Forgetting Superfluous Information in Supervised Pattern Recognition Systems with Ongoing Learning

Ricardo Barandela[a], Francesc J. Ferri[b], J. Salvador Sánchez[c], Mariela Juárez[a]

[a] Instituto Tecnológico de Toluca, Av. Tecnológico s/n, 52140 Metepec, México
[b] Dept. d'Informàtica, U. Valencia, 46100 Burjassot (Valencia), Spain
[c] Dept. Llenguatges i Sistemes Informàtics, U. Jaume I, 12071 Castelló, Spain

Abstract. Ongoing learning refers to the possibility of a system to increase knowledge from the experience obtained when working in the classification of new patterns. In this paper, we present an automatic classification system with ongoing learning capabilities and analyze the importance of using some size reduction algorithm to remove redundant training patterns.

1 Introduction

Learning algorithms and pattern recognition methods have been usually sorted into two broad groups: supervised and unsupervised (predictive and informative in Data Mining terminology) whether training data is available or not, that is, according to the level of previous knowledge about the training instances identifications in the problem to be solved. Supervised classifier's design is based on the information supplied by a training sample (TS), a set of training patterns, instances or prototypes, that are assumed to represent all the relevant classes and to bear correct class labels. Violation of these assumptions may seriously degrade the classification accuracy.

Supervised classification methods operate usually in two chronologically non-overlapping stages:

a) the learning or training phase, for the system to acquire the necessary knowledge from the TS to make itself able to differentiate among the regarded classes
b) the classification or working phase, wherein the system proceeds to identify new unknown patterns as members of the considered classes. Second phase is not started before completion of the first one and thereafter no new knowledge acquisition is attempted.

The present paper discuss an idea to implement a classification system with an ongoing learning capability. That is, a system that not only can learn with the manipulation of the TS but could also benefit from the experience obtained when working in the classification of new patterns. The approach for working with ongoing learning presents some advantages: the classifier is more robust because errors or omissions in the TS can be corrected during operation, and the system is capable to continue adapting to a changing environment. The aim is to create an environment to

facilitate the computer system to progressively increase its knowledge and, consequently, to enhance its classification accuracy.

In our proposal, the Nearest Neighbor (NN) rule is employed as the central classifier, mainly because of its flexibility. The NN rule is also an incremental method since it requires relatively little supplementary computer resources to process one additional training pattern.

Because a basic goal is to make the ongoing learning procedure as automatic as possible, it has been designed to work by incorporating new patterns to the TS after they have been labeled by the own system. This way, however, presents two important challenges. Firstly, the danger of performance deterioration by the incorporation to the TS of potentially wrong-labeled new patterns. This is likely to happen, because these new patterns are identified by the computer system instead of by a human expert.

In the present paper we focus on the second challenge: the –possibly unaffordable- increase in the computational cost of the NN rule due to the steady rise in memory and running time requirements of this classification rule. To this end, we propose to employ a pruning or size reduction technique to eliminate unnecessary training patterns. Experimental results will be shown to demonstrate that elimination of redundant and superfluous information, not only allows to reduce the computational burden, but also permits an increase in the classification accuracy of the system.

2 The NN rule and some related techniques

The NN rule is one of the oldest and better-known algorithms for performing non-parametric classification. The entire TS is to be stored in computer memory. To identify a new pattern, its distance is computed to each one of the stored training instances. The new pattern is then assigned to the class represented by its nearest neighboring training pattern.

As any non-parametric classification method, the NN rule is very sensitive to noisy or atypical elements in the TS. The Edition technique of Wilson [1], removing those training patterns that not coincide with the majority of their k nearest neighbors, eliminates noisy as well as close to border instances, leaving smoother decision boundaries. The algorithm has the following steps:

a) For every x_i in TS, find the k (k=3 has been recommended) nearest neighbors of xi among the other prototypes, and the class associated with the larger number of patterns among these k nearest neighbors. Ties would be randomly broken whenever they occur.
b) Edit the TS by deleting those training patterns xi, whose identification label does not agree with the class associated with the largest number of the k nearest neighbors, as determined in the foregoing.

A modification of the Edition technique -the Generalized Edition (GE)- was proposed by Koplowitz and Brown [2] out of concern for the possibility of too many training patterns being removed. This algorithm produces not only elimination of

some instances but also re-identification (label change) of some others. In Generalized Edition, two parameters must be defined: k and k', in such a way that:

$$(k+1)/2 \leq k' \leq k$$

For each prototype x_i in the TS, its k nearest neighbors are searched in the remainder of the TS. If a particular class has at least k' representatives among these k nearest neighbors then x_i is labeled according to that class, independently of its original label. Otherwise, x_i is edited (removed). In short, the technique looks for modifications of the training sample structure through changes of the labels of some training patterns and removal of some others.

These two techniques together, GE applied repeatedly and Edition, perhaps also reiterated, shape a methodology –Depuration- that has proved profitable by correcting the TS and cleaning errors both in the input features and in the class labels [3]. This Depuration methodology is to be regarded as a cleaning process. It removes some suspicious instances from the TS and corrects the class labels of some others prototypes while retaining them. Accordingly, it is designed to cope with all types of incorrectness in the training instances: mislabeled, noisy and atypical or exceptional cases. The methodology involves the application, several times, of the Generalized Edition and, afterwards, the employment of Wilson's Edition, perhaps also reiterated.

Reject options have been implemented in several classification models for reducing the misclassification rate of the system. In these cases, the error-reject relation is very important because of the relative amount of both costs. The best choice depends on the particular pattern recognition application being handled. In our ongoing learning procedure, we have included a reject option for the Nearest Neighbor rule previously presented in [4]. This implementation has the advantage to permit the user to adjust the error-reject relation to suit it according with his/her problem. This reject option works as follows:

a) For every new pattern X to be classified, its two nearest neighbors are searched into the training sample. If these two neighbors are both from the same class, assign X to that class.
b) If the two NN's labels do not coincide, then compare the rate of these two neighbors' distances to X (distance of X to its first nearest neighbor / distance of X to its second nearest neighbor) with a predefined (by the user) threshold value. If smaller, then classify X as member of the class of its first nearest neighbor. Otherwise, reject X.

In this manner, the error-reject relation can be regulated, within certain limits or bounds, shifting conveniently the threshold value. The best value for this threshold can be estimated by working with the TS and with the leave-one-out method for misclassification probability estimation [4].

It is important to note that, as will be explained in the next Section, in our ongoing learning procedure the reject option is not employed to influence in the classification decisions. It is only used to filter the new patterns after they are identified for the system and before they are accepted for their incorporation into the training sample.

3 Procedure with the Ongoing learning capacity

A basic goal is to make this procedure as automatic as possible. Accordingly, the procedure has been designed to work by incorporating new patterns to the TS after they have been labeled by the own system (without human participation). However, it is evident that this working method can be self-defeating. These new training elements would have the class label assigned by the classifier. Therefore, there is the risk to incorporate several wrong labeled patterns to the TS and, consequently, to degrade the system accuracy. The procedure we have designed attempts to overcome this difficulty by employing two complementary automatic tools: a reject option [4] and the Depuration methodology [3]. Reject options have been proposed for reducing the number of classification errors and, recently, to detect new patterns that belong to classes not represented in the TS (e.g., [5, 6]). However, in our ongoing learning procedure the reject option has the goal to restrict incorporation of new patterns to the TS and does not take part in the classification decisions. Although every new pattern is a candidate to become member of the TS after being identified by the system, only those that are not refused by the reject option are accepted for updating the knowledge of the classifier. This is the first resource against the possible contamination of the TS. After incorporating some of the new patterns to the TS, the Depuration is applied as a second filtering step, to amend labels (by removal or re-identification of some patterns) that could have been incorrectly assigned by the system to new patterns not filtered by the reject option. The NN rule is employed as the central classifier.

In summary, the procedure consists of the following steps (see also Fig. 1):

1) Initial TS is stored in memory.
2) Classification phase starts (1-NN rule, not reject option). After identification of a number (for example: 100, as we have implemented in the case of the Landsat dataset, see below) of new patterns, this process is temporarily stopped.
3) The just identified new patterns are assessed for being incorporated to the TS. To minimize the risk of introducing contamination (by wrong labels) into the TS, the reject option filters the candidates to decide which of them are worthy to be joined. Then, Depuration is employed as a second filter to re-label or remove some of the incorporated patterns.
4) If no new pattern remains unidentified, end. Otherwise, the procedure goes to step 2.

The procedure was evaluated through experiments with three real datasets. The first two datasets were taken from the UCI Repository [7] and the third corresponds to several training fields selected from a Landsat-TM image. Datasets were divided into, approximately, 40% for the TS and 60% for an independent set of test patterns (TP) used for control or validation purposes. A summary with the characteristics of these datasets is shown in Table 1.

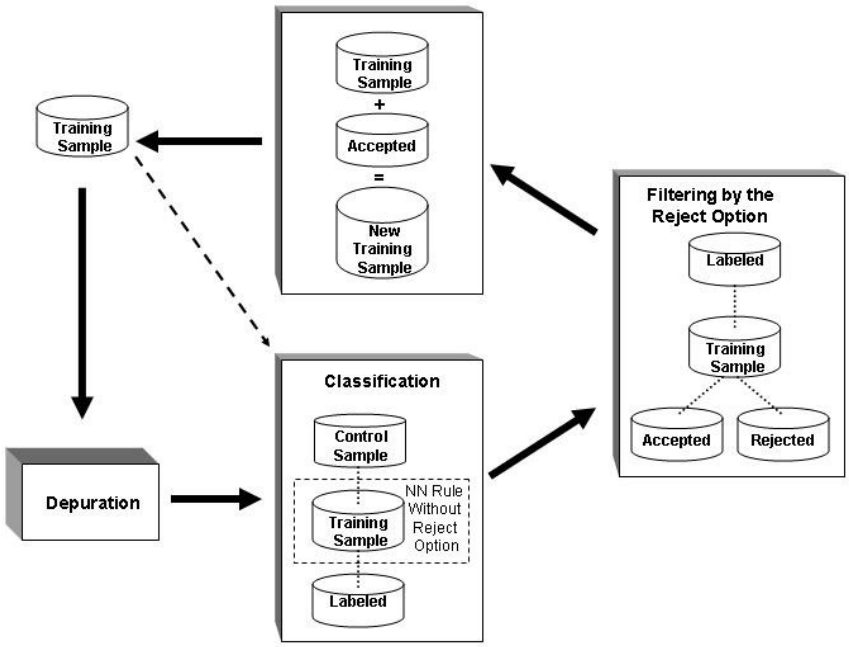

Fig. 1. The ongoing learning procedure

For the experiments, three different partitions were randomly produced for each dataset, all with the same proportion between training and control set, as explained in Table 1. To simulate the sequence required for developing the potentiality of the ongoing learning capacity, the control sets were divided into 10 lots or layers. Each layer contained, approximately, the same proportion of patterns of every class. Since the classification phase stopped temporarily after identification of the members in each of these lots, the system had 10 opportunities to increase its knowledge. An exception was the Landsat dataset whose test set size allowed the creation of 29 layers.

Table 1. Information about the employed real datasets

	#classes	#features	TS size	TP size
Sonar	2	60	83	125
Vowel	11	10	209	325
Landsat	11	4	3033	2993

Results in Table 2 present the accuracy rates averaged over the three replications of each dataset. Confidence levels for the statistical significance (one-tailed t tests) of the difference in accuracy of the ongoing learning procedure with the usual or traditional procedure (when no modification is done on the classification model after the training phase has ended) are also reported. The ongoing learning system was significantly better than the traditional alternative in the three cases.

Table 2. Experimental results with real datasets

	Averaged accuracy (%)		
Dataset	Traditional procedure	Ongoing learning	Statistical significance
Sonar	39.0	49.6	$p<0.001$
Vowel	39.3	52.8	$p<0.001$
Landsat	76.4	86.8	$p<0.001$
Average	51.6	63.1	

4 The issue of the steady increase in the TS size

In the NN rule, the increase in knowledge about the application domain is done through incorporation of new prototypes to the corresponding TS. Accordingly, there is another difficulty when working with the ongoing learning capacity in real practical situations: the possibility for the TS size to become as big as to make it impractical to be handled. Although the Depuration methodology has the property, as a byproduct, of reducing the TS size, this is not achieved in a considerable amount.

Because of the computational burden of the NN rule, many proposals for reducing the TS size have appeared in the literature. In this research line, the idea of Hart [8] has stimulated a sequel of algorithms aimed at eliminating as many training patterns as possible without seriously affecting the accuracy of the classification rule. In the present work, a variant of Hart's idea, the Modified Selective Subset (MSS), has been employed whenever advisable (given the available TS size). The MSS algorithm provides a better approximation to the decision boundaries as they are defined by the whole TS [9].

R_i is defined as the set of all related neighbors to the training pattern x_i, that is, the set of all x_j in TS such that x_j is of the same class of x_i and is closer to x_i than the nearest neighbor of x_i in TS of a different class than x_i,. Then, the MSS is defined as that subset of the TS containing, for every x_i in TS, that element of its R_i that is the nearest to a class different than that of x_i. The MSS algorithm, for the two-class case, consists of the following steps. Only the class 1 is considered, class 2 being afterwards processed in a similar way:

1. Let $x_1, x_2, \ldots, x_{n1}$ be the training patterns of the class 1. *n1* and *n2* are the numbers of training patterns in class 1 and class 2, respectively. These training instances from class 1 have been previously ordered such that $D_1 < D_2 < \ldots < D_{n1}$, where D_i stands for the minimum distance from x_i to the class 2.
2. Place x_1 in MSS. Put *KN*=*n1*-1.
   For *i*=2, …, *n1* {If $d(x_1, x_i) < D_i$ then {K_j=0; *KN*=*KN*-1} else K_i=1;}
3. For *i*=2, …, *n1* begin 1
   Put *IND*=0; If *KN*=0 then exit; else {if K_i =1 then {*Ki* =0; KN=*KN*-1;
   Place x_i in MSS; IND=1;}}
   For *j*=*i*+1, …, *n1* begin 2
   If (K_j =1 AND $d(x_i, x_j) < D_j$ then { K_j=0; *KN*=*KN*-1;
   If *IND*=0 then {K_j=0; IND=1; Place x_i in MSS}}

end 2;
end 1;

The first training pattern, after they have been ordered in step 1, is always selected for the MSS (step 2) since it is the nearest to the other class and, at least, its own related neighbor. K_i indicates whether x_i is already represented in the MSS or not (by represented it is understood that one of its related neighbors is included in the MSS). *KN* is used for the current number of training patterns still remaining to be represented in the MSS. Step 2 is also dedicated to mark all the training patterns of class 1 that are represented in MSS by x_1. Step 3 searches through the rest of the training patterns (in increasing order of their distances from class 2) looking after those that must be transferred to MSS either because they have not been represented by any previous instance or because they are related neighbors of a posterior not yet represented training pattern. *IND* is a flag that prevents duplication of an instance in the subset.

An efficient algorithmic representation of the MSS method that puts forward its worst case quadratic complexity is depicted in Fig. 2 below.

Sort the prototypes $\{x_j\}_{j=1}^{n}$ according to increasing values of D_j.

For $i = 1, \ldots, n$ do // x_i is the next best

add ← FALSE
For $j = i, \ldots, n$ do // check others including x_i
if $x_j \in S \wedge d(x_i, x_j) < D_j$ then // x_j fulfills now the property with x_i
$S \leftarrow S - \{x_j\}$
add ← TRUE
endfor

if *add* then $MSS \leftarrow MSS \cup \{x_i\}$ //add if S has changed

if $S = \varnothing$ then return MSS //every x has a relative
endfor

Fig. 2. Efficient implementation of the Modified Selective algorithm

To explore the convenience of using a pruning algorithm for controlling the TS size, the same experiments that we have reported in the previous Section have been repeated. But now, the MSS reduction technique has been applied whenever the current TS reached a size greater than a third of the initial amount of training patterns. Results are in Table 3.

These results indicate that MSS has not only reduced the TS size (for instance, in Landsat, the ongoing learning procedure yielded a TS seven times bigger than the initial one, but with employment of MSS the final size was only 60% of the original). This pruning algorithm has also improved the classification accuracy, except in Vowel dataset (however, accuracy with MSS in Vowel is still better than in the traditional alternative). Moreover, MSS has showed itself as a resource to cope with new patterns not correctly filtered by the reject option. MSS has removed –more than

Depuration- many of these patterns that had been incorporated to the TS with wrong labels and has helped to decrease the amount of these patterns wrongly accepted. Periodical employment of the MSS has produced also a smaller quantity in the incorporation of new patterns with wrong identification label. Accordingly, MSS has produced much cleaner TSs.

Table 3. Experimental results when applying MSS for controlling the TS size

	Sonar		Vowel		Landsat	
	With MSS	No MSS	With MSS	No MSS	With MSS	No MSS
Added with wrong label	17.0	51.7	86.7	109.7	207.0	318.7
Removed by Depur.	0.3	13.7	24.0	26.3	37.0	56.0
Removed by MSS	15.3	--	28.7	--	118.0	--
Remaining at the end	1.3	38.0	34.0	83.3	52.0	262.7
Increase in size (%)	-84.6	428.7	-20.2	158.5	-39.9	729.2
Accuracy (%)	54.1	49.6	45.6	52.8	89.0	86.8

5 Related work and concluding comments

In the neural network and machine learning communities, online (or lifelong) learning methods have received some attention. One of the first attempts is due to Pratt [10]. She is concerned with the transfer of the knowledge stored in a previously trained multilayer perceptron model to a new neural network. Her interest is to reduce the long time required by the back-propagation training procedure and is motivated by those situations when new training patterns become available after an initial classification system has been already set to work.

Thrun and Sullivan [11] deal with the problem of acquiring and re-using domain-specific knowledge across multiple learning tasks and they propose an algorithm to cluster learning tasks into classes of mutually related tasks. Their interest lies in the improvement in the learning ability of a recognition method when it is applied to a sequence of learning tasks. A detailed review of the strategies to transfer knowledge can be found in Thrun [12],[13].

Bruzzone and Fernandez-Prieto [14] present an incremental learning technique for a classifier based upon a Radial Basis Function neural network. Their purpose is to allow the acquisition of new knowledge whenever a new training set becomes available, while preserving the knowledge acquired on previous training sets. It is an important property in remote sensing applications. In these applications it would be very useful to have a classifier that, after being trained on data related to a specific image, could be able to attain acceptable classification accuracy when employed on different images (provided that the land-cover classes in all the images are the same).

Unlike our approach, all these proposals rely on the periodical availability of new training sets. That is, all these systems are always working with training patterns identified by human experts. Then, these training patterns are assumed to bear correct class labels and to not represent possibility of knowledge contaminated or mistaken.

Dasarathy [15] proposed a decision system with a design very related to ours. He was also concerned with the robustness of the system through varying environments

and with the problem of unrepresentative pre-training. The latter is what he called "partially exposed environments". Consequently, Dasarathy presented an on-line adaptive learning system with two capabilities:

a) to progressively improve the classification of objects belonging to the known classes
b) to detect the objects not belonging to the currently known classes

However, Dasarathy's system requires the steady participation of a human expert to be in charge of the evaluation of the labels assigned by the system to new patterns and to decide which of them are to be incorporated to the training sample. As he himself (page 1271) pointed out: "in real-world operational phase, such operator supervision may be unavailable".

We avoid this bottleneck by incorporating to our procedure the necessary tools to let the system to decide alone which pieces of new knowledge are trustworthy enough to be accepted. Of course, these selections are not to be always correct (it is arguable whether the training patterns identifications assigned by human experts are one hundred percent correct, at least in several domain applications). But the experimental results above indicate that the potential damage that could be produced by distorted information is more than compensated by the enrichment allowed by new useful knowledge and the better understanding of the characteristic of the application at hand.

The importance of using reduction algorithms in ongoing learning systems has also been analyzed in the present paper. When working with the ongoing learning capability in real-world applications, the TS can grow too much and, consequently, problems related to storage and classification time can make such a system useless. The experiments here reported have shown the effectiveness of employing the Modified Selective subset algorithm within an ongoing learning system, not only to appropriately reduce the TS size but also to increase classification accuracy. The results presented here concerning the design of an ongoing learning system, should be viewed as a first step toward a more complete understanding of how to exploit the use of different approaches in a system with such capability.

Acknowledgments

This work has been partially supported by grants 44225 from the Mexican SEP-Conacyt, 744.99P from the Mexican Cosnet, TIC2003-08496 from the Spanish CICYT, SAB2003-0106 from the Spanish MECD and P1-1B2002-07 from the Fundació Caixa Castelló – Bancaixa.

References

1. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man and Cybernetics **2** (1972), 408-421

2. Koplowitz, J., Brown, T.A.: On the relation of performance to editing in nearest neighbor rules. In: Proceedings of the 4th International Conference on Pattern Recognition, Japan (1978)
3. Barandela, R., Gasca, E.: Decontamination of training samples for supervised pattern recognition methods. In: F. Ferri et al. (eds.), Advances in Pattern Recognition, Lecture Notes in Computer Science, vol. 1876. Springer-Verlag, Berlin, Heidelberg, New York (2000)
4. Barandela, R.: The Nearest Neighbor rule, an empirical study of its methodological aspects, Doctoral Thesis, Institute of Cybernetics, Berlin (1987)
5. Tax, D.M., Duin, R.P.: Outlier detection using classifier instability. In: A. Amin et al. (eds.), Advances In Pattern Recognition, Lecture Notes in Computer Science, vol. 1451, Springer-Verlag, Berlin, Heidelberg, New York (1998)
6. Barandela, R., Ferri, F.J., Najera, T.: Some experiments in supervised pattern recognition with incomplete training samples, Lecture Notes in Computer Science, vol. 2396, 518-527, 2002.
7. Merz, C.J., Murphy, P.M.: UCI Repository of Machine Learning Databases, University of California at Irvine, Department of Information and Computer Science (1996)
8. Hart, P.E.: The Condensed Nearest Neighbor rule. IEEE Transactions on Information Theory **6(4)** (1968) 515-516
9. Barandela, R., Cortes, N., Palacios, A.: The Nearest Neighbor rule and the reduction of the training sample size. In: Proceedings of the IX Symposium of the Spanish Society for Pattern Recognition, Castellón, Spain (2001)
10. Pratt, L.Y.: Transferring Previously Learned Back-Propagation Neural Networks to New Learning Tasks, Ph.D. Thesis, Rutgers University, Department of Computer Science (1993)
11. Thrun, S., O'Sullivan, J.: Discovering Structure in Multiple Learning Tasks: The TC Algorithm. In: L. Saitta (ed.), Proceedings of the Thirteenth International Conference on Machine Learning, Morgan Kaufmann, San Mateo, CA (1996)
12. Thrun, S.: Is learning the n-th thing any easier than learning the first? In: Advances in Neural Information Processing Systems 8, MIT Press (1996)
13. Thrun, S.: Explanation-Based Neural Network Learning: A Lifelong Learning Approach, Kluwer Academic Publishers, Boston (1998)
14. Bruzzone, L., Fernandez-Prieto, D.: An incremental-learning neural network for the classification of remote sensing images, Pattern Recognition Letters **20** (1999) 1241-1248
15. Dasarathy, B.V.: Adaptive decision systems with extended learning for deployment in partially exposed environments, Optical Engineering **34** (1995) 1269-1280