# Using a genetic algorithm for editing k-nearest neighbor classifiers

R. Gil-Pita[1] and X. Yao[23] *

[1] Teoría de la Señal y Comunicaciones, Universidad de Alcalá, Madrid (SPAIN)
[2] Computer Sciences Department, University of Birmingham, Birmingham (UK)
[3] Nature Inspired Computation and Applications Laboratory, University of Science and Technology of China, Hefei, Anhui 230027 (P. R. China.)

**Abstract.** The edited $k$-nearest neighbor consists of the application of the $k$-nearest neighbor classifier with an edited training set, in order to reduce the classification error rate. This edited training set is a subset of the complete training set in which some of the training patterns are excluded. In recent works, genetic algorithms have been successfully applied to generate edited sets. In this paper we propose three improvements of the edited $k$-nearest neighbor design using genetic algorithms: the use of a mean square error based objective function, the implementation of a clustered crossover, and a fast smart mutation scheme. Results achieved using the breast cancer database and the diabetes database from the UCI machine learning benchmark repository demonstrate the improvement achieved by the joint use of these three proposals.

## 1 Introduction

Editing a $k$-nearest neighbor (kNN) consists of the application of the kNN classifier with an edited training set in order to improve the performance of the classifier in terms of error rate [1]. This edited training set is a subset of the complete training set in which some of the training patterns are excluded. So, depending on the characteristics of the database [2], and due to the exclusion of these patterns, the kNN may render better results using the edited set, in terms of both error rate and computational cost.

Genetic algorithms (GA) have been successfully applied to select the training patterns included in the edited training set. In [3] a study of editing kNN classifiers using GAs with different objective functions is presented. Several databases like the Iris database or the Heart database are used in the experiments. The paper concludes that, from the analyzed objective functions, the best results are obtained when the counting estimator with penalizing term is selected as objective function. Other interesting article is [4], in which a GA with a novel crossover method is applied. When two parents are crossed, a high number of

* This work has been partially funded by the Comunidad de Madrid/Universidad de Alcalá (CCG06-UAH/TIC-0378) and the Spanish Ministry of Education and Science (TEC2006-13883-C04-04/TCM)

possible offsprings are evaluated, and the best two individuals are selected. The work presented in [5] is other interesting paper that studies the kNN edited with other heuristic techniques, in which the authors study the use of tabu search to solve the problem of editing a 1NN classifier (nearest neighbor rule). They use the counting estimator objective function with a penalizing term, and they evaluate the results with the iris database and a synthetic two-dimensional database. At last, the use of a multi-objective evolutionary algorithm to simultaneously edit and select the features of an 1NN classifier is evaluated in [6].

In this paper we propose a novel application of GAs for editing kNN classifiers. We describe the design of a genetic algorithm in order to edit training sets for kNN classifiers, focusing on the selection of the objective function to be minimized, and on the different parameters of the genetic algorithm like, for example, the crossover and the mutation techniques. Three improvements are proposed to the editing process using genetic algorithms: the use of a mean square error (MSE) based objective function, the implementation of a clustered crossover, and a fast smart mutation scheme. Results are compared using the breast cancer database and the diabetes database from the UCI machine learning benchmark repository. The computational cost after training and the classification error rate are considered in the study.

## 2 Materials and methods

In this section we carry out a brief description of the main classification method this paper deals with: the kNN. After describing the statistical basis of the kNN method, we mathematically describe the editing process of a kNN method, and how the genetic algorithms can be used for editing training sets.

### 2.1 kNN statistical analysis

The kNN classifier is statistically inspired in the estimation of the posterior probability $p(H_i|\mathbf{x})$ of the hypothesis $H_i$, conditioned to the observation point $\mathbf{x}$. Considering a volume around the observation point that encompasses $k$ patterns of the training set and $k[i]$ patters belonging to hypothesis $H_i$, then equation (1) is an approach of the posterior probability of the class $H_i$ [7].

$$p(H_i|\mathbf{x}) = \frac{p(\mathbf{x}|H_i)p(H_i)}{p(\mathbf{x})} \simeq \frac{k[i]}{k} \tag{1}$$

The Maximum A Posteriori criterion establishes that, for a given observation $\mathbf{x}$, the decision that maximizes the associated posterior probability must be taken. The kNN method fixes $k$, the number of patterns included in the volume, being these patterns the $k$ nearest (less distanced) patterns from the observation point. The decision is taken by evaluating the values of $k[i]$, $i = 1, \ldots, C$, and selecting the class which obtains a highest $k[i]$ value, and, therefore, maximizes approximation the posterior probability $p(H_i|\mathbf{x})$ given by equation (1). Concerning the distance measurement, in this paper we use the Euclidean distance, so that the volumes are hyper-spheres around the observation point.

### 2.2 Editing a training set with a GA

The edited training set is defined by the indexes of the patterns included in the subset. In order to apply genetic optimization, each subset is associated to a binary vector $\mathbf{b}$, with $N$ bits, being $N$ the total number of patterns in the original training set, so that if the $n$-th bit is activated $b[n] = 1$, then the corresponding $n$-th training pattern is included in the subset. So, being $S$ the number of patterns included in the reduced set so that $S \leq N$, then $\sum_n b[n] = S$. The bit-stream $\mathbf{b}$ is determined by the minimization of a given objective function. In this task, different algorithms can be applied, and we focus on the application of GAs to obtain the value of $\mathbf{b}$.

Concerning the objective function, its selection is an important issue in the design of edited kNN methods. Most of the papers use the classification error as objective function (equation (2)), adding in some cases a penalizing term, to consider the number of patterns in the edited training set.

$$F = \frac{1}{N} \sum_{n=1}^{N} h(\mathbf{x}_n) + \alpha b[n] \tag{2}$$

where $h(\mathbf{x}_n)$ is 1 if the $n$-th pattern is wrongly classified, and 0 in other cases. This objective function is equivalent to the *counting estimator with penalizing term* (CEPT) function, described in [3].

In this paper GAs are applied to determine the optimal bit-stream $\mathbf{b}$, in order to implement an edited kNN rule. So, the process to determine an edited subset for the kNN method using a GA is described as follows:

- The objective function is evaluated with the validation set for several values of $k$, and the value that performs better results is selected.
- Using a GA, a subset is selected using the training patterns, trying to minimize the objective function measured with the training set. The genetic algorithm is based on the use of natural selection, local mutations and crossover. The population is composed of $P = 100$ individuals, and the best 10 are selected in each generation. The remaining 90 individuals are then generated by single point crossover of the 10 survivors. Then, binary mutations are applied to the whole population, changing a bit with a probability of 0.8%. This process is iterated $G = 100$ generations.
- During the training process, the objective function over the validation set is calculated for the best individual of each population. The subset that achieves the lowest objective value over the validation set is selected as the final subset.

## 3 Description of the proposals

In this section descriptions of the three proposals of this paper are included. The first proposal deals with the definition of a novel objective function, inspired in the mean square error (MSE) function, used in many adaptive systems, like, for

example, neural networks. The second proposal defines a novel crossover strategy for the GA, designed using a priori knowledge of the problem. The third and last proposal establishes a new mutation scheme for the GA, that allows to lead the mutations in the population, improving the performance of the GA in the search of local minima of the objective function.

### 3.1 Editing kNN with a MSE-based objective function

In this paper we propose the use of a novel objective function, based on the MSE function. Let's consider the kNN as a system with $C$ outputs, so that each output is calculated using equation (1). Therefore, the $C$ outputs of the kNN system are approximations of the posterior probabilities of the data. So, $y_n[i]$, described in equation (3), is the $i$-th output of the kNN system, obtained for the $n$-th training pattern $\mathbf{x}_n$.

$$y_n[i] = \frac{k_n[i]}{k} \tag{3}$$

where $k_n[i]$ is the number of nearest patterns of class $i$ for the $n$-th training pattern. Considering the approximation described in equation (1), the outputs of this kNN system are estimations of the posterior probabilities of the classes. So, the objective function to minimize is designed using all the outputs of the system, by minimizing the mean square error, defined by equation (4).

$$F = \frac{1}{NC} \sum_{n=1}^{N} \sum_{i=1}^{C} \left( y_n[i] - d_n[i] \right)^2 \tag{4}$$

where $y_n[i]$ is the $i$-th output of the system for the $n$-th training pattern, and $d_n[i]$ is the desired $i$-th output for the $n$-th training pattern, so that $d_n[i]$ is 1 for the patterns of the $i$-th class and 0 for the rest. In function of the Kronecker delta, $d_n[i] = \delta[i - c_n]$, being $c_n$ the index of the class for the $n$-th training pattern. Replacing (3) in (4), we obtain (5).

$$F = \frac{1}{NC} \sum_{n=1}^{N} \sum_{i=1}^{C} \left( \frac{k_n[i]}{k} - \delta[i - c_n] \right)^2 \tag{5}$$

The error surface defined by this function is smoother than those obtained using counting estimator based functions, making easier the obtaining of its local minima.

### 3.2 Editing kNN using a GA with clustered crossover

Single point crossover (SPC) and random crossover (RC) are two of the most commonly used crossover methods used in the literature. Single point crossover generates the offspring by combining the first part of the bit stream of one parent with the last part of the bit stream of the other parent. On the other hand, random crossover generates the offsprings randomly selecting each bit-gene

from one of the parents. These two schemes do not consider possible relationship between the genes.

In this paper we propose a novel crossover scheme for the GA, denominated clustered crossover (CC), in order to improve the determination of the best subset of the training set, that minimizes the selected objective function. In the problem of the selection of the edited training set, it is possible to determine the relationship between the different bits of the bit stream. Each gene is related to the inclusion in the reduced subset of a training pattern. So, the value of a given gene is related to the performance of the classifier in the region of the space around the associated training pattern. Therefore, genes can be grouped into clusters considering the spatial position of their training patterns, using a non supervised clustering technique. In this paper we apply the k-means clustering algorithm [8] to the training data, so that these patterns are grouped in some sets of near patterns (clusters). Each group of patterns defines a cluster of genes, that is considered as a transfer unity in the crossover process. So, the bit-stream of the offsprings are obtained by mixing the clusters of genes of two parents. The clustering process is carried out every generation, and the number of clusters has been selected at random. So, every generation the crossover is carried out with different gene clusters.

### 3.3 Editing kNN using a GA with a fast smart mutation scheme

In this section we describe the application of a mutation scheme that allows to select the best gene or group of genes to be changed, taking into account the variations of the objective function with respect to each gene for a given edited set. We design a fast method for evaluating the error variation when each gene is changed, and we propose a mutation strategy based on these variations of the objective function. We denominate this mutation scheme as "fast smart mutation" (FSM), as it allows to increase the effectiveness of the mutation stage in the genetic algorithm.

The evaluation of the objective function for all the possible bit mutations of a pattern is implemented taking into account prior knowledge of the objective function. Let's consider the bit stream $\mathbf{b}$, then the goal is to find the bit or bits which changes produce the highest reduction in the performance associated to $\mathbf{b}$.

The change of one bit of $\mathbf{b}$ produces the addition or the removal of a training pattern from the edited subset. It causes changes in the values of $k_n[i]$, with a consequent change in the value of the objective function, that might be considered. Let's consider $B_n(k)$ is the distance from the $n$-th training pattern to its $k$-th nearest pattern, then:

- If the $m$-th bit changes from 0 to 1, then the pattern $\mathbf{x}_m$ must now be considered in the subset. If the distance from this pattern to a training pattern $\mathbf{x}_n$ is lower than $B_n(k)$, then this new pattern replaces the $k$-th nearest neighbor of the training pattern $\mathbf{x}_n$. Due to the addition of this new pattern of class $c_m$, the value of $k_n[c_m]$ is incremented in 1, and due to the

removal of the $k$-th training pattern, the value of $k_n[c_n^k]$ is decremented in 1, where $c_n^k$ is the class of the $k$-th nearest neighbor of the training pattern $\mathbf{x}_n$.
– If the $m$-th bit changes from 1 to 0, then the pattern $\mathbf{x}_m$ is removed from the subset. If the distance from this pattern to a training pattern $\mathbf{x}_n$ is lower than $B_n(k)$, then this pattern will cause changes in the values of $k_n[i]$. The pattern $\mathbf{x}_m$ will not continue in the group of the $k$ nearest neighbors of the pattern $\mathbf{x}_n$, and there will be a new pattern in this group. Due to the removal of this pattern of class $c_m$, the value of $k_n[c_m]$ is decremented in 1, and due to the inclusion of the $k+1$-th training pattern in the group, the value of $k_n[c_n^{k+1}]$ is incremented in 1.

Equation (6) represents the function $f_{mn}[i]$, the variations in the values of $k_n[i]$ due to a change in the $m$-th.

$$f_{mn}[i] = \begin{cases} D_{mn} \cdot (\delta[i - c_m] - \delta[i - c_n^k]), & if\ b[m] = 0 \\ D_{mn} \cdot (\delta[i - c_n^{k+1}] - \delta[i - c_m]), & if\ b[m] = 1 \end{cases} \tag{6}$$

where $D_{mn}$ is 1 if the distance from the pattern $\mathbf{x}_m$ to the pattern $\mathbf{x}_n$ is lower than $B_n(k)$, and 0 in other case. So, the MSE-based objective function $F_m$ obtained after changing the $m$-th gene is represented in equation (7).

$$F_m = \frac{1}{CN} \sum_{n=1}^{N} \sum_{i=1}^{C} \left( \frac{k_n[i] + f_{mn}[i]}{k} - \delta[i - c_n] \right)^2 \tag{7}$$

The variation in the objective function $\Delta_m = F_m - F$ due to a change in the $m$-th bit can be expressed using equation (8).

$$\Delta_m = \frac{1}{CNk^2} \sum_{n=1}^{N} -2k f_{mn}[c_n] + \sum_{i=1}^{C} f_{mn}[i]^2 + 2k_n[i] f_{mn}[i] \tag{8}$$

Using (6) in (8), we obtain (9).

$$\Delta_m = \frac{2}{k^2 CN} \sum_{n=1}^{N} D_{mn}(1 + g_{mn}) \tag{9}$$

where $g_{mn}$ is defined by equation (10).

$$g_{mn} = \begin{cases} -1, & if\ b[m] = 0\ \ and\ \ c_n^k = c_m \\ h_{0mn}, & if\ b[m] = 0\ \ and\ \ c_n^k \neq c_m \\ -1, & if\ b[m] = 1\ \ and\ \ c_n^{k+1} = c_m \\ h_{1mn}, & if\ b[m] = 1\ \ and\ \ c_n^{k+1} \neq c_m \end{cases} \tag{10}$$

being $h_{0mn}$ and $h_{1mn}$ defined by equations (11) and (12), respectively.

$$h_{0mn} = \begin{cases} k_n[c_m] - k_n[c_n^k] + k, & if\ \ c_n = c_n^k \\ k_n[c_m] - k_n[c_n^k] - k, & if\ \ c_n = c_m \\ k_n[c_m] - k_n[c_n^k], & other\ case \end{cases} \tag{11}$$

$$h_{1mn} = \begin{cases} -k_n[c_m] + k_n[c_n^{k+1}] - k, \; if \;\; c_n = c_n^{k+1} \\ -k_n[c_m] + k_n[c_n^{k+1}] + k, \; if \;\; c_n = c_m \\ -k_n[c_m] + k_n[c_n^{k+1}], \qquad other \; case \end{cases} \tag{12}$$

The value of $\Delta_m$ (equation (9)) is evaluated for all the possible values of $m$, in each generation and for every individual. The described algorithm allows to quickly evaluate the variation of the objective function with a unique bit change. So, the change in the value of $m$ that efforts the lowest $\Delta_m$ will cause the highest reduction of the objective function.

The GA can be speeded up changing more than one bit in every mutation. In many classification environments, the large size of the training set makes this method quite slow, in so only a gene is changed for each individual every mutation stage. On the other hand, using the clustering process described in subsection 3.2, it is possible to establish groups of "independent" genes. A change in a bit that belongs to a cluster affects to the performance of the classifier in the region of the space nearer to the corresponding training pattern. So, we propose to use the gene clusters to select a group of genes to be mutated. For each cluster, the value of $m$ that efforts the lowest $\Delta_m$ is changed, which allows to mutate as many genes as clusters.

The implementation of the algorithm requires the previous calculation of the values of $k_n[i]$, $c_n^k$ and $c_n^{k+1}$. The process of the genetic algorithm with fast smart mutation is described as follows:

1. The initial population with 100 individuals is generated, all the variables are initialized.
2. The mean square error is measured for every individual of the population. The values of $\Delta_m$ are obtained.
3. The k-means algorithm is applied to the training set. The number of clusters is selected at random.
4. For each cluster and each individual, the gene with the value of $m$ that efforts the lowest value of $\Delta_m$ is muted.
5. Every 10 generations, clustered crossover is applied to the data. 10 best individuals are chosen as parents, and remaining 90 individuals are generated by clustered crossover of the parents.
6. The validation error of the best individual is calculated.
7. The process is iterated in step 2, until 100 generations are reached.
8. Finally, the selected individual is the one that achieved the lowest validation error.

## 4 Results

This section includes the results obtained by the methods described in the paper. The databases used in the experiments of the paper have been the breast cancer database and the diabetes database, collected from the UCI machine learning

benchmark repository. Choosing these two databases we try to be able to compare the performance of the different methods in two different environments, allowing to extract more general conclusions.

In order to carry out the experiments, each database has been divided in three subsets: the training set, the validation set and the test set. The training set has been used to generate the edited subsets. The validation set has been used to select the best classifier, and to determine the values of $k$. The test set has been used to evaluate the final error rate for each classifier. This third set has not been used during the design of the classifier. These three databases and the data preparation techniques are identical to those used in other papers [9, 10], allowing to make comparisons of the obtained results with other different type of classifiers. Table 1 shows a summary of the main characteristics of the used two databases.

**Table 1.** Characteristics of the databases.

| | Breast cancer | Diabetes |
|---|---|---|
| Number of classes $C$ | 2 | 2 |
| Number of inputs $L$ | 9 | 8 |
| Total number of patterns | 699 | 768 |
| Number of training patterns $N$ | 349 | 384 |
| Number of validation patterns | 175 | 192 |
| Number of test patterns | 175 | 192 |
| $k$ value (using the validation set) | 3 | 27 |

The parameter $k$ of the kNN method is a user-specific parameter. In this work we have selected it in a first stage, making use of the validation set, and it has remained fixed for the rest of the experiments. Different kNN classifiers with values of $k$ from 1 to 50 have been implemented using each database, and the value of $k$ that efforts the lowest classification error rate over the validation set has been selected. This value has been $k = 3$ for the breast cancer database and $k = 27$ for the diabetes database.

In order to assess the performance of the classification methods, the error rate over the test set is used. Due to the small size of the test sets, the precision in the estimation of the error rate is considerably low, and some statistical analysis of the results must be used. So, each experiment has been repeated 30 times, measuring the error rate for each experiment. Results are represented in function of the mean, the standard deviation, the maximum, the minimum of the error rate over the 30 experiments. The average number of patterns selected in the edited subset ($S$) is also included. Table 2 shows the results obtained by the different methods for the breast cancer database and the diabetes database.

From the obtained results, we can derive the next conclusions:

**Table 2.** Results (%) obtained for the different methods studied in the paper.

| *Editing* | *Breast Cancer* | | | | | *Diabetes* | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| *technique* | *Mean* | *Std* | *Max* | *Min* | *S* | *Mean* | *Std* | *Max* | *Min* | *S* |
| none | 1.14 | 0.00 | 1.14 | 1.14 | 349 | 21.88 | 0.00 | 21.88 | 21.88 | 384 |
| Wilson [1] | 1.71 | 0.00 | 1.71 | 1.71 | 323 | 27.08 | 0.00 | 27.08 | 27.08 | 262 |
| GA CEPT SPC [3] | 1.96 | 1.06 | 4.57 | 0.00 | 101 | 22.76 | 2.00 | 26.04 | 18.75 | 173 |
| GA MSE SPC | 1.43 | 0.76 | 2.86 | 0.00 | 157 | 19.84 | 1.27 | 21.88 | 16.67 | 192 |
| GA MSE RC | 1.68 | 0.78 | 4.00 | 0.57 | 163 | 19.62 | 1.18 | 21.88 | 17.19 | 193 |
| GA MSE CC | 1.22 | 0.65 | 3.43 | 0.00 | 186 | 19.60 | 1.00 | 22.40 | 18.23 | 191 |
| GA MSE CC FSM | 0.72 | 0.54 | 2.29 | 0.00 | 174 | 19.39 | 1.63 | 22.92 | 16.67 | 195 |

- The use of the proposed MSE-based objective function has an associated reduction greater than 12%, when it is compared to the use of the CEPT objective function [3].
- The use of the proposed clustered crossover doest not significantly improve the performance in the case of the diabetes database, but it achieves a reduction of 15% in the error rate in the case of the breast cancer database.
- The results obtained by the joint use of the three proposals has an associated reduction greater than 10%, compared to the use of a kNN classifier without editing technique. Obtained results demonstrate the good accuracy of the proposed GA-based editing technique.

## 5 Conclusions

In this paper genetic algorithms have been successfully applied to select the training patterns included in an edited set of a kNN classifier. We have proposed three improvements of the editing process using genetic algorithms. Considering the statistical properties of the kNN classifier, we have proposed a novel mean square error based objective function, which performs better than the counting estimator based objective function. The second proposal presents an analysis of the relationship of the genes in the GA, which is used to propose a clustered crossover. At last, a new fast smart mutation scheme that allows to quickly evaluate the variations in the MSE-based objective function for a change in one bit is described.

Results achieved using the breast cancer database and the diabetes database from the UCI machine learning benchmark repository have been included. The obtained results make the joint use of the three proposed methods quite interesting. Comparing these results with the best one obtained using kNN without editing, with Wilson's editing, and with GA-based editing using CEPT and SPC, the proposed method achieves an average reduction of greater than 10% for the considered databases (36% for the breast cancer database and 12% for the biabetes database).

## References


1. D.L. Wilson: Asymptotic properties of nearest neighbor rules using edited datasets. IEEE Transactions on Systems, Man and Cybernetics, Vol. 2, (1972) 408-421.
2. R.A. Mollineda, J.S. Sánchez and J.M. Sotoca: Data characterization for effective prototype selection. Lecture Notes in Computer Sciences, Vol. 3523, (2005) 27-34.
3. L.I. Kuncheva: Fitness functions in editing k-NN reference set by genetic algorithms. Pattern Recognition, Vol. 30, No. 6, (1997) 1041-1049.
4. S.Y. Ho, C.C. Liu and S. Liu: Design of an optimal nearest neigbor classifier using an intelligent genetic algorithm. Pattern Recognition Letters, Vol. 23, (2002) 1495-1503.
5. V. Cerverón and F.J. Ferri: Another move toward the minimum subset: a tabu search approach to the condensed nearest neighbor rule. IEEE Transactions on System, Man, and Cybernetics-Part B: Cybernetics, Vol. 31, No. 3, (2001) 408-413.
6. J.-H. Chen, H.-M. Chen and S.-Y Ho: Design of Nearest Neighbor Classifiers Using an Intelligent Multi-objective Evolutionary Algorithm. Lecture Notes in Artificial Intelligence, Vol. 3157, (2004) 262-271.
7. C.M. Bishop: Neural networks for pattern recognition. Oxford University Press Inc, New York (1995).
8. J.A. Hartigan and M.A. Wong: Algorithm AS 163: A k-means clustering algorithm. Applied Statistics, Vol. 28, No. 1, (1979) 100-108.
9. X. Yao and Y. Liu: A new evolutionary system for evolving artificial neural networks. IEEE Transactions on Neural Networks, Vol. 8, No. 3, (1997) 694-713.
10. M.M. Islam, X. Yao and K. Murase: A constructive algorithm for training cooperative neural network ensembles. IEEE Trainsactions on Neural Networks, Vol. 14, No. 4, (2003) 820-834.