{\text{1-nn}}(S) - \alpha \frac{(\text{card}(S_1) + \text{card}(S_2))}{(N+n)},$$

where α is a coefficient. The classification accuracy is measured on a validation set, different from the training set Z. We assume that the reader is familiar with the basics of GAs, so only the specific features are listed below:

- population size = 10;
- initialization probability = 0.8 (the number of 1's in the initial population is around 80% of all bit values generated);
- terminal number of generations = 100;
- the whole population is taken as the mating set;
- 5 couples are selected at random (repetitions are permitted) to produce 10 offspring chromosomes (probability of crossover = 1.0);
- probability of mutation = 0.1;
- selection strategy: elitist, i.e., the current population and the 10 offsprings are pooled and the "fittest" 10 survive as the next population.

4. Experiments

We used 2 data sets:

1. The *SATIMAGE data* from ELENA database (anonymous ftp at ftp.dice.ucl.ac.be, directory pub/neural-nets/ELENA/databases): 36 features, 6 classes, 6435 data points with 3 different training–validation–test splits of the same size: 100/200/6135.
2. A *generated data* set (see Jain and Zongker, 1997): 20 features, 2 classes, 10 different samplings with training–validation–test sizes as 100/200/1000. The classes were equiprobable, distributed as $p_1(\boldsymbol{x}) \sim N(\mu_1, I)$ and $p_2(\boldsymbol{x}) \sim N(\mu_2, I)$, where

$$\mu_1 = -\mu_2 = \left[\frac{1}{\sqrt{1}}, \frac{1}{\sqrt{2}}, \ldots, \frac{1}{\sqrt{20}}\right]^{\mathrm{T}}.$$

The experiments carried out are displayed in Table 1. The editing and feature selection techniques (the 4th row in the table) were applied in the given order. For example, SFS + W means that we first apply the SFS procedure for feature selection, and then Wilson's editing method with the so reduced feature set. The results expressed as error rates on the testing set and the number of parameters retained ($\text{card}(S_1), \text{card}(S_2)$, and their product), averaged over 3 experiments with

Table 1
1-nn experiments with the two data sets

Original set	No editing, no feature selection
Editing	Hart's condensing method (H) (Hart, 1968) Wilson's editing method (W) (Wilson, 1972) W followed by H (W + H)
Feature selection	Sequential forward selection (SFS)
Editing and feature selection (separate)	H + SFS W + SFS W + H + SFS SFS + H SFS + W SFS + W + H
GA	**Simultaneous editing and feature selection**

SATIMAGE and 10 experiments with the generated data, are presented in Tables 2 and 3.

Figs. 5 and 6 show the scatterplot of the methods with respect to: the logarithm of the number of parameters (the fewer, the better) and the testing classification error ($1 - P_{1\text{-nn}}$, the smaller, the better). The number of parameters of the reduced set is calculated as $\text{card}(S_1) \times \text{card}(S_2)$. Points that are close to the origin of the coordinate system are more desirable than those at a greater distance. Considering the two criteria: high $P_{1\text{-nn}}$ and low $\text{card}(S_1) \times \text{card}(S_2)$, a Pareto-optimal set of results can be defined for each of the two data sets. Included in a Pareto-optimal set are the *non-dominated* methods. A method is called non-dominated if there is no other method from the original set which is better on both criteria, or equivalent on the one and better on the other criterion. Methods in the Pareto-optimal sets are marked with "P" in the last columns of Tables 2 and 3 and shown with a thick line in Figs. 5 and 6. For the SATIMAGE data, the Pareto-optimal set is $\{(\text{W} + \text{H}) + \text{SFS},\ \mathbf{GA},\ \text{W} + \text{SFS},\ \text{SFS}\}$ and for the generated data, $\{(\text{W} + \text{H}) + \text{SFS},\ \mathbf{GA},\ \text{SFS} + \text{W}\}$.

5. Discussion and conclusions

Hart's method was originally designed for finding a consistent subset, not taking generalization into consideration. It is not surprising then, that the method on its own, and various combinations thereupon did not show high testing accuracy in our experiments. The combined editing (W + H) followed by feature selection provides a compact data set but the classification

Table 2
Averaged results with the SATIMAGE data (3 experiments)

Method	Error (%)	$\text{card}(S_1)$	$\text{card}(S_2)$	Total # of parameters	Pareto-optimality
All	17.87	100	36	3600	
H	21.59	37.33	36	1344	
W	18.97	78.33	36	2820	
W + H	21.23	12	36	432	
SFS	17.54	100	14.67	1467	P
H + SFS	20.65	38	14.33	545	
W + SFS	17.68	78.33	14.67	1149	P
W + H + SFS	18.83	12	11	132	P
SFS + H	21.06	34.33	14	481	
SFS + W	18.60	80.33	14.67	1178	
SFS + W + H	19.84	12.67	14.67	186	
GA	**18.09**	**27**	**10.33**	**279**	**P**

Table 3
Averaged results with the generated data (10 experiments)

Method	Error (%)	card(S_1)	card(S_2)	Total # of parameters	Pareto-optimality
All	11.94	100	20	2000	
H	14.91	28.7	20	574	
W	9.27	93.4	20	1868	
W + H	12.62	20	20	400	
SFS	9.59	100	13.9	1390	
H + SFS	12.73	28.7	9.5	273	
W + SFS	10.21	93.4	10.6	990	
W + H + SFS	10.95	20	9.7	194	P
SFS + H	14.96	26.7	13.9	371	
SFS + W	8.41	91.1	13.9	1266	P
SFS + W + H	11.94	14.6	13.9	203	P
GA	**9.17**	**26.28**	**8.64**	**227**	**P**

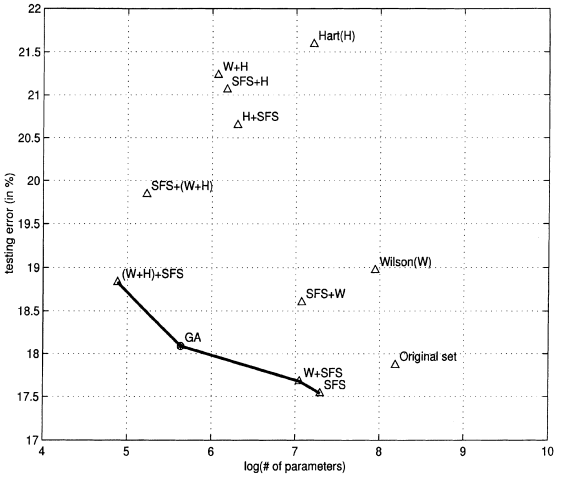

Fig. 5. Scatterplot of the methods for the SATIMAGE data.

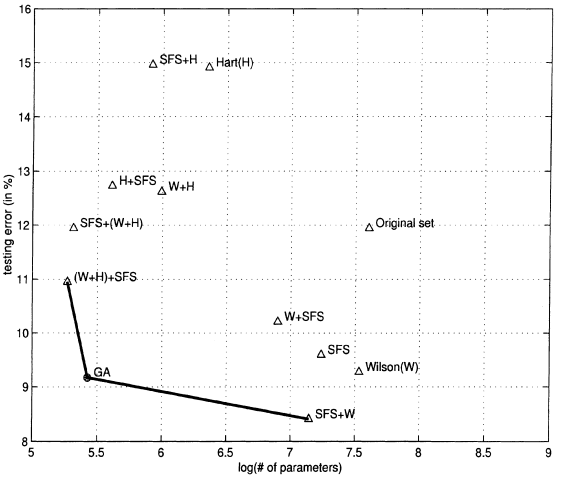

Fig. 6. Scatterplot of the methods for the generated data.

accuracy is not very high compared to other methods in the experiment. Different combinations of SFS and Wilson's method appeared in the two Pareto-optimal sets. They have a comparatively high accuracy but also a large number of parameters. GAs turn out to be a good compromise with both high accuracy and a moderate number of parameters and can be favored as the best choice.

Discussion

Szirányi: Observing that the population size of 10 is rather low and the mutation probability is high, to me it looks like a random search rather than a genetic algorithm.

Kuncheva: True, this is a genetic algorithm that deviates only slightly from random search. In fact, random search would be my favorite strategy.

I could make the genetic algorithm more focused, using a larger population, roulette wheel selection and a smaller mutation rate, but I decided to try this very basic and fast thing first to see whether there is any rabbit behind the bush.

Egmont-Petersen: First of all, the experiments by Sklansky, published in Pattern Recognition Letters in 1989 (*Note of the editors*: *see* (*Siedlecki and Sklansky*, *1989*) *in this paper*) also confirm that the genetic algorithm they used, leads to only very minor improvements in the error rate. I just want to confirm that. I personally have good experience with backward search. The reason is that backward search takes all dependencies into account when you start with the whole feature set. So, if you have a very good classification technique, you include all dependencies, whereas if you do a sequential forward search, you do not take all these mutual dependencies optimally into account, I think. Could you comment on that?

Kuncheva: I think that Siedlecki and Sklansky used a slightly different criterion function. I did not notice that they were not very happy with their results. The second comment about the backward search: using the nearest neighbour classifier both in forward selection and in backward selection, the criterion is the accuracy of the nearest neighbour. I found that it does not make much difference which one is used. With other types of classifier one of the two, forward or backward selection, may be better than the other.

Egmont-Petersen: It depends on how the dependencies are taken into account. There is another question: in your criterion function, you have the same weight for all features. But sometimes, in practical classification problems, some features are much more difficult to compute, much more expensive than others. Can you always assume that you can weigh them in this way?

Kuncheva: One may think of variations of the criterion function. Of course you can use more parameters (individual weights for the different features), but you have to make sure that you choose proper values for these parameters in the criterion function.

Pudil: I just want to make a little comment: when you talked at the beginning about various search strategies, you mentioned that the floating search strategy is the most efficient one and you referred to Jain and Zongker. This might lead to the misleading conclusion that they developed it, but that is not the case. I developed it and published the algorithm; they just did the experimental comparison study.

Kuncheva: Yes, I apologise for this. I know the algorithm is due to you (*Note of the editors*: *see* (*Jain and Zongker*, *1997*; *Pudil et al.*, *1994*) *in the paper*).

References

Chtioui, Y., Bertrand, D., Barba, D., 1998. Feature selection by a genetic algorithm. Application to seed discrimination by artificial vision. J. Sci. Food Agric. 76, 77–86.

Dasarathy, B.V., 1990. Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos, CA.

Duda, R.O., Hart, P.E., 1973. Pattern Classification and Scene Analysis. Wiley, New York.

Hart, P.E., 1968. The condensed nearest neighbor rule. IEEE Transactions on Information Theory 16, 515–516.

Jain, A., Zongker, D., 1997. Feature selection: evaluation, application and small sample performance. IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (2), 153–158.

Kuncheva, L.I., 1995. Editing for the *k*-nearest neighbors rule by a genetic algorithm. Pattern Recognition Letters 16, 809–814.

Kuncheva, L.I., 1997. Fitness functions in editing *k*-nn reference set by genetic algorithms. Pattern Recognition 30, 1041–1049.

Kuncheva, L.I., Bezdek, J.C., 1998. On prototype selection: Genetic algorithms or random search?. IEEE Transactions on Systems, Man, and Cybernetics C28 (1), 160–164.

Leardi, R., 1994. Application of a genetic algorithm to feature selection under full validation conditions and to outlier detection. Journal of Chemometrics 8, 65–79.

Leardi, R., 1996. Genetic algorithms in feature selection. In: Devillers, J. (Ed.), Genetic Algorithms in Molecular Modeling. Academic Press, London, pp. 67–86.

Pudil, P., Novovičová, J., Kittler, J., 1994. Floating search methods in feature selection. Pattern Recognition Letters 15, 1119–1125.

Sahiner, B., Chan, H.-P., Wei, D., Petrick, N., Helvie, M.A., 1996. Image feature selection by a genetic algorithm: application to classification of mass and normal breast tissue. Medical Physics 23 (10), 1671–1684.

Siedlecki, W., Sklansky, J., 1989. A note on genetic algorithms for large-scale feature selection. Pattern Recognition Letters 10, 335–347.

Stearns, S., 1976. On selecting features for pattern classifiers. In: 3-d International Conference on Pattern Recognition, Coronado, CA, pp. 71–75.

Wilson, D.L., 1972. Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics SMC-2, 408–421.