# A Stochastic Approach to Wilson's Editing Algorithm

Fernando Vázquez[1], J. Salvador Sánchez[2], and Filiberto Pla[2]

[1] Dept de Ciencia de la Computación, Universidad de Oriente, Av. Patricio Lumunba s/n, Santiago de Cuba, CP 90100, Cuba
fvazquez@csd.uo.edu.cu
[2] Dept. Lenguajes y Sistemas Informáticos, Universitat Jaume I, 12071 Castellón, Spain
{sanchez,pla}@uji.es

**Abstract.** Two extensions of the original Wilson's editing method are introduced in this paper. These new algorithms are based on estimating probabilities from the *k*-nearest neighbor patterns of an instance, in order to obtain more compact edited sets while maintaining the classification rate. Several experiments with synthetic and real data sets are carried out to illustrate the behavior of the algorithms proposed here and compare their performance with that of other traditional techniques.

## 1 Introduction

Among non-parametric statistical classifiers, the approaches based on neighborhood criteria have some interesting properties with respect to other non-parametric methods. The most immediate advantage makes reference to their simplicity, that is, the classification of a new pattern in the feature space is based on the local distribution of patterns in the training set that surround the targeted point.

The Nearest Neighbor (NN) rule [1] is one of the most extensively studied algorithms within the non-parametric classification techniques. Given a set of previously labeled prototypes (a training set, TS), this rule assigns a sample to the same class as the closest prototype in the set, according to a measure of similarity in the feature space. Another extended algorithm is the *k* nearest neighbors rule (*k*-NN), in which a new pattern is assigned to the class resulting from the majority voting of its *k* closest neighbors. Obviously, the *k*-NN rule becomes the NN rule for *k*=1.

In order to achieve an appropriate convergence of the *k*-NN rule, it is well known its asymptotic behavior with respect to the Bayes classifier for very large TS. On the other hand, the larger the TS, the more computational cost is needed, becoming unaffordable for large data sets.

Prototype Selection (PS) techniques for the *k*-NN rule are aimed at selecting prototypes from the original TS to improve and simplify the application of the NN rule. Within the PS techniques, we can differentiate two main approaches. A first category of techniques try to eliminate from the TS prototypes erroneously labeled, commonly outliers, and at the same time, to "clean" the possible overlapping between regions of different classes. These techniques are referred in the literature to as Editing, and the resulting classification rule is known as Edited NN rule [2].

A second group of PS techniques are aimed at selecting a certain subgroup of prototypes that behaves, employing the 1-NN rule, in a similar way to the one obtained

J.S. Marques et al. (Eds.): IbPRIA 2005, LNCS 3523, pp. 35–42, 2005.

by using the totality of the TS. This group of techniques are the so called Condensing algorithms and its corresponding Condensed NN rule [2].

The application of editing procedures are interesting not only as a tool to reduce the classification error associated to NN rules, but also to carry out any later process that could benefit from a TS with simpler decision borders and reduced presence of outliers in the distributions [5], [7]. The Wilson's editing algorithm [6] constitutes the first formal proposal in these PS techniques, which is still widely used because of its effectiveness and simplicity. The present paper presents a new classification rule based on the distances from a sample to its *k*-nearest neighbor prototypes. Using this likelihood rule, we present two modifications of Wilson's editing.

## 2 Editing Algorithms

The common idea to most editing algorithms consists of discarding prototypes that are placed in a local region corresponding to a class different from its [5]. As we will see later, basically the only thing that varies among the various editing algorithms is how they estimate the probability that a sample belongs to a certain class.

All the algorithms employed in this work are based on the *k*-NN classifier. Thus the *k*-NN rule can be formally expressed as follows. Let $\{X, \theta\}=\{(x_1,\theta_1) , (x_2,\theta_2) , \ldots, (x_N,\theta_N)\}$ be a TS with $N$ prototypes from $M$ possible classes, and let $P_j = \{P_{j,i} / i = 1,2,\ldots, N_j \}$ be the set of prototypes from X belonging to class $j$. The neighborhood $N_k(\mathbf{x})$ of a sample $\mathbf{x}$ can be defined as the set of prototypes:

$$N_k(\mathbf{x}) \subseteq P \; ; \quad |N_k(\mathbf{x})| = k$$

$$\forall\, p \in N_k(\mathbf{x}) \,,\, q \in P - N_k(\mathbf{x}) \Rightarrow \mathrm{d}(p, \mathbf{x}) \leq \mathrm{d}(q, \mathbf{x}); \text{ where } \quad P = \bigcup_{i=1}^{M} P_i$$

If we now define a new distance between a point and a set of prototypes such as

$$\mathrm{d}_k (\mathbf{x}, P_i) = k - |N_k(\mathbf{x}) \cap P_i|$$

then the *k*-NN classification rule can be defined as:

$$\delta_{k\text{-NN}}(\mathrm{x}) = \Theta_\mathrm{i} \;\Leftrightarrow\; \mathrm{d}(\mathrm{x}, P_i) = \min_{\mathrm{j}=1,2,\ldots,\mathrm{M}} \mathrm{d}_k (\mathrm{x}, P_j)$$

**Wilson's Editing**

Wilson's editing relies on the idea that, if a prototype is erroneously classified using the *k*-NN, it has to be eliminated from the TS. Thus, all the prototypes in the TS are used to determine the *k* nearest neighbors, except the one being considered, that is, it uses the leaving-one-out error estimate. Thus, the Wilson's editing algorithm [6] can be expressed as follows:

Initialization: $\mathrm{S} \leftarrow \mathrm{X}$
*For each prototype* $\mathrm{x_i} \in \mathrm{X}$ do
    Search for the *k*-nearest neighbors of $x_i$ inside $\mathrm{X} - \{x_i\}$
    If $\delta_{k\text{-NN}}(x_i) \neq \theta_\mathrm{i}$ then $\mathrm{S} \leftarrow \mathrm{S} - \{x_i\}$.

This algorithm provides a set of prototypes organized in relatively compact and homogenous groups. However, for reduced data sets, it turns out incorrect considering that the estimation made on each prototype is statistically independent, which is the basis for a correct interpretation of the asymptotic behavior of the NN rule.

**Holdout Editing**

With the aim of avoiding such restrictions, a new editing algorithm was proposed based on Wilson's scheme, but changing the error estimate method. This algorithm, called Holdout editing [2], consists of partitioning the TS in $m$ not overlapped blocks, making an estimation for each block $j$, using the block (($j$+1) module $m$) to design the sort key. This procedure allows to consider statistical independence, whenever $m > 2$.

Make a random partition of X in $m$ blocs, $T_1$, ...,$T_m$
For each block $T_j$ (j = 1 , . . . , $m$):
  For each $x_i \in T_j$
    Search for the $k$ nearest neighbors of $x_i$ in $T_{((j+1) \bmod m)}$
    If $\delta_{k\text{-NN}}(x_i) \neq \theta_i$ then $X \leftarrow X - \{x_i\}$

**Multiedit**

The scheme based on partitions allows the possibility of repeating the editing process a certain number of times, say $f$ [2]. In this case, the corresponding algorithm is called Multiedit and consists of repeating the Holdout editing but using the 1-NN rule.

1. Initialization : $t \leftarrow 0$
2. Repeat until in the last $t$ iterations ($t > f$) do not take place any prototype elimination from the set X.
   2.1 Assign to S the result of applying Holdout editing on X using the NN rule.
   2.2 If no new elimination has taken place in 2.1, that is, ($|X| = |S|$), then $t \leftarrow t+1$ and go to step 2.
   2.3 else, assign to X the content of S and make $t \leftarrow 0$.

For sufficiently large sets, the main advantage of the iterative version is that its behavior is significantly better because of the fact it does not have a dependency on parameter $k$, opposite to the previous algorithm.

The behavior of the editing approaches based on partition gets worse as the size of the TS decreases. This degradation of the effectiveness becomes more significant when increasing the number of blocks by partition [3]. In fact, for relatively small sets, Wilson's editing works considerably better than the Multiedit algorithm.

## 3 Editing by Estimating Conditional Class Probabilities

For all methods described in previous section, the elimination rule in the editing process is based on the $k$-NN rule. In the editing rules here proposed, the majority voting scheme of the $k$-NN rule is substituted by an estimation of the probability of sample to belong to a certain class.

The rationale of this approach is aimed at using a classification rule based on local information of an instance, like the *k*-NN, but considering the form of the underlying probability distribution in the neighborhood of a point. In order to estimate the values of the underlying distributions, we can use the distance between the sample and the prototypes. Given a sample, the closer a prototype, the more likely this sample belongs to the same class as the one of such a prototype.

Therefore, let us define the probability $P_i(\mathbf{x})$ that a sample **x** belongs to a class *i* as:

$$P_i(x) = \sum_{j=1}^{k} p_i^j \frac{1}{(1 + d(x, x^j))}$$

where $p_i^j$ denotes the probability that the *k*-nearest neighbor $x^j$ belongs to class *i*. Initially, the values of $p_i^j$ for each prototype are set to 1 for its class label assigned in the TS, and 0 otherwise. These values could change in case an iterative process is used, but this is not the case in the approach we are presenting here.

The meaning of the above expression states that the probability that a sample **x** belongs to a class *i* is the weighted average of the probabilities that its *k*-nearest neighbors belong to that class. The weight is inversely proportional to the distance from the sample to the corresponding *k*-nearest neighbor. After normalizing,

$$p_i(x) = P_i(x) / \sum_{j=1}^{M} P_j(x)$$

the class *i* assigned to a sample **x** is estimated by the decision rule

$$\delta_{\text{k-prob}}(\text{x}) = i \; ; \quad i \,/\, p_i(x) = \arg\max_j (p_j(x))$$

Using this rule, we propose the editing algorithms described below applying a Wilson's scheme, that is, if the class assigned by the above decision rule does not coincide with the class label of the sample, this sample will be discarded. As we will show in the experiments, the use of the rule just introduced, instead of the *k*-NN rule, makes the editing procedure to produce a TS with a good trade-off between TS size and classification accuracy, because of the fact that such a decision rule estimates in a more accurate way the values of the underlying probability distributions of the different classes, estimating locally these values from the *k*-nearest neighbor samples.

**Editing Algorithm Estimating Class Probabilities (WilsonProb)**

1 Initialization: S ←X
2 For each prototype $x \in$ X do
  2.1 Search for the *k* nearest neighbors of *x* inside X−{*x*}
  2.2 If $\delta_{k\text{-prob}}(\text{x}) \neq \theta$ , then S ← S − {*x*}, θ denotes the class of the object *x*.

**Editing Algorithm Estimating Class Probabilities and Threshold (WilsonTh)**

A variation of the previous algorithm consists of introducing a threshold, $0<\mu<1$ , in the classification rule, with the aim of eliminating those instances whose probability

to belong to the class assigned by the rule is not significant. Correspondingly, we are removing samples from the TS that are in the decision borders, where the class conditional probabilities overlap and are confusing, in order to obtain edited sets whose instances have a high probability of belonging to the class assigned in the TS.

1 Initialization: S $\leftarrow$X
2 For each prototype $x \in$ X do
    2.1 Search for the $k$ nearest neighbors of $x$ inside X $-$ $\{x\}$
    2.2 If $\delta_{k\text{-prob}}(x) \neq \theta$ or $p_j \leq \mu$, do S $\leftarrow$ S $-$ $\{x\}$, $p_j$ is the maximum of all the probabilities of the object $x$ to belong to a class.

## 4 Experimental Results and Discussion

In this section, the behavior of the editing algorithms just introduced is analyzed using 14 real and synthetic databases taken from the UCI Machine Learning Database Repository [4]. The main characteristics of these data sets are summarized in Table 1.

**Table 1.** A brief summary of the experimental databases

| | No. classes | No. features | No. instances |
|---|---|---|---|
| Cancer | 2 | 9 | 683 |
| Liver | 2 | 6 | 345 |
| Heart | 2 | 13 | 270 |
| Wine | 3 | 13 | 178 |
| Australian | 2 | 42 | 690 |
| Balance | 3 | 4 | 625 |
| Diabetes | 2 | 8 | 786 |
| German | 2 | 24 | 1002 |
| Glass | 6 | 9 | 214 |
| Ionosphere | 2 | 34 | 352 |
| Phoneme | 2 | 5 | 5404 |
| Satimage | 6 | 36 | 6453 |
| Texture | 11 | 40 | 5500 |
| Vehicle | 4 | 18 | 846 |

The experiments consist of applying the 1-NN rule to each of the test sets, where the training portion has been preprocessed by means of different editing techniques. In particular, apart from the schemes here proposed, Wilson's editing, the Holdout method and the Multiedit algorithm have been also included in this comparative study. The 5-fold cross-validation method (80% of the original instances have been used as the TS and 20% for test purposes) has been here employed to estimate the overall classification accuracy and size reduction rates.

Table 2 reports the experimental results (classification accuracy and size reduction) yielded by the different algorithms over the 14 databases. These results have been averaged over the five partitions. Bold figures indicate the bests methods in terms of classification accuracy for each database. Italics indicates the highest size reduction. Note that typical settings for the algorithms used in the present study have been tried and the ones leading to the best performance have been finally included in Table 2. In

the case of WilsonTh, we provide the results obtained from using different values of the threshold parameter. The results corresponding to the plain NN classifier over the original TS have been also included for comparison purposes.

**Table 2.** Classification accuracy (acc) and size reduction rate (size) using different editings

| | | NN | Wils. | Hold. | Mult. | WProb | WilsonTh | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | 0.6 | 0.7 | 0.8 |
| Cancer | acc | 95.60 | 96.19 | 96.63 | 96.63 | 96.34 | 96.48 | 96.63 | **96.78** |
| | size | | 3.44 | 4.28 | 7.43 | 3.36 | 4.09 | 5.49 | *7.68* |
| Liver | acc | 65.79 | **70.70** | 70.40 | 59.49 | 68.67 | 68.97 | 69.55 | 68.95 |
| | size | | 32.89 | 37.10 | 75.79 | 27.89 | 45.94 | 61.37 | *67.82* |
| Glass | acc | **71.40** | 67.62 | 66.03 | 58.63 | 66.16 | 63.97 | 62.29 | 62.31 |
| | size | | 28.50 | 46.14 | *61.21* | 36.68 | 20.32 | 50.58 | 58.17 |
| Heart | acc | 58.16 | 67.00 | **67.34** | 66.64 | 66.26 | 65.17 | 65.12 | 64.78 |
| | size | | 34.44 | 38.70 | *69.25* | 28.51 | 40.09 | 53.61 | 65.09 |
| Vehicle | acc | **64.41** | 60.26 | 63.22 | 52.81 | 62.16 | 61.32 | 61.08 | 59.67 |
| | size | | 36.08 | 39.83 | *66.66* | 20.41 | 43.17 | 46.01 | 58.86 |
| Wine | acc | **73.04** | 70.90 | 75.24 | 72.42 | 69.69 | 69.74 | 69.20 | 69.20 |
| | size | | 34.97 | 30.75 | *45.50* | 14.60 | 33.28 | 35.67 | 41.43 |
| Ionosphere | acc | **83.46** | 82.02 | 82.31 | 69.58 | 81.74 | 81.74 | 80.89 | 80.64 |
| | size | | 16.66 | 14.52 | *34.11* | 18.01 | 18.01 | 24.21 | 25.21 |
| Texture | acc | **98.96** | 98.63 | 98.56 | 94.62 | 98.74 | 98.49 | 98.29 | 98.32 |
| | size | | 1.34 | 3.69 | *15.31* | 1.01 | 1.50 | 3.17 | 3.06 |
| Balance | acc | 79.20 | 85.11 | 85.62 | 86.41 | 84.96 | 86.73 | 88.50 | **89.13** |
| | size | | 14.80 | 14.52 | 37.04 | 10.76 | 24.40 | 32.08 | *38.40* |
| Australian | acc | 65.67 | 69.27 | **70.72** | 68.99 | 69.56 | 69.70 | 68.39 | 68.54 |
| | size | | 31.88 | 36.88 | *59.52* | 25.90 | 37.02 | 50.76 | 57.53 |
| German | acc | 64.81 | 70.40 | **72.00** | 70.00 | 70.70 | 71.10 | 70.50 | 70.50 |
| | size | | 30.50 | 32.27 | 54.72 | 26.90 | 39.62 | 52.72 | *60.00* |
| Phoneme | acc | 70.26 | 73.53 | 74.29 | **75.35** | 73.42 | 73.44 | 74.02 | 73.99 |
| | size | | 10.56 | 16.07 | *37.43* | 11.98 | 17.26 | 24.36 | 29.15 |
| Satimage | acc | **83.62** | 83.29 | 83.32 | 82.35 | 83.09 | 83.18 | 83.24 | 83.50 |
| | size | | 9.43 | 10.19 | *24.51* | 9.25 | 15.61 | 19.22 | 23.90 |
| Diabetes | acc | 67.32 | 73.70 | 73.69 | 71.09 | 74.35 | 74.60 | 74.48 | **74.74** |
| | size | | 26.36 | 44.40 | *55.76* | 21.09 | 37.33 | 45.47 | 54.91 |

The first significant result from this empirical analysis is that the editing algorithms here proposed obtain similar classification accuracy to that of other classical methods. It is especially remarkable the fact that no editing outperforms the plain NN classifier in 6 out of 14 databases, although differences in such cases are not statistically significant. Focusing on these results, it seems rather difficult to draw any conclusion because of the little significant differences among them in terms of accuracy.

Examining the other factor of interest in Table 2, that is, the size reduction, the results show that both Multiedit and the proposed WilsonTh achieve the highest rates in all cases, consequently giving the most important decrease in computational loads in the classification phase. Although Multiedit achieves the highest set size reduction rate almost in all databases (10 out of 14), differences with respect to WilsonTh are only marginal. A final remark from the experiments, and perhaps the most important one, refers to the fact of comparing both classification accuracy and reduction rate simultaneously, WilsonTh outperforms Multiedit in most cases. In other words, the proposed WilsonTh algorithm obtains a better trade-off between accuracy and reduction than that given by Multiedit (or any other editing).

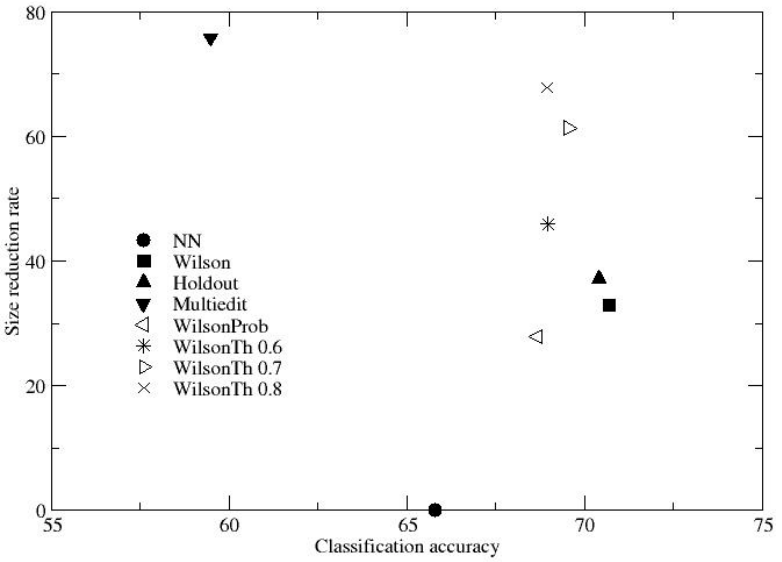


**Fig. 1.** Comparing classification accuracy and set size reduction for different editing methods over the Liver database

In order to assess the performance relative to these two competing goals simultaneously, Fig. 1 illustrates the behaviour of the editing techniques in terms of both classification accuracy and set size reduction over the Liver database. As can be observed, Multiedit algorithm yields the highest reduction rate, but it produces a very poor classification accuracy. On the contrary, Wilson's editing obtains the highest classification accuracy, but it retains too many training instances. Finally, WilsonTh schemes (0.7 and 0.8) provide a suitable trade-off between both issues: high enough classification accuracy and reduction rate.

## 5 Concluding Remarks

When using a NN classifier, the presence of mislabeled prototypes can strongly degrade the corresponding classification accuracy. Many models for identifying and removing outliers have been devised. In this paper, we propose two editing algorithms that consider the probabilities of an instance to belong to a class.

A series of experiments over 14 data sets has been carried out in order to evaluate the performance of those new editing methods and compare them with other traditional techniques. From the experiments carried out, it is to be noted that the two stochastic approaches to Wilson's editing attain a suitable trade-off between TS size and classification accuracy.

These editing methods are currently being applied in research works about ongoing learning, where throughout these processes, it is necessary to eliminate erroneously classified instances in the TS, with the objective of improving the classifier, acquiring experience from new unlabeled samples to be incorporated in the TS.

## Acknowledgements

This work has been partially supported by projects IST-2001-37306 from EU, TIC2003-08496 from the Spanish CICYT, GV04A/705 from Generalitat Valenciana, and P1-1B2004-08 from Fundació Caixa Castelló-Bancaixa...

## References

1. Dasarathy, B. V.: Nearest Neighbor Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamos, CA (1991)
2. Devijver, P.A., Kittler, J.: Pattern Recognition: A Statistical Approach. Prentice Hall, Englewood Cliffs, NJ (1982).
3. Ferri, F.J., Albert, J.V., Vidal, E.: Consideration about sample-size sensitivity of a family of edited nearest-neighbor rules. IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics 29 (1999) 667-672.
4. Merz, C.J., Murphy., Murphy, P.M.: UCI Repository of Machine Learning Database. Department of Information and Computer Science, University of California, Irvine, CA (1998).
5. Sánchez, J.S., Barandela, R., Marqués, A.I., Alejo, R., Badenas, J. : Analysis of new techniques to obtain quality training sets. Pattern Recognition Letters 24 (2003) 1015-1022.
6. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data sets. IEEE Trans. on Systems, Man and Cybernetics 2 (1972) 408-421.
7. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Machine Learning 38 (2000) 257-286.