Edition Schemes Based on BSE*

J. Arturo Olvera-López, J. Fco. Martínez-Trinidad, and J. Ariel Carrasco-Ochoa

Computer Science Department,
National Institute of Astrophysics, Optics and Electronics,
Luis Enrique Erro No. 1 Sta. María Tonantzintla, Puebla, CP: 72840, México
{aolvera,fmartine,ariel}@inaoep.mx

Abstract. Edition is an important and useful task in supervised classification specifically for instance-based classifiers because edition discards from the training set those useless or harmful objects for the classification accuracy and it helps to reduce the size of the original training sample and to increase both the classification speed and accuracy. In this paper, we propose two edition schemes that combine edition methods and sequential search for instance selection. In addition, we present an empirical comparison between these schemes and some other edition methods.

1 Introduction

Supervised classifiers work on a training set T or sample, that is, a set of objects previously assessed and labeled to classify a new object O. However, it is common that T contains objects with a null or even negative contribution for classification accuracy, these objects could be:

- *Noisy Objects*. These objects come from wrong measurements and they do not contribute to improve the classification accuracy because they lead wrong classification since the features values that describe the objects are not correct at all.
- *Superfluous Objects.* These objects have the characteristic that another object in T can generalize their description, that is, the superfluous objects are unnecessary objects.

These kinds of objects (noisy and superfluous) are useless or even harmful for the classification process. Therefore, it is convenient to consider only objects from the training set which are useful to obtain higher accuracy, that is, to apply an edition method to the training set.

The edition is defined as: given a training set T, choosing objects from T which contribute to improve the classification accuracy. The goal of edition methods is to find a training sample $S \subset T$ such that the classification accuracy using S would be higher than using T.

When a subset S from T is searched, we can proceed in three directions [1]:
Incremental. An incremental search begins with $S=\varnothing$ and in each step adds objects that fulfill the selection criteria.

* This work was financially supported by CONACyT (México) through the project J38707-A.

M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, LNCS 3773, pp. 360 – 367, 2005.

Decremental. This search begins with $S=T$ and removes from S objects that do not fulfill the selection criteria.

Batch. This search involves deciding if each object fulfills the removal criteria before removing any of them. Then all those objects that fulfill the criteria are removed at once, that is, this strategy does not remove one object at each step, it removes sets of objects.

In this paper, we will refer to edition schemes as those edition methods that are based on two steps; the first one consists of applying a pre-processing over the training set and the second one consists of editing the subset obtained in the first step.

In this paper, we propose two edition schemes, which reduce the runtimes of the decremental method *Backward Sequential Edition* (*BSE*) [2] and present an empirical comparison between these edition schemes and some other edition methods.

The structure of this paper is as follows: in section 2, the related work about edition methods is presented. Section 3 describes our proposed edition schemes. Section 4 presents some experimental results and finally in section 5 the conclusions and future work are given.

2 Related Work

In this section, some previous works related to edition methods are reviewed.

Wilson [3] introduced an edition method called *Edited Nearest Neighbor Algorithm (ENN)*, this method removes from S objects that do not agree with the majority of their k nearest neighbors. Wilson suggested a small and odd value for k, the *ENN* method uses $k=3$.

Wilson and Martínez [1] introduced the *DROP1,..., DROP5* methods (*Decremental Reduction Optimization Procedure*). The *DROP1* method is based on the rule: *remove an object O if at least as many of its associates in S would be classified correctly without O.* In this rule, an associate is an object such that O is one of its nearest neighbors. *DROP2* method considers the effect in T of removing an object in S, that is, *DROP2* removes the object O if its associates in T would be classified correctly without O. *DROP3* uses a noise-filtering step before applying *DROP2*; the noise filter used is similar to *ENN*.

DROP4 differs from *DROP3* in the filtering criterion since it is different to *ENN*. In this case, an object is removed only if it is misclassified by its k nearest neighbors and it does not hurt the classification of any other object. *DROP5* is similar to *DROP2* but *DROP5* starts with objects that are nearest to their nearest enemy, that is, nearest neighbors with different output class.

Brighton and Mellish [4] introduced the batch edition method *Iterative Case Filtering (ICF)*, this edition method is based on the *Reachable(O)* and *Coverage(O)* sets, which are based on the neighborhood and the set of associates of an object O. The edition rule is: *remove objects that have a Reachable set size greater than the Coverage set size*, that is, an object O is removed when some other objects could generalize the information from O. *ICF* starts applying *ENN* as noise filter.

In [2] the *Backward Sequential Edition* (*BSE*) was introduced, this method is based on backward sequential search; the *BSE* method starts from the original training sam-

ple *T* and finds a subset *S*. At each step, *BSE* removes the object (*WorstO*) with the smallest contribution for the subset quality, in terms of the accuracy of a classifier, which is calculated by the *Classifier()* function. In [2], *k-Nearest Neighbors (k-NN)* with k=3 is used as *Classifier()* function. The *BSE* method is depicted in figure 1.

```
BSE( Training set T ): Object set S
    Let S=T
    BestEval = Classifier(S)
    Repeat
        WorstO = null
        For each object O in S
            S' = S – {O}
            If Classifier(S') ≥ BestEval then
                WorstO = O
                BestEval = Classifier(S')
        If WorstO ≠ null then
            S = S – {WorstO}
    Until WorstO == null or S == ∅
    Return S
```

Fig. 1. *BSE* Method

In *BSE*, if there is more than one object with the smallest contribution, only the last is removed.

In [5] three edition methods were introduced: *Depuration, k-NCN* and iterative *k-NCN. Depuration* is based on the *generalized editing*, in which two parameters k and k' have to be defined, using the parameters the objects are removed or re-labeled (the original class label is changed). *k-NCN* editing is a modification of *ENN* and it consists of using the *k-NCN* (*Nearest Centroid Neighborhood*) instead of *k-NN*. Iterative *k-NCN* consists of applying repeatedly *k-NCN* until no more objects are removed.

In [6] the *NNEE* (*Neural Network Ensemble Editing*) method was proposed. It constructs a neural network ensemble from the training set *T* and changes the class label of each object in *T* to the class label predicted by the ensemble. *NNEE* does not remove objects, just changes class labels in order to increase the classification accuracy.

3 Proposed Schemes

In this section, we introduce two edition schemes in order to reduce the runtimes of *BSE* without a significant reduction in the classification accuracy. These schemes consist of a pre-processing over the training set before applying *BSE.*

It is very common that a training set contains noisy and/or superfluous objects. These objects are useless or harmful for the classification process because noisy objects lead to wrong predictions by classifiers and it is not necessary to store superfluous objects in the training set. Therefore, it is convenient to detect and discard those objects before starting the classification process.

The edition schemes proposed in this section are based on two main steps; the first one pre-processes the sample in order to detect and discard the objects above described, in this way, the size of the original sample is reduced. The second step edits the resultant pre-processed sample in order to increase the classification accuracy.

In the pre-processing step our proposed schemes uses either a noise filter method (remove noisy objects) or an edition method (remove superfluous objects). In the edition step we use *BSE* because according the results shown in [2], *BSE* reduce significantly the number of objects and increases the classification accuracy.

The first edition scheme (*ENN+BSE*) consists of applying *ENN* as noise filter in order to remove those useless noisy objects in the sample and after the clean subset is edited using *BSE*. When a set have been cleaned (filtered) the amount of comparisons in the classification process is reduced because a filtered set contains fewer objects than an unfiltered set. We use *ENN* as noise filter because it is a typical noise filter used in other edition schemes such as *ICF* and *DROP3.*

This scheme supposes that there are noisy objects in the training set, which can be removed in order to obtain a sample reduction in the pre-processing step. If there is not any noisy object, the scheme becomes the *BSE* method.

The second scheme (*DROP+BSE*) is based on editing an edited sample because after editing a sample, it is possible that some objects in the edited set do not contribute for the accuracy in the classification process (superfluous) because other objects in the edited set can generalize their description. This scheme consists of re-editing an edited sample in order to increase the classification accuracy. Our scheme uses *DROP3-DROP5* methods in the pre-processing step because these are the best *DROP* edition methods according to results reported in [1] and [2]. Finally, this scheme uses *BSE* to edit the edited sample.

In contrast to *ENN+BSE,* the sample reduction in *DROP+BSE* does not depend on the kind of objects in the original sample because the edition methods used in the pre-processing step remove some objects before the editing step.

The kind of objects preserved before the editing step depend on the method used in the pre-processing step, for example: *ENN* just removes noisy objects, *DROP3* and *DROP4* remove noisy and some other unnecessary objects, *DROP5* removes central, nosy and border objects.

4 Experimental Results

In this section, we present some experiments in order to compare the *BSE* method against *ENN+BSE* and *DROP+BSE* schemes. In addition, we compare these schemes against *DROP3, DROP4* and *ICF* methods because these methods could be considered as edition schemes since they apply a pre-processing step. Each method was tested on 10 datasets taken from the Machine Learning Database Repository at the University of California, Irvine [7].

The distance function for the experiments was the *Heterogeneous Value Difference Metric* (*HVDM*) [1], which is defined as:

$$HVDM(x,y)=\sqrt{\sum_{a=1}^{F} d_a^{\,2}(x_a,y_a)} \tag{1}$$

where $d_a(x,y)$ is the distance for the feature a and it is defined as:

$$d_a(x,y)=\begin{cases}1 & \text{if } x \text{ or } y \text{ is unknown}\\ vdm_a(x,y) & \text{if } a \text{ is nominal}\\ \dfrac{|x-y|}{4\sigma_a} & \text{if } a \text{ is numeric}\end{cases} \tag{2}$$

where σ_a is the standard deviation of the values occurring for feature a and $vdm_a(x,y)$ is defined as:

$$vdm_a(x,y)=\sum_{c=1}^{C}\left(\frac{N_{a,x,c}}{N_{a,x}}-\frac{N_{a,y,c}}{N_{a,y}}\right)^2 \tag{3}$$

Where $N_{a,x}$ is the number of times that the feature a takes value x in the training set; $N_{a,x,c}$ is the number of times that the feature a takes value x in the class c; and C is the number of classes.

In each experiment, 10 fold cross validation was used. The dataset was divided into 10 partitions and each edition algorithm was applied to T which is built with 9 of the 10 partitions (90% of the data) and the left partition (10% of the data) was the testing set. Each partition was used as testing set, so 10 tests were made with each dataset.

In Table 1, the results obtained with *k-NN* considering 100% of the data, *BSE* method and *ENN+BSE, DROP+BSE* schemes are shown. For each method, there are two columns; the left one (acc.) is the average classification accuracy and the right one (stor.) shows the percentage of the original training set that was retained by the edition method.

Based on the results shown in Table 1, we can see that the average accuracy of *ENN+BSE* and *DROP+BSE* schemes was higher than such obtained using the original set. On the other hand, the schemes accuracy was slightly smaller than *BSE*'s but the schemes had a lower average number of retained objects.

The schemes *ENN+BSE* and *DROP+BSE* do not improve the accuracy obtained with *BSE,* but the main advantage in these schemes is that their runtimes are shorter than the *BSE* runtimes since *BSE* is an expensive method because it analyses the accuracy impact of leaving out each object at each edition step.

In Table 2, average runtime results for *BSE, ENN+BSE* and *DROP+BSE* are shown. From Table 2 it could be noticed that the *ENN+BSE* and *DROP+BSE* runtimes are shorter than the spent by *BSE*. The complexity of *BSE* is $O(N^4F)$ where N is the total number of objects in the sample and F is the number of features.

The complexity of *ENN+BSE* and *DROP+BSE* schemes is also $O(n^4F)$ which is the same than *BSE*'s but applied with $n < N$ for *ENN+BSE* and $n << N$ for *DROP+BSE*. According to this, the proposed schemes do not reduce the complexity even though they reduce the runtimes.

Table 1. Accuracy and retention percentage for: *k-NN* with 100% of the data, *BSE* method and *ENN+BSE, DROP+BSE* schemes

Dataset	*k*-NN		BSE		ENN+BSE		DROP3+BSE		DROP4+BSE		DROP5+BSE	
	acc.	stor.	acc.	stor.	acc.	stor.	acc.	stor.	acc.	stor.	acc.	stor.
Breast Cancer(WI)	96.28	100	98.71	2.09	96.58	1.25	98.13	0.84	97.27	0.82	97.28	0.89
Cleveland	82.49	100	97.35	15.04	95.01	9.70	91.74	7.29	92.73	7.44	91.39	6.78
Glass	71.90	100	89.67	13.18	81.64	9.45	79.30	8.56	80.32	8.25	77.94	8.30
Hepatitis	80.62	100	97.41	9.24	92.87	4.08	82.20	3.22	89.04	3.58	86.41	3.43
Hungarian	79.55	100	94.27	14.88	91.13	4.64	86.72	3.40	91.12	4.95	91.09	5.29
Iris	94.67	100	99.33	6.14	98.66	5.55	99.30	5.85	98.66	5.03	96.66	5.03
Liver(Bupa)	65.22	100	96.52	12.69	91.58	14.20	90.45	7.85	91.63	9.34	89.02	9.31
Pima Indians	72.79	100	94.27	9.33	90.76	5.45	89.45	4.78	92.31	5.85	91.79	7.40
Thyroid	95.39	100	97.70	3.61	96.29	3.25	97.25	3.61	97.70	3.46	97.70	3.36
Zoo	94.44	100	97.77	10.86	93.33	10.24	91.11	21.36	95.56	8.27	96.82	8.64
Average	**83.33**	**100**	**96.30**	**9.70**	**92.78**	**6.78**	**90.56**	**6.67**	**92.63**	**5.69**	**91.61**	**5.84**

Table 2. Runtimes spent by *BSE,* and *ENN+BSE, DROP+BSE* schemes (*hrs.* = hours, *min.* = minutes and *sec.* = seconds)

Dataset	BSE	ENN+BSE	DROP3+BSE	DROP4+BSE	DROP5+BSE
Breast Cancer(WI)	6.9 hrs.	5.9 hrs.	18.4 sec.	45.5 sec.	59 sec.
Cleveland	7.2 hrs.	4.6 hrs.	40.6 sec.	1.19 min.	1.95 min.
Glass	6.5 min.	2.2 min.	19.9 sec.	39.3 sec.	22.5 sec.
Hepatitis	38.5 min.	24.4 min.	2.0 sec.	3.6 sec.	2.0 sec.
Hungarian	4.9 hrs.	3.4 hrs.	1.1 min.	58.8 sec.	2.2 min.
Iris	1.4 min.	1.2 min.	2.5 sec.	2.8 sec.	2.9 sec.
Liver(Bupa)	29.2 min.	8.4 min.	1.21 min.	1.29 min.	2.0 min.
Pima Indians	9.1 hrs.	3.4 hrs.	3.9 min.	7.4 min.	6.6 min.
Thyroid	18.7 min.	8.3 min.	2.8 sec.	4.2 sec.	2.3 sec.
Zoo	5.1 min.	2.8 min.	3.2 sec.	4.1 sec.	3.0 sec.

A second experiment was a comparison among the proposed schemes, *DROP3*, *DROP4* and *ICF*. The results are shown in Table 3.

From Table 3 we can see that schemes accuracy was better than the obtained with *ICF* and even with *DROP3*, which was better than *DROP4*. With *DROP4+BSE* were obtained both results: almost the best accuracy and the lowest percent of retention.

Finally, the proposed schemes were compared against other kind of edition methods: *Depuration* method (the best edition method reported in [5]) and the *NNEE* method. The results obtained are shown in Table 4 using the results reported in [6] for *Depuration* and *NNEE*. Here also, 10 fold cross validation was used.

In all cases the proposed schemes had better accuracy than *NNEE* and *Depuration*. *ENN+BSE* and *DROP+BSE* schemes have the advantage that they do not change the original distribution of the objects among the classes as *Depuration* and *NNEE* do.

Table 3. Accuracy and retention percentage for: *ICF, DROP3, DROP4* and *ENN+BSE, DROP3+BSE, DROP4+BSE* schemes

Dataset	*k*-NN		ICF		ENN+BSE		DROP3		DROP3+BSE		DROP4		DROP4+BSE	
	acc.	stor.	acc.	stor.	acc.	Stor.	acc.	stor.	Acc.	stor.	acc.	stor.	acc.	stor.
Breast Cancer(WI)	96.28	100	96.42	18.53	96.58	1.25	95.42	3.26	98.13	0.84	95.99	3.70	97.27	0.82
Cleveland	82.49	100	91.44	43.63	95.01	9.70	78.89	11.44	91.74	7.29	79.53	13.53	92.73	7.44
Glass	71.90	100	68.39	32.91	81.64	9.45	66.28	24.35	79.30	8.56	67.77	29.39	80.32	8.25
Hepatitis	80.62	100	77.95	18.71	92.87	4.08	81.87	7.81	82.20	3.22	78.75	9.75	89.04	3.58
Hungarian	79.55	100	84.58	29.63	91.13	4.64	80.84	12.76	86.72	3.40	78.19	15.26	91.12	4.95
Iris	94.67	100	94.00	45.03	98.66	5.55	95.33	15.33	99.30	5.85	94.67	15.26	98.66	5.03
Liver(Bupa)	65.22	100	59.68	27.63	91.58	14.20	67.82	26.83	90.45	7.85	66.41	33.11	91.63	9.34
Pima Indians	72.79	100	75.43	32.52	90.76	5.45	72.91	16.44	89.45	4.78	71.23	21.70	92.31	5.85
Thyroid	95.39	100	92.05	53.22	96.29	3.25	93.98	9.77	97.25	3.61	93.51	10.39	97.70	3.46
Zoo	94.44	100	81.22	16.54	93.33	10.24	90.00	20.37	91.11	21.36	91.11	21.36	95.56	8.27
Average	**83.33**	**100**	**82.12**	**31.84**	**92.78**	**6.78**	**82.33**	**14.83**	**90.56**	**6.67**	**81.71**	**17.34**	**92.63**	**5.69**

Table 4. Accuracy classification percentage for*: Depuration* (*Dep.*), *NNEE and ENN+BSE, DROP+BSE* schemes

Dataset	Dep.	NNEE	ENN + BSE	DROP3 + BSE	DROP4 + BSE	DROP5 + BSE
Glass	59.90	67.94	**81.64**	79.30	80.32	77.94
Iris	95.67	95.47	98.66	**99.30**	98.66	96.66
Liver	57.28	64.06	91.58	90.45	**91.63**	89.02
Pima Indians	72.42	75.57	90.76	89.45	**92.31**	91.79
Wine	94.94	96.05	**99.44**	**99.44**	**99.44**	**99.44**
Zoo	90.75	94.48	93.33	91.11	95.56	**96.82**
Average	**78.49**	**82.26**	**92.57**	**91.51**	**92.99**	**91.95**

5 Conclusions

The main disadvantage in instance-based classifiers is that they are expensive because the classification cost depends on the amount of objects in the training set and it is common that a training set contains useless or harmful objects for the classification accuracy. Therefore, it is necessary editing the training set in order to detect useful objects.

According to results shown in [2], *BSE* is a good edition method but a disadvantage of *BSE* is its high complexity. Our schemes reduce significantly the runtimes edition and the accuracy results are not too low with respect to *BSE.*

From the obtained results, we can conclude that our edition schemes are good options for solving edition problems since they obtained higher accuracy than *ICF, DROP3*, *DROP4* and even than *Depuration* and *NNEE*.

We used *ENN* and *DROPs* in the pre-processing step, but our edition schemes have not been proposed particularity to work only using these methods, some other methods can be used for pre-processing/pre-editing the sample before applying *BSE.*

Based on our experimental results, the main advantages of our schemes over other edition methods are: better accuracy results and low runtimes. In addition, our schemes do not change the original label of the objects as *Depuration* and *NNEE* do.

As future work, we will propose and test some edition schemes that do not depend on the *k-NN* rule and they do not hurt on both classification accuracy and edition runtimes.

References

1. Wilson, D. Randall and Martínez, Tony R. Reduction Techniques for Instance-Based Learning Algorithms. *Machine Learning*, vol 38, pp. 257-286, 2000.
2. Olvera-López, José A., Carrasco-Ochoa, J. Ariel and Martínez-Trinidad, José Fco. Sequential Search for Decremental Edition. Proceedings of the *6th International Conference on Intelligent Data Engineering and Automated Learning, IDEAL 2005*. Brisbane, Australia, vol 3578, pp. 280-285, LNCS Springer-Verlag, 2005.
3. Wilson, D. L. Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 2(3), pp. 408-421, 1972.
4. Brighton, H. and Mellish, C. Advances in Instance Selection for Instance-Based Learning Algorithms. *Data Mining and Knowledge Discovery,* 6, pp. 53-172, 2002.
5. Sánchez, J. S., Barandela, R., Marqués, A. I., Alejo, R., Badenas, J. Analysis of new techniques to obtain quality training sets. *Pattern Recognition Letters,* 24-7, pp. 1015-1022, 2003.
6. Jiang Y., Zhou, Z.-H. Editing training data for kNN classifiers with neural network ensemble. In: *Advances in Neural Networks,* LNCS 3173, pp. 356-361, Springer-Verlag, 2004.
7. Blake, C., Keogh, E., Merz, C.J.: UCI repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html], Department of Information and Computer Science, University of California, Irvine, CA, 1998.