ELSEVIER

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Pattern Recognition Letters 24 (2003) 1015–1022

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Analysis of new techniques to obtain quality training sets

J.S. Sánchez [a,*], R. Barandela [b], A.I. Marqués [a], R. Alejo [b], J. Badenas [a]

[a] *Universitat Jaume I, Av. Vicent Sos Baynat s/n, 12006 Castellón, Spain*
[b] *Instituto Tecnológico de Toluca, Av. Tecnológico s/n, 52140 Metepec, Mexico*

Abstract

This paper presents new algorithms to identify and eliminate mislabelled, noisy and atypical training samples for supervised learning and more specifically, for nearest neighbour classification. The main goal of these approaches is to enhance the classification accuracy by improving the quality of the training data. Several experiments with synthetic and real data sets are carried out in order to illustrate the behaviour of the schemes proposed here and compare their performance with that of other traditional techniques. It is also analysed the ability of these new algorithms to "reduce" the possible overlapping among regions of different classes.

Keywords: Nearest neighbour; Editing; Classification accuracy; Nearest centroid neighbourhood; Outlier; Quality training set

1. Introduction

One goal of any learning algorithm is to form a generalization from a set of labelled training samples such that the classification accuracy for new samples is maximised. The maximum accuracy achievable depends on the quality of the input data and on the appropriateness of the chosen learning algorithm for the data.

The work described in this paper concentrates on improving quality of training data by identifying and eliminating mislabelled and atypical samples prior to applying the chosen learning scheme, thereby increasing classification accuracy. An immediate positive effect of eliminating such samples refers to the fact that the possible overlapping among different classes is drastically reduced.

The problem of handling mislabelled, atypical and noisy training samples has been the focus of much attention in both pattern recognition and machine learning domains (Devijver and Kittler, 1982; Brodley and Friedl, 1999; Wilson and Martinez, 2000). For example, extensive efforts have been given to the improvement of the classification performance of the well-known nearest neighbour (NN) rule.

Accordingly, this paper addresses the problem of selecting prototypes in order to improve the classification accuracy of an NN classifier. In this context, some approaches to remove *outliers* from the training set (TS) are here introduced. An

[*] Corresponding author. Tel.: +34-964-728350; fax: +34-964-728435.

E-mail addresses: sanchez@uji.es (J.S. Sánchez), rbarandela@hotmail.com (R. Barandela).

0167-8655/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0167-8655(02)00225-8

outlier has traditionally been defined as a prototype that does not follow the same model as the rest of the data (Weinsberg, 1985). Now this term is being employed to cover a broad range of circumstances, including noisy and atypical data, new unidentified patterns that do not belong to any of the classes represented in the TS, and also mislabelled prototypes. From a practical point of view, a quality TS can be seen as a data set without outliers and consequently, the decision boundaries derived from it will result much simpler.

In addition, an empirical study is here provided in order to compare the classification performance of new approaches to that of other well-known prototype selection techniques. A second experiment evaluates the capability of our algorithms to "clean" the possible overlapping among regions of different classes. We conclude with a summary of the main results presented in this paper and a discussion on future research directions, which are primarily aimed at handling the imbalanced training sample problem by combining some of the techniques described here.

2. The *k*-NN classification rule

One of the most widely studied non-parametric classification approaches corresponds to the k-NN rule. Given a set of n previously labelled prototypes or TS, the k-NN classifier (Dasarathy, 1991) consists of assigning an input sample to the class most frequently represented among the k closest prototypes in the TS, according to a certain dissimilarity measure. A particular case of this rule is when $k = 1$, in which each input sample is assigned to the class indicated by its closest neighbour.

The asymptotic classification error of the k-NN rule (that is, when n grows to infinity) tends to the optimal Bayes error rate as $k \to \infty$ and $k/n \to 0$. Moreover, if $k = 1$, the error is bounded by approximately twice the Bayes error (Devijver and Kittler, 1982). This behaviour in asymptotic classification performance combines with a conceptual and implementational simplicity, which makes it a powerful classification technique capable of dealing with arbitrarily complex problems, provided there is a large enough TS available.

However, in many practical settings, this theoretical behaviour can hardly be achieved because of certain inherent weaknesses that significantly reduce the applicability of k-NN classifiers in real-world tasks. For example, the performance of these rules, as with any non-parametric approach, is extremely sensitive to incorrectness or imperfections in the TS.

That is the reason why a considerable amount of works have been devoted to improve the NN classification accuracy by eliminating outliers from the original TS and also cleaning possible overlapping among classes. This strategy has generally been referred as to *editing* (Devijver and Kittler, 1982), whereas the corresponding classifier has been called *edited NN rule*.

2.1. On editing the NN rule

The general idea behind almost any editing procedure consists of estimating the true classification of prototypes in the TS to retain only those which are correctly labelled. Differences among most editing schemes refer to the classification rule employed for editing purposes along with the error estimate and the stopping criterion (Ferri et al., 1999).

The first proposal to select a representative subset of prototypes for a further NN classification corresponds to Wilson's editing algorithm (Wilson, 1972), in which a k-NN classifier (in experiments k was set to three) is used to retain in the TS only "good" samples (that is, training samples that are correctly classified by the k-NN rule). Tomek (1976) extended this approach with a procedure that utilised all the l-NN classifiers, with l ranging from 1 through k, for a given value of k.

Koplowitz and Brown (1981) proposed an alternative to Wilson's scheme in which some samples are discarded from the TS and others are relabelled according to the classification of a k-NN rule. More recently, a genetic algorithm was proposed (Kuncheva, 1995) as a way of editing the NN classification rule. Sánchez et al. (1997a) introduced a method to select training prototypes by using some proximity graphs.

An alternative to traditional prototype selection consists of generating new samples to replace the original TS. Kohonen (1990) proposed a method, called learning vector quantization, to adjust the samples of the NN classifier by considering the boundary information of confusing classes so that the samples are correctly classified by a small number of prototypes.

3. New algorithms to select training samples

The present section describes some new procedures for handling outliers in a TS. The first scheme has not the same results as the general editing algorithms, but it is also aimed at refining the structure of the TS and consequently, at increasing the classification accuracy of the NN rule. On the other hand, the second and third alternatives are based on the employment of a surrounding neighbourhood to obtain a filtered TS, that is, to detect and remove outliers from the TS.

3.1. Depuration

The first prototype selection technique (Barandela and Gasca, 2000) consists of removing some "suspicious" samples from the TS and also changing the class labels of some other instances. Its purpose is to cope with all types of imperfections of the training prototypes (mislabelled, noisy and atypical cases).

This method is based on the *generalised editing* (Koplowitz and Brown, 1981), in which two parameters have to be defined: k and k', in such a way that $(k+1)/2 \leqslant k' \leqslant k$. The algorithm can be written as follows:

```
1. Let $S = X$ ($X$ is the original TS, and $S$ will be the edited TS).
2. For each $x_i \in X$ do:
   • Find the $k$-NN of $x_i$ in $X - \{x_i\}$.
   • If a class has at least $k'$ representatives among the $k$ neighbours, change the label of $x_i$ according to that class. Otherwise, discard $x_i$ from $S$.
```

In brief, the depuration algorithm consists of two main stages. The first step is to repeat the application of the generalised editing until stability is reached in the structure of the TS and in the estimated error rate (by leaving-one-out). Afterwards, the second step is to utilise the Wilson's method, perhaps also iterated.

3.2. Editing with the nearest centroid neighbourhood

The nearest centroid neighbourhood (Chaudhuri, 1996) refers to a concept in which neighbourhood is defined taking into account not only the proximity of prototypes to a given input sample but also their *symmetrical distribution* around it. From this general idea, the corresponding classification rule, the k-nearest centroid neighbours (k-NCN) (Sánchez et al., 1997b), has been proven to overcome the traditional k-NN classifier in many practical situations.

Now the editing approach presented here corresponds to a slight modification of the original work of Wilson and basically consists of using the leaving-one-out error estimate with the k-NCN classification rule. Algorithmically, the k-NCN editing scheme can be expressed as follows:

```
1. Let $S = X$.
2. For each $x_i$ in $X$ do:
   • Discard $x_i$ from $S$ if it is misclassified using the $k$-NCN rule with prototypes in $X - \{x_i\}$.
```

3.3. Iterative k-NCN editing

This alternative basically consists of editing repeatedly the already edited TS until no more prototypes are removed. The idea of this moderate extension to the k-NCN algorithm is that, if a single application of editing can generally improve the NN classification accuracy, it is expected that successive deletions of outliers can mean an additional increase in performance. The procedure can be summarised as follows:

```
1. Let $S = \emptyset$.
2. While $S \neq X$
   • Let $S = X$.
   • Assign to $X$ the result of applying the $k$-NCN editing to $S$.
```

4. Experiments and results

In this section, two different sets of experiments have been carried out. The purpose of the first is to test the classification accuracy of our editing approaches as well as that achieved by Wilson's algorithm and (All k)-NN editing. The second experiment aims at comparing the k-NCN scheme with its iterative version in order to evaluate the effect of repeating the editing process. In this case, the focus is on the ability of each procedure to cope with the overlapping among classes.

All experiments consist of applying the NN rule to each of the test sets, where the training portion has been preprocessed by means of an editing algorithm. This gives a check on the power of the procedures to select the most "efficient" prototypes.

4.1. Performance evaluation on real data

From the UCI Repository (Merz and Murphy, 1998), five standard benchmark data sets have been chosen in order to study the behaviour of the different editing algorithms introduced in the previous section. To increase statistical significance of the results in domains with a limited number of prototypes, the N-fold cross validation technique (with $N = 5$) has been applied to the experiments in this paper. About 80% out of the total number of samples available has been used for the TS and the rest for a test set. The results reported here correspond to the average over the five random partitions.

Table 1 summarises the main characteristics of each data set: the number of different class labels, the number of attributes, the number of prototypes in the TS and also the number of instances in the test set.

Table 1
A brief summary of the experimental databases

Data set	No. classes	No. features	TS size	Test set size
Liver	2	6	276	69
Pima	2	8	615	153
Cancer	2	9	546	137
Heart	2	13	216	54
Wine	3	13	144	34

For each domain, *no editing* (NN), *Wilson's editing* (W(k)), *(All k)-NN editing* (A(k)), *Depuration* (D(k, k')), *k-NCN editing* (NCN(k)) and *iterative k-NCN editing* (I(k)) have been considered in this experiment. Only one typical set of parameters ($k = 3$, $k' = 2$) has been tried for each editing.

Table 2 provides the average classification accuracy achieved by the NN rule with the resulting edited sets. The recognition rate for each entire original TS (i.e., no editing) has also been included for comparison purposes. Values in brackets correspond to the standard deviation, whereas bold indicates the best method for each domain.

Results in Table 2 show that, on average, the best alternative for these data sets corresponds to the depuration technique, followed by the iterative k-NCN editing and also its plain version. In fact, one can see that the depuration scheme consistently outperforms all the other algorithms. On the other hand, the performance of the k-NCN editing and that of its iterative extension are very similar and they are not too far from that achieved by the depuration approach.

Finally, it is worth noting that, in the Wine database, the NN rule without editing leads to the highest classification accuracy; this suggests that the sets available for this particular domain are not large enough for properly applying the current editing algorithms.

4.2. Analysis of the k-NCN editings

For the experiments considered in this section, a two-dimensional artificial database with two classes has been employed. The first class is represented by a multivariate normal distribution with zero mean and standard deviation equal to 1, and the second class by a normal distribution with zero mean and standard deviation equal to 2. For this domain, with a relatively large number of samples (2500 from each class), a single random partition into training and test sets has been performed.

The aim of this experiment is to compare the plain k-NCN editing with its iterative algorithm in a problem with a very high overlapping degree between the regions of both classes (see Fig. 1a).

Table 2
Classification rates of the NN rule applied to different edited sets

	Liver	Pima	Cancer	Heart	Wine	Average
NN	65.2 (4.82)	63.9 (5.70)	95.6 (2.49)	58.2 (6.23)	**72.3** (3.37)	71.04 (14.62)
W(3)	69.3 (6.24)	72.0 (2.59)	96.0 (1.90)	64.4 (1.39)	71.8 (8.02)	74.70 (12.29)
A(3)	68.1 (7.39)	71.7 (3.84)	96.3 (2.28)	65.2 (2.41)	67.7 (5.5)	73.80 (12.79)
D(3,2)	**70.3** (7.15)	**75.9** (2.58)	**96.7** (2.34)	**69.3** (5.32)	70.6 (11.76)	**76.56** (11.55)
NCN(3)	69.9 (4.62)	72.9 (3.08)	96.0 (1.90)	67.4 (1.89)	68.2 (2.20)	74.88 (11.99)
I(3)	69.6 (6.38)	72.9 (2.85)	96.3 (2.24)	68.5 (4.06)	68.8 (5.13)	75.22 (11.91)

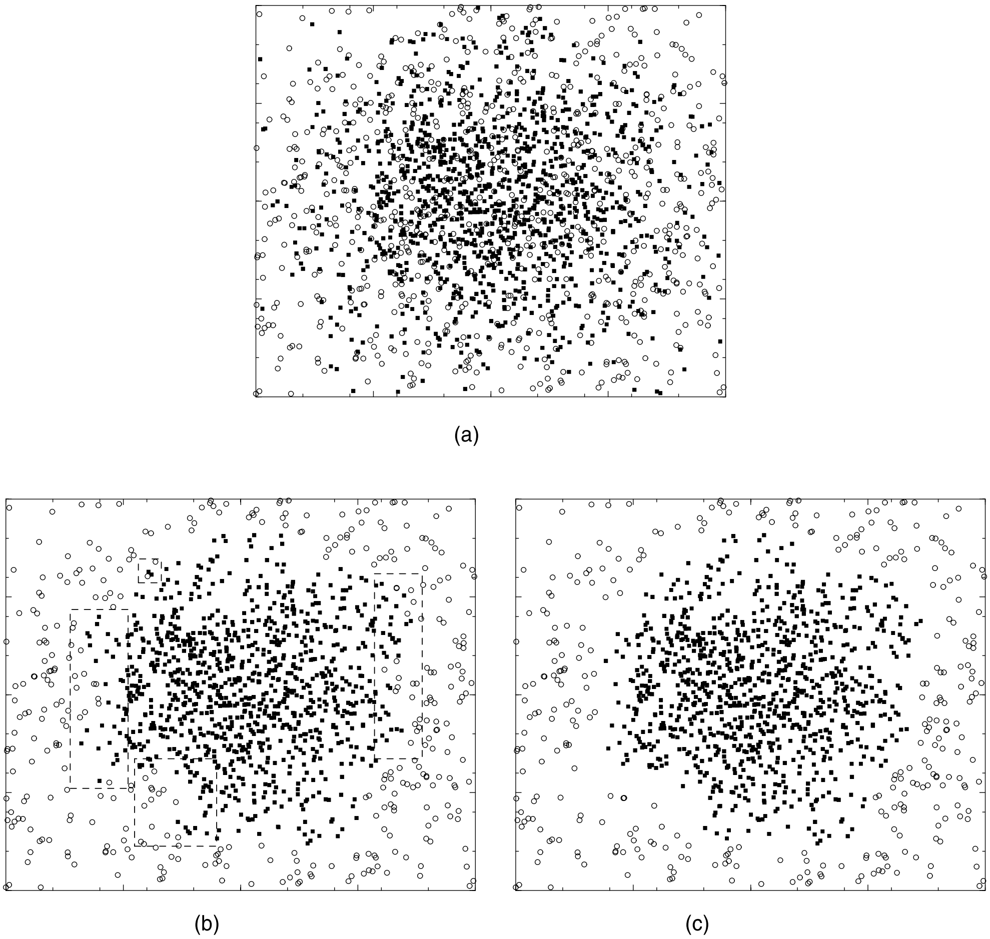

Fig. 1. The artificial data set before and after "cleaning" the overlapping: (a) the original TS (no editing), (b) the TS edited with NCN(35) and (c) the TS edited with I(33).

For this experiment, the values of k used for editing the TS are taken in the range of $1 \leqslant k \leqslant 35$ (only odd values are considered to avoid ties). Then, classification accuracy is estimated by applying the NN rule to the edited sets. It is to be noted that the classification rate achieved by the NN classifier without editing (i.e., using the 2500 prototypes available in the TS) was 64.40%.

Table 3 summarises some results obtained with the plain k-NCN editing and its iterative version:

Table 3
Comparing the k-NCN editing with the iterative extension

k	k-NCN		Iterative k-NCN		
	Set size	Accuracy	Set size	Accuracy	Iterations
1	1601	67.64	1561	68.92	2
3	1738	69.00	1590	70.84	5
5	1738	70.12	1644	71.68	6
7	1749	69.80	1659	72.32	5
9	1800	69.80	1695	72.24	6
11	1786	70.08	1708	72.08	4
13	1799	70.40	1724	72.04	5
15	1807	70.08	1736	72.12	6
17	1799	71.08	1732	72.56	8
19	1802	71.24	1742	72.28	5
21	1796	71.40	1741	72.56	5
23	1792	71.24	1743	72.52	5
25	1805	71.56	1760	72.68	5
27	1797	71.60	1755	72.60	5
29	1813	71.64	1758	72.68	5
31	1820	71.76	1765	72.76	4
33	1820	71.88	1771	**72.88**	4
35	1820	**72.16**	1770	72.68	5

number of prototypes taken from the original TS to form the edited set, classification accuracy, and also number of iterations until no more instances are removed. Once again, bold represents the highest classification accuracy achieved by each editing.

As can be observed from the results in Table 3, the iterative procedure systematically outperforms the plain k-NCN editing for all values of the parameter k. It is also to be remarked the fact that the highest differences are found with small values of k.

On the other hand, in both cases, the highest classification accuracy is over 72% (that is, about 8% more than the classification performance achieved by the NN rule without editing the TS). Nevertheless, while the plain k-NCN editing reaches this "optimum" rate with $k = 35$, the iterative procedure increases its performance much more quickly; for example, with $k = 7$, the classification accuracy is even higher than that of the best plain k-NCN. Therefore, it seems that the iterative approach is found to yield better results with relatively small values of k.

With respect to the edited TS size, in both approaches, the condition for a prototype to be included in the edited set becomes severer with increasing k and therefore, the number of prototypes in the TS is generally higher as the parameter k increases. On the other hand, as is to be expected from the algorithmic description, the plain k-NCN scheme retains more prototypes in the TS than the iterative method. For example, comparing the edited set size for the best k in each procedure, the NCN(35) editing (72.16%) retains 1820 prototypes, whereas the iterative I(33) scheme (72.88%) preserves only 1771 samples.

Moreover, the classification accuracy achieved by the I(7) algorithm with 1659 instances is even higher than that of the plain NCN(35) editing with 1820 samples. From this result, it seems that the prototypes selected by means of the iterative extension are "better" than those of the plain editing. Finally, another aspect to be mentioned refers to the fact that the algorithm converges to a solution in a relatively low number of iterations (about five times).

One can also see in Table 3 that the plain NCN(31), NCN(33) and NCN(35) schemes retain the same number of prototypes in the resulting set. Nevertheless, the classification accuracy increases with k. This means that quality of the NCN(35) edited set is higher than that of the other settings. In other words, the prototypes selected by the NCN(35) algorithm approximate better the NN decision boundaries than the instances selected by the NCN(31) and NCN(33) procedures. An analogous situation is found with $k = 3$ and 5: both of them retain 1738 samples, but the NCN(5) classification accuracy is 1.12% higher than that of the NCN(3).

Fig. 1 depicts the original TS and the edited sets using the plain and the iterative k-NCN algorithms with the best k. Samples from class 1 are shown with black squares and samples from class 2 are shown with white circles. Boxes in Fig. 1b show regions of the feature space in which overlapping remains after applying the plain NCN(35) editing.

In general, it is clear that both the plain and the repeated application of editing extremely reduce the high overlapping degree between classes 1 and 2. Nevertheless, it is important to compare the regions represented by a box in Fig. 1b with respect to those in Fig. 1c. In these specific regions, there is enough evidence to say that the iterative procedure has been able to clean better the strong overlap between both classes than the plain k-NCN editing and as a by-product, the NN decision boundaries approximated by the iterative scheme are much simpler than those of the plain algorithm. Consequently, in practice, the computational burden to classify new input samples will be lower when using the iterative I(33) edited set than in the case of employing the NCN(35) edited set.

5. Conclusion and further extensions

When using an NN classifier, the presence of mislabelled prototypes can strongly degrade the corresponding classification accuracy. Many models for identifying and removing outliers have been proposed. This paper has reviewed some works in the frame of editing the NN rule and three new approaches have been described. A number of experiments over five real data sets have been carried out in order to evaluate the precision of those new editing methods and compare to other traditional procedures. The experiments illustrate that depuration and k-NCN editing generally improve the classification accuracy of the Wilson's algorithm.

Although the percentage reduction of training samples has not been included in this paper, it is to be mentioned that all editing procedures gave very similar results. Nevertheless, as is to be expected, the iterative version of k-NCN provided the highest reduction rate (about 20% more reduction than the rest of schemes).

Future work is primarily addressed to investigate the potential of these editing methods applied to problems where one class is much more represented than the others in the TS. It has been observed that this situation, which arises in several practical domains, may produce an important deterioration of the classification accuracy, specially with patterns belonging to the less represented class. In this context, we have already carried out the first experiments using a fusion of techniques based on the k-NCN editing, obtaining very promising results (Barandela et al., 2001).

Acknowledgements

This work has partially been supported by grants TIC2000-1703-C03-03 from the Spanish CICYT (Department of Science and Technology) and 32016-A from the Mexican Conacyt.

References

Barandela, R., Gasca, E., 2000. Decontamination of training data for supervised pattern recognition methods. In: Advances in Pattern Recognition Lecture Notes in Computer Science 1876. Springer-Verlag, pp. 621–630.

Barandela, R., Sánchez, J.S., García, V., Rangel, E., 2001. Fusion of techniques for handling the imbalanced training sample problem. In: Proceedings of 6th Ibero American Symposium on Pattern Recognition, pp. 34–40.

Brodley, C.E., Friedl, M.A., 1999. Identifying mislabeled training data. Journal of Artificial Intelligence Research 11, 131–167.

Chaudhuri, B.B., 1996. A new definition of neighborhood of a point in multi-dimensional space. Pattern Recognition Letters 17, 11–17.

Dasarathy, B.V., 1991. Nearest Neighbor Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamos, CA.

Devijver, P.A., Kittler, J., 1982. Pattern Recognition: A Statistical Approach. Prentice Hall, Englewood Cliffs, NJ.

Ferri, F.J., Albert, J.V., Vidal, E., 1999. Considerations about sample-size sensitivity of a family of edited nearest-neighbor rules. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 29, 667–672.

Kohonen, T., 1990. The self-organizing map. Proceedings of IEEE 78, 1464–1480.

Koplowitz, J., Brown, T.A., 1981. On the relation of performance to editing in nearest neighbor rules. Pattern Recognition 13, 251–255.

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

Merz, C.J., Murphy, P.M., 1998. UCI Repository of Machine Learning Databases. Department of Information and Computer Science, University of California, Irvine, CA.

Sánchez, J.S., Pla, F., Ferri, F.J., 1997a. Prototype selection for the nearest neighbour rule through proximity graphs. Pattern Recognition Letters 18, 507–513.

Sánchez, J.S., Pla, F., Ferri, F.J., 1997b. On the use of neighbourhood-based non-parametric classifiers. Pattern Recognition Letters 18, 1179–1186.

Tomek, I., 1976. An experiment with the edited nearest neighbor rule. IEEE Transactions on Systems, Man and Cybernetics 6, 448–452.

Weinsberg, S., 1985. Applied Linear Regression. John Wiley and Sons.

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

Wilson, D.R., Martinez, T.R., 2000. Reduction techniques for instance-based learning algorithms. Machine Learning 38, 257–286.