TWO MODIFICATIONS OF THE DECONTAMINATION METHODOLOGY

R. Barandela[1,2], J. S. Sánchez[3], E. Rangel[1]
[1] Lab for Pattern Recognition, Inst. Tecnológico de Toluca, 52140 Metepec, México
[2] Instituto de Geografía Tropical, La Habana, Cuba
[3] Dept. Llenguatges i Sistemes Informàtics, U. Jaume I, 12071 Castelló, Spain
{rbarandela, erangel_lugo}@hotmail.com; sanchez@uji.es

1 Introduction

Learning algorithms have been sorted into two broad groups: supervised and unsupervised, whether training data is available or not. Supervised classifier 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. In several practical applications, however, class identification of the prototypes is a difficult and very costly task. As a consequence, some incorrectly labeled instances may be present in the TS, leading to situations lying in between supervised and unsupervised methods, or as they have been called: imperfectly supervised environments [1]. Examples have been reported in drawing of mineral deposit maps and, particularly, in the interpretation of remotely sensed data. There is another source of distortion in the training data: instances with errors in some attribute values and those that are atypical or exceptional cases.

Generalization accuracy of the learning algorithm may be degraded by the presence of incorrectness or imperfections in the training data. Particularly sensitive to these deficiencies are nonparametric classifiers whose training is not based upon any assumption about probability density functions. That explains the emphasis given in the machine learning and pattern recognition communities to the evaluation of procedures used to collect and to clean the TS, critical aspects for the effective automation of discrimination tasks. In previous works a methodology for correcting and decontaminating a TS has been presented [2,3]. This methodology, the Decontamination procedure, can be regarded as a cleaning process removing some elements of the TS and correcting the labels of several others while retaining them. It has been conceived specifically for the Nearest Neighbor (NN) rule although exploratory results with a Multi-Layer Perceptron point to a broader applicability. Experimental results with both simulated and real datasets have shown that the Decontamination methodology allows coping with all types of imperfections (mislabeled, noisy, atypical or exceptional) in the set of training instances, improving the classifier's performance and lowering its computational cost.

In several real domains, some available information can help to refine the knowledge about the kind of contamination potentially present in the TS. For instance, in the interpretation of remote sensing data, it is very likely that some prototypes labeled as member of those classes with relative small spatial size had received a wrong identification. This is true, particularly for those pixels located on the border of these small training fields. The reason behind this fact lies in the difficulty to isolate these small-area fields when marking them with the computer mouse. In the present paper, we introduce a modification of the decontamination procedure to cope more effectively with these particular situations. Extensions of these ideas to work with the surrounding neighborhood concept are also presented.

2. Incorrectness in the Training Data and Related Works

As above mentioned, approaches to supervised learning methods require the fulfillment of two basic assumptions concerning the TS, in order to guarantee accurate identification of new cases. Practical experience has shown that in many real applications one or both of these hypothesis do not entirely hold. Violations of these assumptions may strongly degrade classification accuracy, particularly when employing nonparametric methods. In conformity with this view, the number of papers and proposals for handling this subject has significantly increased in the last years.

Outlier data is a concept that has been considered in Statistics for some time. It can be defined as a case that differs significantly from the rest of the instances in its class or group. Difficulties for isolating these cases and to cope with them have long been discussed. Now the term has come to the fore also in the Machine Learning, Pattern Recognition and Data Mining areas. Reports about the effect of these "noisy" or atypical patterns when included in the training sample and how to counteract it have been published (e.g., [4,5]). Gopalakrishnan et al. [6] are concerned by the long training time required usually by some neural networks models. They propose the use of a clustering technique for identifying noisy training cases that slow down the learning phase of these models. John [7] addresses the identification of outliers in categorical data and promotes an iterative procedure to clean the training data. He works with the C4.5 tree induction algorithm and removes those training instances linked to the nodes eliminated during the pruning phase (that is, those cases whose removal from the TS will most increase the model's estimate of its own performance). Afterwards, the model is built again up considering only the cleaned training data. His purpose is to obtain a smaller and more accurate decision tree. Even for unsupervised methods outlier data has been the concern of several researchers (e.g., [8]).

In the interpretation of remotely sensed data, the difficulties introduced into the classification process by those prototypes that represent more than one class (e.g., those allocated on the border between classes) or by unknown patterns that belong to a class not considered when collecting the training sample have been discussed [9,10]. Mather [11] refers to atypical elements in the TS that may belong to another class or may be hybrid or mixed elements.

Brodley and Friedl [12] employ a combination of classifiers to filter the training patterns looking for detection and elimination of wrong-labeled training cases prior to applying the chosen learning algorithm. They are very concerned with the difficulty of correctly separating instances that are merely exceptions to the general rule from noisy prototypes that are incorrectly labeled. They considered that the elimination of exceptional training cases must be avoided and mention some previous proposals aimed at discerning among true exceptions and noisy data. Some of these proposals require the participation of a human expert.

The terms outlier, noisy and atypical are generally employed to cover a broad range of circumstances That reflects some confusion among dissimilar situations and a lack of a rigorous and unified concept of outlier data. In general, there are three of these potential situations:

1. Noisy data that are usually produced by errors (measuring, recording, etc), an unfortunate property of many real-world databases.
2. New unidentified patterns appearing in the classification phase and that do not belong to any of the classes represented in the TS (partially exposed environments, [13,14]). These cases are usually handled by a reject option [15-17].
3. Some authors employ the term outlier for denoting mislabeled instances in the TS, what constitutes the main focus of the present work.

Key differences of the just mentioned works with the procedures presented in this paper are:

- i) The previous proposals do not consider correcting labels of some imperfectly identified training data. Most of them try to solve the problem only through elimination of those instances that appear to be “doubtful”. Another line [18] rests upon the measurement of a confidence level or “typicality” for each training instance. Then, this information is employed when classifying an unknown pattern with a weighted *k*-NN rule.
- ii) In some of these previous publications, real data is used for demonstration purposes. But then, the labels of some training patterns are intentionally modified to simulate an imperfectly supervised situation (e.g., [12,19]). Since there is not control or knowledge of the original contamination level in those real datasets, it is possible, by working in that way, to produce a dataset cleaner than the original one.

3. The NN Rule and the Decontamination methodology

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

An important drawback of the NN rule is its sensitivity to the presence of erroneous or noisy prototypes in the training set. The NN rule shares this characteristic with other non-parametric methods such as the Neural Networks models. Imperfections in the TS are the usual situation in many real-world applications. It is widely accepted that the presence of these imperfections may produce a serious decrease in classification accuracy. Editing techniques are mainly aimed at improving the performance of the NN rule by discarding outliers and cleaning the overlap among classes. As a by-product, they also obtain a decrease in the TS size and, consequently, a reduction of the computational burden of the classification method. The first work of editing corresponds to Wilson [20] and several others have followed.

2.1 Wilson’s Editing procedure

This technique consists of applying the *k*-NN ($k > 1$) classifier to estimate the class label of every prototype in the training set and discard those instances whose class label does not agree with the class associated to the majority of the *k* neighbors. The procedure is:

1. Let S = X (X is the original training set and S will be the edited TS)
2. For each x_i in X do:
   a) Find the k nearest neighbors of x_i in X \ $\{x_i\}$
   b) Discard x_i from S if its label disagrees with the class associated with the largest number of the k neighbors.

2.2 Generalized Editing (GE: [21])

This is a modification of the Wilson's algorithm, proposed out of concern with the possibility of too many prototypes being removed from the TS. This approach consists of removing some suspicious prototypes and changing the class labels of some other instances. Accordingly, it can be regarded as a technique for modifying the structure of the training sample (through re-labeling of some training instances) and not only for eliminating atypical instances. In GE, two parameters have to be defined: k and k' in such a way that

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

This editing algorithm can be written as follows:

1. Let S = X.
2. For each x_i in X do:
   a) Find the k nearest neighbors of x_i in X \ $\{x_i\}$.
   b) If a class has at least k' representatives among those k neighbors, then identify x_i according to that class (independently of its original class label). Otherwise, discard x_i from S.

2.3 The Decontamination methodology in brief

The Decontamination methodology involves several applications of the GE technique followed by the employment, also repeatedly, of the Wilson's Editing algorithm. Repetition in the application of each one of these techniques stops if one of the following criteria is fulfilled:

a) Stability in the structure of the training sample has been reached (no more removals and no more re-labeling).
b) Estimate of the misclassification rate (leave-one-out method) has begun to increase.
c) One class has resulted emptied (all its representatives in the training sample have been removed or transferred to another class) or has resulted with a very small size/dimensionality rate (less than five training instances for each considered feature in our implementation).

When this size/dimensionality rate is not adequate, feature selection must be done to allow further decontamination of the training sample.

4. Proposals to modify the Decontamination procedure

In the present paper, we present experimental results to assess the benefits of some modifications of the already explained Decontamination methodology. In the first place, the variant that is called Restricted Decontamination.

This restricted procedure has been designed to handle situations such as those explained in the introductory section. Information about the particular application area could imply existence of contamination in only some of the classes. The source for this information could be given by some characteristics of the process applied to collect the TS or by the intrinsic nature of the real problem to be handled. In these cases, it seems to be convenient to employ the Decontamination methodology only on those classes suspected of containing wrong identified prototypes. That is, change of label or removal from the TS will affect only to those prototypes representing the classes presumed as being contaminated.

At the same time, we wanted to explore the extensions of these ideas when employing a different concept of neighborhood. Neighbors of the prototypes are searched both in the GE technique and in the Wilson's Edition. In the scheme explained in the precedent section, these neighbors are searched by merely employing the usual Euclidean distance. In the experiments to be reported hereafter, we have also obtained results when these neighbors are searched by means of the surrounding neighborhood concept.

5. Experimental results

The proposals introduced in the previous section have been evaluated through experiments with two sets of artificial or synthetic data. One of these sets, the Gaussian data, corresponds to simulated patterns from two bivariate Gaussian populations with different mean vectors ($m_1 = (0.0, 0.0)^t$ and $m_2 = (2.58, 0.0)^t$, respectively) and an equal covariance (identity) matrix [22]. This is a model with a not negligible amount of overlap between the two classes. The other set of artificial data, the Uniform set, corresponds to data pseudo-randomly generated from the model employed in the paper of Hart [23] and several others. In every case, ten replications of each experiment were carried out. Each of the ten training samples consisted of 200 prototypes. One independent test sample (used for validation purposes) contained 500 elements. In each dataset there has been always a half of the prototypes from each class.

For evaluating the already explained modifications of the Decontamination methodology, class 1 of each dataset was intentionally contaminated. That is, prototypes actually belonging to class 2 have received a class 1 label. Six levels of noise or contamination were produced: 0%, 10%, 20%, 30%, 40% and 50%. After the contamination, the TS was processed with the different variants of the Decontamination procedure. Evaluation was done by recording the misclassifications produced when classifying the test set, with the contaminated TS and with the processed TS. Of course, this test set was never contaminated. Classification of the NN rule when employing the original TS is also reported as baseline.

	Percentage of noise level					
	0%	10%	20%	30%	40%	50%
Non-processed TS	86.7	82.6	79.0	75.1	71.5	67.7
Usual Decontamination	90.9	90.5	88.0	85.0	75.2	62.2
NCN Decontamination	91.8	90.9	89.5	85.3	77.1	64.0
Restricted Decontamination	87.7	88.1	88.2	88.2	87.9	84.1
NCN Restricted decontamination	87.1	87.3	87.8	88.1	88.3	85.3

Table 1. Experimental results with the Gaussian data. Mean values of the correct classification

Experimental results in Table 1 (corresponding to the Gaussian dataset), indicate that the Usual Decontamination procedure can handle quite well these situations with contamination in only one of the classes, but only up to 20% of noise. From 30% of contamination upwards, the previous methodology does not clean enough the TS.

The two variants here introduced are able to cope with high levels of partial contamination. Both, Restricted Decontamination and NCN Restricted Decontamination, allow classification accuracies not only better than those obtained when classification is done with the corresponding non-processed TS. They also permit to excel the classification accuracy yielded by the original no-contaminated TS (86.7%), up to 40% of contamination. That is, they produce a very satisfactory correction in the TS.

Restricted Decontamination with the surrounding neighborhood concept (NCN Restricted Decontamination) produces a slight increase in the classification accuracy, when the level of noise is very high (40% and 50%).

These results are better understood by looking at the behaviour of the different variants when analysing the number of prototypes in each class (see Table 2). The usual decontamination procedure is able to reach the original equilibrium among the classes, only when the level of noise is low (20% or less). In particular, with 50% of contamination (the last two columns of Table 2), application of the usual methodology results in an increased difference between the two amounts of prototypes. This behavior can be motivated not only for the high contamination present. It could be also an example of the effects introduced by the imbalance in the TS [24]. It is to remark that, when the methodology receives the contaminated TS, number of instances in class 1 is three times greater than that of class 2. The two variants proposed in the present paper (the last two rows), are more efficient when facing this problem.

	Percentage of noise level											
	0%		10%		20%		30%		40%		50%	
	c. 1	c. 2	c. 1	c.2	c. 1	c. 2	c. 1	c. 2	c. 1	c. 2	c. 1	c.2
Non-processed TS	100	100	110	90	120	80	130	70	140	60	150	50
Usual Decontamination	102.9	97.1	103.9	93.2	111.8	86.1	118.8	76.2	137.6	52	167.6	24.5
NCN Decontamination	99.9	97.5	103.3	92.7	108.5	85.1	118.5	77.3	136.8	55.9	165.3	26.4
Restricted decontamin.	82.4	117.4	85.4	114.4	87.8	110.3	91.0	107.3	98.1	100.9	108.0	89.6
NCN Restricted decont.	80.4	117.8	82.1	116.2	86.3	112.2	89.5	107.8	94.0	104.2	106.1	89.8

Table 2. Number of prototypes per class in the Gaussian dataset.

Results with the Uniform Dataset are very similar. See Tables 3 and 4.

	Percentage of noise level					
	0%	10%	20%	30%	40%	50%
Non-processed TS	96.2	91.7	86.4	81.7	77.6	73.6
Usual Decontamination	94.1	89.4	83.7	81.5	74.6	72.3
NCN Decontamination	93.5	88.1	84.0	81.2	75.7	72.1
Restricted Decontamination	94.6	92.3	87.0	82.9	79.9	73.7
NCN Restricted decontamination	95.1	93.1	87.2	83.0	80.4	73.9

Table 3. Experimental results with the Uniform dataset. Mean values of the correct classification

	Percentage of noise level											
	0%		10%		20%		30%		40%		50%	
	c. 1	c. 2	c. 1	c. 2	c. 1	c. 2	c. 1	c. 2	c. 1	c. 2	c. 1	c. 2
Non-processed TS	98	98	108	88	118	78	128	68	138	58	147	49
Usual Decontamination	93.1	96.9	107.2	83.2	115.7	74.8	122.7	68.3	137.1	50.4	147.3	45.5
NCN Decontamination	93.1	94.0	105.7	83.1	115.6	72.5	125.4	64.3	138.6	51.4	146.7	46.4
Restricted decontamination	90-8	102.3	95.4	95.0	108.2	85.5	117.8	77.9	127.2	66.6	143.5	51.5
NCN Restricted decontam.	92.9	100.9	95.2	97.7	107.5	87.2	118.6	76.8	125.5	66.6	144.5	50.9

Table 4. Number of prototypes per class in the uniform dataset experiments

5. Some final comments

Adequacy of the training data is one of the factors with a great influence on the performance of any learning algorithm. Many researches have been involved with the problems produced by the presence in the TS of atypical data. In previous works [2,3], a methodology for correcting the TS has been presented. The present paper presents two modifications to this methodology, for using it in those real applications when contamination is not suspected in all the classes. These two variants offer an important contribution to amend deficiencies of the available TS and to increase its usefulness. Experimental results with artificial data have revealed that these proposals can cope with very high contamination levels and that they permit generalization accuracies even greater than those obtained with the original (without mislabeled instances) training sets.

With 50% of noise, the problem is not only the contamination present in the TS but also the imbalance among the classes. We intend to do further research on this issue. One of the techniques we are going to explore is the employment of a weighted distance in the classification stage [24].

Acknowledgements. This work has been partially supported by grants 32016-A (Mexican CONACyT), TIC2000-1703-C03-03 (Spanish CICYT) and P1-1B2002-07 (Fundació Caixa Castelló-Bancaixa).

References

[1] B. V. Dasarathy, All you need to know about the neighbors, *Proceedings International Conference on Cybernetics and Society*, Denver U.S.A., 1979.

[2] R. Barandela and E. Gasca, Decontamination of training samples for supervised pattern recognition methods. *Lecture Notes in Computer Science*, 1876, 2000. 621-630.

[3] R. Barandela, E. Gasca and R. Alejo, Correcting the training data, in D. Chen and X. Cheng (eds.) *Pattern Recognition and String Matching* (Kluwer, 2003).

[4] G. Ritter and M.T. Gallegos, Outliers in statistical pattern recognition and an application to automatic chromosome classification. *Pattern Recognition Letters*, 18, 1997, 525-539.

[5] K- Urahama and Y. Furukawa, Gradient descent learning of nearest neighbor classifiers with outlier rejection. *Pattern Recognition*, 28(5), 1995, 761-768.

[6] M. Gopalakrishnan, V. Sridhar and H. Krishnamurthy, Some applications of clustering in the design of neural networks. *Pattern Recognition Letters*, 16, 1995, 59-65.

[7] G. H. John, *Enhancements to the Data Mining Process*. PhD Thesis, Stanford University (1997).

[8] S. Guha, R. Rastogi and K. Shim, CURE: An efficient clustering algorithm for large databases., Proceedings of the *ACM-SIGMOD International Conference On Management of Data*, Seattle, Washington, 1998.

[9] G. M. Foody, Directed ground survey for improved Maximum Likelihood classification of remotely sensed data, *International Journal of Remote Sensing*, 11(10), 1990, 1935-1940.

[10] G. M. Foody, N.A. Campbell, N.M. Trodd and T.D. Wood, Derivation and application of probabilistic measures of class membership from the maximum likelihood classification, *Phot. Eng. & Remote Sensing*, 58(9), 1992, 1335-1341.

[11] P. M. Mather, *Computer processing of remotely sensed images - an introduction*, (Wiley and Sons, Chichester, second edition, 1999).

[12] C. E. Brodley and M.A. Friedl. Identifying Mislabeled Training Data, *Journal of Artificial Intelligence Research*, 11, 1999, 131-167.

[13] B. V. Dasarathy, Is your Near Enough Neighbor friendly enough? Recognition in Partially Exposed Fuzzy Learning Environments., *Proceedings of the North American Fuzzy Information Processing Society*, 1993.

[14] R- Muzzolini, Y.H. Yang and R. Pierson, Classifier design with incomplete knowledge, *Pattern Recognition*, 31(4), 1998, 345-369.

[15] T- Denouex, A k-nearest neighbor classification rule based on Dempster-Shafer theory, *IEEE Transactions on Systems, Man and Cybernetics*, 25(5), 1995, 804-813.

[16] D. J. Hand, *Construction and assessment of classification rules*, (John Wiley & Sons, Chichester, 1997).

[17] D. M. J. Tax and R.P.W. Duin, Outlier Detection using Classifier Instability. In: A. Amin, D. Dori, P. Pudil and H. Freeman (eds.), *Advances in Pattern Recognition, Lecture Notes in Computer Science*, vol.1451, Springer, Berlin (1998).

[18] B. V. Dasarathy and B.V. Sheela, Design of composite classifier systems in imperfectly supervised environments, *Proceedings of the IEEE Computer Society on Pattern Recognition and Image Processing*, Chicago, 1979.

[19] D. R. Wilson and T.R. Martinez, Reduction Techniques for Instance-Based Learning Algorithms. *Machine Learning*, 38(3), 2000, 257-286.

[20] D. L. Wilson, Asymptotic properties of Nearest Neighbor rules using edited data sets. *IEEE Transactions on Systems, Man and Cybernetics,* SMC-2, 1972, 408-421.

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

[22] P. Devijver and J. Kittler, On the edited nearest neighbor rule with the resolution of a controversy, *Phillips Research Laboratory Report R410*, 1979.

[23] P. Hart, The condensed nearest neighbor rule. *IEEE Transactions on Information Theory,* IT-14, 1968, 505-516.

[24] R. Barandela, J. S. Sánchez, V. García and E. Rangel, Strategies for learning in class imbalance problems, Pattern Recognition, 36, 2003, 849-851.