Ongoing Learning for Supervised Pattern Recognition.

RICARDO BARANDELA[1]
MARIELA JUÁREZ[1]
[1]Lab for Pattern Recognition
Instituto Tecnológico de Toluca
Ave. Tecnológico s/n, 52140 Metepec
Edomex, México
{rbarandela, mjuarez}@hotmail.com

Abstract. This paper presents a procedure to implement an automatic system for supervised pattern recognition with an ongoing learning capability. The purpose is to continuously increase the knowledge of the system and, accordingly, to enhance its performance in classification tasks. The Nearest Neighbor rule is employed as the central classifier and several techniques are added to cope with the increase in computational load and with the peril of incorporating noisy data to the training sample. Experimental results confirm the improvement in classification accuracy.

1 Introduction

Learning algorithms and pattern recognition methods have been 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 training sample (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 presents 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 training set can be corrected during operation, and the system is capable to continue adapting to a changing environment. 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 procedure as automatic as possible, it is 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 –possibly unaffordable- increase in the computational cost of the NN rule due to the steady rise in memory and running time requirements. Secondly, the danger of performance deterioration by the incorporation to the TS of potentially wrong-labeled new patterns. The latter because these new patterns are identified by the computer system instead of by a human expert.

The procedure proposed here attempts to handle these difficulties with the help of some techniques that will be explained hereafter. The aim is to create an environment to facilitate the computer system to progressively increase its knowledge and, consequently, to enhance its classification accuracy. Experimental results with artificial and real datasets are presented.

1530-1834/01 $10.00 © 2001 IEEE

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. This definition implies a main drawback: large memory -to store the whole TS (that one wishes to be as big as possible)- and response time requirements. This computational burden has been considerably cut down by developing suitable data structures and associated non-exhaustive search algorithms or by reducing the TS size. The idea of Hart [7] in this latter research line 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, because it provides a better approximation to the decision boundaries as they are defined by the whole TS (Barandela et al. [3]).

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 [15], removing those training patterns that not coincide with the majority of their k nearest neighbors, eliminates noisy as well as close border instances, leaving smoother decision boundaries. The algorithm has the following steps:

1. For every x_i in TS, find the k (k=3 has been recommended) nearest neighbors of x_i 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.
2. Edit the TS by deleting those training patterns x_i, 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 [8] 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 (Barandela and Gasca, [2]). This Depuration methodology is to be regarded as a cleaning process. It removes some suspicious instances from the training sample 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 being handled. In our procedure, we have included a reject option for the Nearest Neighbor rule (Barandela, [1]). 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 training sample and with the leave-one-out method for misclassification probability estimation (Hand, [6]).

It is important to note that, as will be explained in the next Section, in our 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

As already mentioned, the system proposed here is based on the widely popular nearest neighbor rule, because of the flexibility and other properties of this classifier, and is designed with an on-line learning capability for progressively increasing the knowledge and the classification accuracy of the system.

Since we intend to develop a procedure that works as automatically as possible (that is, without human participation), it is necessary to cope with two potential dangers. The system is to enrich its knowledge by incorporating to the training sample those patterns identified by the own system during the classification phase. However, it is evident that this 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 present procedure attempts to overcome this difficulty by employing two complementary automatic tools: a reject option and the Depuration methodology, both outlined in the preceding Section. 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., Tax and Duin [11]). However, in our procedure the reject option has the goal to restrict incorporation of new patterns to TS and does not take part in the classification decisions. That is, 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 training sample. After incorporation of some of the new patterns to the TS, the Depuration is applied as a second filtering step, to amend (by removal or re-identification) labels that could have been incorrectly assigned by the system to those new patterns that were not detected by the reject option.

The other problem is related with the possibility for the training sample size to become so big as to make it impractical to be handled. As for this concern, it is to remark that the Depuration methodology has the property, as a byproduct, of reducing the training sample size (although in a not considerable amount). The MSS algorithm is employed when, even after the Depuration application, the TS size is still a too high number.

In summary, the procedure consists of the following steps (see also Figure 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. When the TS size is too high, reduction algorithm (MSS) is applied.
4) If no new pattern remains unidentified, end. Otherwise, the procedure goes to step 2.

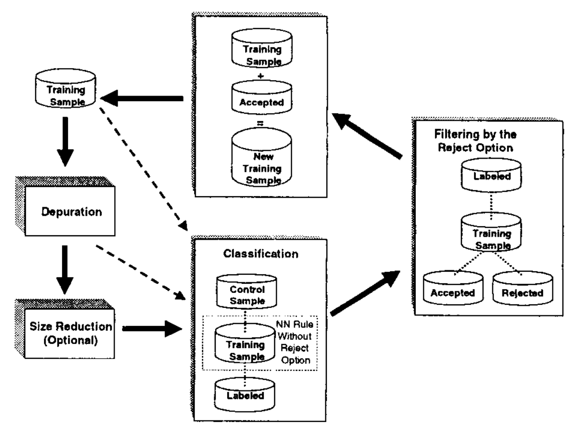

Figure 1. The ongoing learning procedure.

4 Experimental Results

To gain some initial insight, an assay was carried out with simulated data (a two-class problem with bivariate data pseudo-randomly generated according to Gaussian distributions). Three different sizes for the training sample were assessed. The ongoing learning capability allowed lowering the classification error in more than 20% when compared to the obtained by the traditional alternative. See Table 1.

Variant	Training sample size		
	20	50	100
1	15.6	15.0	15.4
2	13.1	11.7	11.8

Table 1.Results (% of misclassifications) with simulated data. The control set consisted of 500 patterns.

All the experimental comparison included two variants:

Variant 1: usual or traditional behavior of the classifier, identifying all the test patterns in a single batch and without knowledge increase during the second phase.

Variant 2: the proposed procedure with ongoing learning. The test set was partitioned in several groups. After identification of the patterns in each group, operational phase stopped temporarily to incorporate to the TS those new patterns not filtered by the reject option, to depurate the resulting TS and to reduce its size when necessary.

For a more precise evaluation, the procedure was applied afterwards to five real datasets. The first four of them were taken from the UCI Repository [9] and the fifth corresponds to several training fields selected from a Landsat-TM image. Datasets were divided in, approximately, 40% for the training sample (TS) and 60% for an independent set (CS) used for control or validation purposes. Characteristics of these datasets are in Table 2.

Dataset	#classes	#features	TS size	CS size
Liver	2	6	138	207
Ionosphere	2	34	140	210
Sonar	2	60	83	125
Vowel	11	10	209	325
Landsat	11	4	3033	2993

Table 2. Information about the employed real datasets.

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 the Table 2. 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 control set size allowed the creation of 29 layers.

Results in Table 3 present the misclassification rates – both alternatives- 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 variant are also reported. The ongoing learning system was significantly better than the traditional alternative in four of the five cases and there was not significant difference in the Ionosphere application.

	Averaged misclassification (%)		
Dataset	Variant 1	Variant 2	Stat. significance
Liver	40.2	38.2	$p<0.005$
Ionosphere	25.4	26.0	no significance
Sonar	61.0	44.8	$p<0.001$
Vowel	60.7	47.2	$p<0.001$
Landsat	23.6	13.2	$p<0.001$
Average	42.2	33.9	

Table 3. Experimental results with real datasets.

Landsat dataset is the one with the highest size in the control set. Besides, this is the dataset from which more ancillary information was available. For these reasons, we have selected it for a more detailed analysis of some issues of the procedure proposed here. Table 4 presents a dynamical description of the training sample structure as it is being gradually modified by the incorporation of new examples. Columns 2 and 3 indicate the number of patterns that were accepted for incorporation by the reject option, the majority of them with the labels assigned by the domain experts (1762 with the same label against only 60 with different labels). Column 7 shows the number of patterns that remain in the training sample after the application of the Depuration methodology and the composition of this resulting training sample is reflected through columns 4-6. Columns 8-11 present the same information concerning the results of the posterior employment of the Modified Selective Subset algorithm. See also Figures 2 and 3 for graphical depictions.

Layer number	Incorporated patterns		Structure After Depuration				Structure After Modified Selective			
	Same label	Different label	Originals	Incorporated Same label	Incorporated Different label	Total	Originals	Incorporated Same label	Incorporated Different label	Total
0	0	0	383	0	0	383	383	0	0	383
1	67	7	368	67	7	442	145	6	2	153
2	77	4	140	83	6	229	87	5	4	96
3	84	3	83	88	7	178	64	13	3	80
4	74	9	55	86	9	150	43	6	6	55
5	59	10	43	65	16	124	40	6	13	59
6	56	0	40	62	13	115	37	10	12	59
7	61	1	37	71	13	121	36	15	12	63
8	62	0	36	77	12	125	36	16	10	62
9	57	0	36	73	10	119	36	17	9	62
10	62	2	36	79	11	126	35	20	11	66
11	63	2	35	83	13	131	35	23	9	67
12	59	3	35	82	12	129	34	23	12	69
13	68	0	34	91	12	137	34	24	12	70
14	57	1	34	81	13	128	33	27	13	73
15	51	3	33	78	16	127	32	31	16	79
16	39	0	32	70	16	118	32	33	15	80
17	37	1	32	70	16	118	30	32	15	77
18	51	0	30	83	15	128	30	32	15	77
19	46	0	30	78	15	123	30	31	15	76
20	38	2	30	69	17	116	30	30	16	76
21	46	2	30	76	18	124	30	30	18	78
22	54	0	30	84	18	132	30	31	17	78
23	65	1	30	96	18	144	30	31	18	79
24	75	0	30	106	18	154	30	32	17	79
25	70	0	30	102	17	149	30	34	17	81
26	63	7	30	97	24	151	29	36	19	84
27	70	2	29	106	21	156	29	36	21	86
28	79	0	29	115	21	165	27	37	21	85
29	72	0	27	109	21	157	27	39	20	86
Total	1762	60								

Table 4. Landsat data. Steady updating of the training sample in one of the experimental runs with the ongoing learning procedure

It can be observed in Table 4 that the combination Depuration-MSS has produced an additional filtering and at the end of the procedure, in the training sample remained only 20 of those 60 new patterns that were acquired with a label different from the original they had in the control set. That is, two thirds of them were gradually removed. The available ancillary information (spatial distribution of the patterns) allowed a more deeper analysis of this situation and we were able to detect that 15 of these 20 supposedly wrong-labeled incorporated patterns corresponded to pixels located at the edges of the selected test fields. Because of the way these training sites are chosen, this fact implies the possibility of mixed elements or pixels that combines the spectral characteristics of two adjacent classes. Therefore, these patterns could have received a wrong identification when originally collecting the CS. On the contrary, 967 pixels that were correctly identified by the NN rule were not accepted by the reject option.

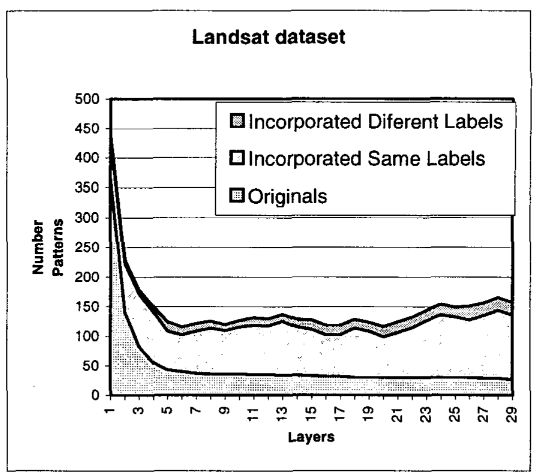

Figure 2. Structure after depuration

		Employing Modified Selective after new incorporations				Without applying Modified Selective			
Layer number	No. of Patterns	Training sample size	% Increment./ Decrement	Misclassifications Number	Misclassifications %	Training sample size	% Increment	Misclassifications Number	Misclassifications %
1	100	383		15	15.00	383		15	15.00
2	100	153	-60.05	8	8.00	442	15.40	7	7.00
3	100	96	-37.25	3	3.00	539	21.95	4	4.00
4	100	80	-16.67	11	11.00	637	18.18	7	7.00
5	101	55	-31.25	17	16.83	736	15.54	13	12.87
6	100	59	7.27	4	4.00	834	13.32	12	12.00
7	102	59	0.00	3	2.94	932	11.75	4	3.92
8	101	63	6.78	1	0.99	1032	10.73	2	1.98
9	101	62	-1.59	0	0.00	1132	9.69	3	2.97
10	102	62	0.00	8	7.84	1232	8.83	16	15.69
11	103	66	6.45	9	8.74	1324	7.47	15	14.56
12	100	67	1.52	13	13.00	1422	7.40	5	5.00
13	101	69	2.99	5	4.95	1517	6.68	8	7.92
14	102	70	1.45	7	6.86	1616	6.53	3	2.94
15	101	73	4.29	8	7.92	1713	6.00	4	3.96
16	101	79	8.22	13	12.87	1812	5.78	16	15.84
17	100	80	1.27	8	8.00	1912	5.52	9	9.00
18	101	77	-3.75	2	1.98	2009	5.07	11	10.89
19	100	77	0.00	8	8.00	2096	4.33	3	3.00
20	102	76	-1.30	13	12.75	2194	4.68	3	2.94
21	102	76	0.00	12	11.76	2290	4.38	7	6.86
22	102	78	2.63	6	5.88	2385	4.15	5	4.90
23	103	78	0.00	1	0.97	2485	4.19	4	3.88
24	101	79	1.28	1	0.99	2586	4.06	4	3.96
25	102	79	0.00	3	2.94	2684	3.79	4	3.92
26	101	81	2.53	8	7.92	2783	3.69	5	4.95
27	102	84	3.70	4	3.92	2879	3.45	3	2.94
28	100	86	2.38	3	3.00	2978	3.44	3	3.00
29	102	85	-1.16	1	0.98	3077	3.32	1	0.98
Total	**2933**		**-77.81**	**195**	**6.65**		**703.39**	**196**	**6.68**

Table 5. Landsat data. Analysis of the impact of the Modified Selective Algorithm on the training sample size and on the classification accuracy.

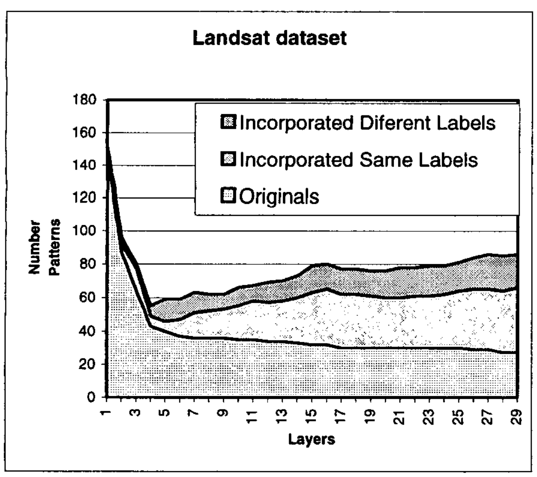

Figure 3. Structure after Modified Selective.

As the lecture of the last row in Table 4 might produce concern about the reduced size of the training sample as a consequence of the repeated application of the Modified Selective Subset algorithm, an analysis of the impact of this pruning method is shown in Table 5. Column 4 indicates –in percentage- the modifications caused in the training sample size in each step of the ongoing learning procedure. Increments are produced when the number of incorporated patterns is greater than the number of patterns eliminated by the Modified Selective method. Obviously, in Column 8 only increments appear since this corresponds to the variant without the employment of the pruning algorithm. Comparison of Columns 5-6 with Columns 9-10 leads to an important remark: classification errors are the same with and without the Modified Selective application (in fact, there is one misclassification less when employing the Modified Selective). On the other hand, the reduction in the computational burden of the classification rule is huge. Modified Selective permits to work with less than 23% of the initial training sample size (Column4, last row). On the contrary, when the pruning algorithm is not employed, the last training sample size is more than seven times the initial one.

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 Multi-Layer 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 [14] 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 [4] 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. That means that these training patterns are assumed to bear correct class labels and to not represent possibility of knowledge contaminated or mistaken.

Dasarathy [5] 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 pretraining. 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.

As it can be observed in Table 3, the best results (classification accuracy in comparison with the traditional procedure) in our system are obtained with those datasets with an important number of patterns in their control sets. This behavior is consistent with the basic goal of the ongoing learning capability: to steadily improve the knowledge and to enhance the performance of the system as it is employed for the classification of new patterns. That is, the greater the practical experience acquired the more reliable the performance of the classifier.

We intend to do further work to tune some of the parameters of the procedure. One of these parameters is the threshold value for the reject option. In all the experiments above reported we have employed 0.25 as the threshold value. Preliminary experiments with a value of 0.1 did not offer good results. However, this is an issue that deserves additional study. Euclidean distance metric was employed until now. Although, because of the nature of the data we have used, we do not expect important changes, it could be interesting to explore with other distance functions. Also some research for adding to the system the capacity to detect new classes and to cope with the shortcoming of unbalanced training sets is to be done in the next future.

Acknowledgements

This work has been partially supported by Grant 32016-A from Conacyt and by Grant 744.99P from Cosnet, Mexico.

References

[1] R. Barandela. *The Nearest Neighbor rule: an empirical study of its methodological aspects*. Ph.D. thesis, Institute of Cybernetics, Berlin (1987).

[2] R. Barandela and E. Gasca. Decontamination of training samples for supervised pattern recognition methods. In *Advances in Pattern Recognition, Lecture Notes in Computer Science*, vol. 1876, F. Ferri et al. (eds.), Springer, Berlin (2000), 621-630.

[3] R. Barandela, N. Cortes and A. Palacios. The Nearest Neighbor rule and the reduction of the training sample size. In: Proceedings of the 9th Spanish Symposium on Pattern Recognition and Image Analysis I (2001), 103-108.

[4] L. Bruzzone and D. Fernandez-Prieto. An incremental-learning neural network for the classification of remote sensing images. *Pattern Recognition Letters*, 20 (1999), 1241-1248.

[5] B. V. Dasarathy. Adaptive decision systems with extended learning for deployment in partially exposed environments. *Optical Engineering*, 34 (1995), 1269-1280.

[6] D. J. Hand. Construction and Assessment of Classification rules. John Wiley and Sons (1997).

[7] P. E. Hart. The condensed nearest neighbor classification rule. *IEEE Transactions on Information Theory*, 14 (1968), 515-516.

[8] J. Koplowitz and T. A. Brown. On the relation of performance to editing in nearest neighbor rules. In *Proceedings of the 4th International Joint Conference on Pattern Recognition*, Japan (1978).

[9] C. J. Merz and P.M. Murphy. *UCI Repository of Machine Learning Databases*. University of California at Irvine, Department of Information and Computer Science. URL :http://www.cs.uci.edu/~mlearn/MLRepository.html (1996).

[10] L. Y. Pratt. *Transferring Previously Learned Back-Propagation Neural Networks to New Learning Tasks.* Ph.D. thesis, Rutgers University, Department of Computer Science, New Brunswick, NJ (1993).

[11] D. M. Tax and R. P. Duin. Outlier detection using classifier instability. In Advances in Pattern Recognition, Lecture Notes in Computer Science, vol. 1451, A. Amin et al. (eds.),Springer, Berlin (1998).

[12] S. Thrun. Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems 8*, MIT Press (1996).

[13] S. Thrun. *Explanation-Based Neural Network Learning: A Lifelong Learning Approach.* Kluwer Academic Publishers, Boston MA (1998).

[14] S. Thrun and J. O'Sullivan. Discovering Structure in Multiple Learning Tasks: The TC Algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*. L. Saitta (ed.), Morgan Kaufmann, San Mateo CA (1996).

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