Reducing Training Sets by NCN-based Exploratory Procedures [*]

M. Lozano, J.S. Sánchez, and F. Pla

Dept. Lenguajes y Sistemas Informáticos, Universitat Jaume I
Campus Riu Sec, 12071 Castellón, Spain
[lozano,sanchez,pla]@uji.es

Abstract. In this paper, a new approach to training set size reduction is presented. This scheme basically consists of defining a small number of prototypes that represent all the original instances. Although the ultimate aim of the algorithm proposed here is to obtain a strongly reduced training set, the performance is empirically evaluated over nine real datasets by comparing not only the reduction rate but also the classification accuracy with those of other condensing techniques.

1 Introduction

Currently, in many domains (e.g., in text categorisation, biometrics, and retrieval of multimedia databases) the size of the datasets is so extremely large that real-time systems cannot afford the time and storage requirements to process them. Under these conditions, classifying, understanding or compressing the available information can become a very problematic task. This problem is specially dramatic in the case of using some distance-based learning algorithm, such as the Nearest Neighbour (NN) rule [7]. The basic NN scheme must search through all the available training instances (large memory requirements) to classify a new input sample (slow during classification). On the other hand, since the NN rule stores every prototype in the training set (TS), noisy instances are stored as well, which can considerably degrade classification accuracy.

Among the many proposals to tackle this problem, a traditional method consists of removing some of the training prototypes, so the storage requirements and time necessary for classification are correspondingly reduced. In the Pattern Recognition literature, those methods leading to reduce the TS size are generally referred to as *prototype selection* [9]. Two different families of prototype selection methods can be defined. First, the *condensing* algorithms aim at selecting a sufficiently small subset of prototypes without a significant degradation of classification accuracy. Second, the *editing* approaches eliminate erroneous prototypes from the original TS and "clean" possible overlapping among regions from different classes.

[*] This work has been supported by grants TIC2000-1703-C03-03 and CPI2001-2956-C02-02 from CICYT Ministerio de Ciencia y Tecnología and project IST-2001-37306 from European Union.

Wilson introduced the first editing method [13]. Briefly, this consists of using the k-NN rule to estimate the class of each prototype in the TS, and removing those whose class label does not agree with that of the majority of its k-NN. This algorithm tries to eliminate mislabelled prototypes from the TS as well as those close to the decision boundaries. Subsequently, many researchers have addressed the problem of editing by proposing alternative schemes [1, 7, 9, 14].

Within the condensing perspective, the many existing proposals can be categorised into two main groups. First, those schemes that merely select a subset of the original prototypes [1, 8, 10] and second, those that modify the prototypes using a new representation [2, 4, 6]. It has been proven that the former family is partially inferior to the latter [3]. One problem related with using the original instances is that there may not be any vector located at the precise points that would make the most accurate learning algorithm. Thus, prototypes can be artificially generated to exist exactly where they are needed.

This paper focuses on the problem of appropriately reducing the TS size by selecting a subset of prototypes, in such a way that these represent all the instances in the original TS. The primary aim of the proposal presented in this paper is to obtain a considerable size reduction rate, but without an important decrease in classification accuracy.

The structure of the rest of this paper is as follows. Section 2 briefly reviews a set of TS size reduction techniques. The condensing algorithm proposed here is introduced in Section 3. The databases used and the experiments carried out are described in Section 4. Results are shown and discussed in Section 5. Finally, the main conclusions along with further extensions are depicted in Section 6.

2 Training Set Size Reduction Techniques

The problem of prototype selection is primarily related to prototype deletion as irrelevant and harmful prototypes are removed from a TS. This is the case, e.g., of Hart's condensing [10] and MCS scheme of Dasarathy [8], in which only critical prototypes are retained in the TS. On the other hand, some other algorithms artificially generate prototypes in locations accurately determined in order to reduce the TS size, instead of deciding which ones to retain. Within this category, we can find the algorithm presented by Chang [4] and by Chen and Józwik [6].

Hart's [10] algorithm is based on reducing the set size by eliminating prototypes. It is the earliest attempt at minimising the number of prototypes by retaining only a consistent subset of the original TS. A consistent subset, S, of a TS, T, is a subset that correctly classifies every prototype in T using the 1-NN rule. The minimal consistent subset is the most interesting, to minimise the cost of storage and the computing time. Hart's condensing does not guarantee finding the minimal subset as different subsets are given when the TS order is changed.

Chang's algorithm [4] consists of repeatedly attempting to merge the nearest two existing prototypes into a new single one. Two prototypes p and q are merged only if they are from the same class and, after replacing them with prototype z, the consistency property can be guaranteed.

Chen and Józwik [6] proposed an algorithm which consists of dividing the TS into some subsets using the concept of *diameter of a set* (i.e., the distance between the two farthest points). The algorithm starts by partitioning the TS into two subsets by the middle point between the two farthest cases. The next division is performed for the subset that contains a mixture of prototypes from different classes. If more than one subset satisfies this condition, then that with the largest diameter is divided. The number of partitions will be equal to the number of instances initially defined. Finally, each resulting subset is replaced by its centroid, which will assume the same class label as the majority of instances in the corresponding subset.

Recently, Ainslie and Sánchez introduced the family of IRSP algorithms [2], which are based on the idea of Chen's algorithm. The main difference between Chen and IRSP4 is that in the former, any subset containing a mixture of prototypes from different classes could be chosen to be divided. On the contrary, by IRSP4, the subset with the biggest overlapping degree (ratio of the average distance between prototypes belonging to different classes, and the average distance between instances being from the same class) is the one picked to be split. Furthermore, with IRSP4 the splitting process continues until every subset is homogeneous (i.e., all prototypes from a given subset are from a same class).

3 A New Approach to Training Set Size Reduction

The geometrical distribution among prototypes in a TS can become even more important than just the distance between them. In this sense, the so-called *surrounding neighbourhood-based rules* [12] try to obtain more suitable information about prototypes in the TS and specially, for those being close to decision boundaries. This can be achieved by taking into account not only the proximity of prototypes to a given input sample but also their *symmetrical distribution* around it.

Chaudhuri [5] proposed a neighbourhood definition, the Nearest Centroid Neighbourhood (NCN) concept, that can be viewed as a particular realization of the surrounding neighbourhood. Let p be a given point whose k NCN should be found in a TS, $X = \{x_1, \dots, x_n\}$. These k neighbours can be searched for through an iterative procedure in the following way:

1. The first NCN of p is also its NN, q_1.
2. The i-th NCN, q_i, $i \geq 2$, is such that the centroid of this and previously selected NCN, $q_1, \dots, q_i$ is the closest to p.

Neighbourhood obtained by this algorithm satisfies some interesting properties that can be further used to reduce the TS size by generating new prototypes. In particular, it is worth mentioning that the NCN search method is incremental and that the prototypes around a given sample have a geometrical distribution that tends to surround the sample, thus compensating the distribution of prototypes around the sample. It is also important to note that the region of influence of the NCN results bigger than that of the NN, as can be seen in Fig. 1.

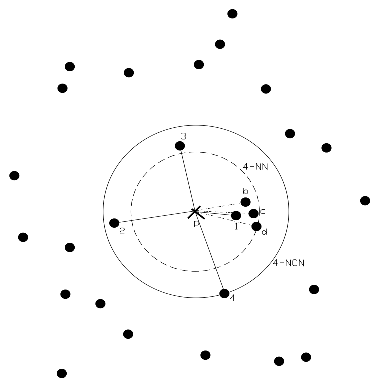

Fig. 1. Example of the NCN concept.

3.1 Algorithm Outline

The TS size reduction technique here proposed rests upon the NCN search algorithm. NCN search is used as an exploratory tool to bring out how prototypes in the data set are geometrically distributed. The use of the NCN of a given sample can provide local information about what is the shape of the probability class distribution depending on the nature and class of its NCN, that is, of the nature of the prototypes in its surrounding area.

The rationale behind it is that prototypes belonging to the same class are located in a neighbouring area and can be replaced by a single representative without significantly affecting the original boundaries. The main reason to employ the NCN, instead of the NN, is to benefit from the aforementioned properties that the NCN covers a bigger region than that of the NN and that they locate an area of influence around a given sample which is compensated in terms of their geometrical distribution.

The algorithm attempts to replace a group of neighbouring prototypes that belong to the same class by a representative. In order to decide which group of prototypes are to be replaced, we compute the NCN of each prototype p in the TS until reaching a neighbour with a class label different from that of p.

The prototype with the largest number of neighbours is defined as a representative of its corresponding group, which lie in the area of influence defined by the NCN distribution and consequently, all its members can be now removed from the TS. Another possibility is to replace the group by its centroid. In this case, the reduction of the data set is done by introducing new samples that replace groups of existing ones.

After this, for each prototype remaining in the set, we update the number of its neighbours if some was previously eliminated as belonging to the group of an already existing representative. This is repeated until there is no group of prototypes to be replaced by a representative. The basic scheme has been here named *MaxNCN*.

In order to obtain a more important size reduction, a further extension to the idea just described consists of iterating the general process until no more prototypes are removed from the TS. Algorithmically, the iterative version can be written as follows:

Algorithm 1 *Iterative MaxNCN*

```
while $eliminated\_prototypes > 0$ do
  for $i = eachprototype$ do
    $neighbours\_number[i] = 0$
    $neighbour = next\_neighbour(i)$
    while $neighbour.class == i.class$ do
      $neighbours\_vector[i] = Id(neighbour)$
      $neighbours\_number[i] + +$
      $neighbour = next\_neighbour(i)$
    end while
  end for
  while $Max\_neighbours() > 0$ do
    $EliminateNeighbours(id\_Max\_neighbours)$
  end while
end while
```

4 Databases and Experiments

Nine real data sets (see Table 1) have been taken from the UCI Repository [11] to assess the behaviour of the algorithms introduced in the previous section. The experiments have been conducted to compare MaxNCN and iterative MaxNCN with IRSP4, Chen's scheme and Hart's condensing, in terms of both TS size reduction and accuracy rate of the condensed 1-NN classification rule.

Table 1. Data sets used in the experiments.

Data set	No. classes	No. features	TS size	Test set size
Cancer	2	9	546	137
Pima	2	6	615	153
Glass	6	9	174	40
Heart	2	13	216	54
Liver	2	6	276	69
Vehicle	4	18	678	168
Vowel	11	10	429	99
Wine	3	13	144	34
Phoneme	2	5	4324	1080

The algorithms proposed in this paper, like in the case of Chen's and IRSP4, need to be applied in practice to overlap-free data sets (that is, there is no overlapping among regions from different classes). Thus, as a general rule and according to previously published results [2, 14], the Wilson's editing has been considered to properly remove possible overlapping between classes. The parameter involved (k) has been obtained in our experiments by performing a five-fold cross-validation experiment using only the TS and computing the average classification accuracies for different values of k and comparing them with the "no editing" option. The best edited set (including the non-edited TS) is thus selected as input for the different condensing schemes.

5 Experimental Results

Table 2 reports the 1-NN accuracy results obtained by using the different condensed sets. Values in brackets correspond to the standard deviation. Analogously, the reduction rates with respect to the edited TS are provided in Table 3. The average values for each method on the nine data sets are also included.

Table 2. Experimental results: 1-NN classification accuracy.

	Chen's	IRSP4	Hart's	Iterative	MaxNCN
Cancer	96.78 (1.25)	93,55 (3,70)	94,61 (2,94)	68,60 (3,42)	89,92 (4,61)
Pima	73.64 (2.85)	72,01 (4,52)	73,31 (3,69)	53,26 (5,80)	67,71 (5,45)
Glass	67.18 (3.90)	71,46 (3,13)	67,91 (4,60)	57,19 (9,69)	66,65 (6,28)
Heart	61.93 (5.22)	63,01 (5,11)	62,87 (4,27)	58,16 (7,26)	59,92 (5,53)
Liver	59.58 (5.15)	63,89 (7,73)	62,40 (5,76)	53,31 (8,55)	60,65 (6,74)
Vehicle	58.56 (2.46)	63,47 (1,96)	62,17 (2,16)	55,20 (4,42)	59,33 (2,17)
Vowel	60.16 (9.27)	96,02 (1,77)	90,74 (2,30)	78,63 (5,18)	90,73 (1,78)
Wine	69.31 (7.31)	69,66 (3,47)	71,71 (6,72)	62,50 (6,65)	60,77 (6,19)
Phoneme	70.03 (9.14)	71,60 (8,74)	71,04 (7,90)	65,06 (7,57)	70,00 (8,05)
Average	68.57 (5.17)	73,85 (4,46)	72,97 (4,48)	61,32 (9,95)	69,52 (5,20)

Table 3. Experimental results: set size reduction rate.

	Chen's	IRSP4	Hart's	Iterative	MaxNCN
Cancer	98.79	93,72	93,09	99,11	96,10
Pima	90.61	70,03	79,04	95,99	85,35
Glass	67.58	32,71	51,33	73,13	62,15
Heart	85.18	55,80	67,22	92,53	78,35
Liver	82.97	45,41	63,20	91,21	74,83
Vehicle	65.79	35,60	45,98	74,85	56,59
Vowel	79.64	39,54	75,97	84,23	75,09
Wine	86.75	73,13	78,79	89,03	84,83
Phoneme	94.51	69,90	87,91	98,16	90,88
Average	83.54	57,32	71,39	88,69	78,24

Several comments can be made from the results in these tables. As expected, classification accuracy strongly depends on the number of prototypes in the condensed set. Correspondingly, IRSP4 and Hart's algorithm obtain the highest classification accuracy almost without exception for all the data sets, but they also retain more prototypes than Chen's scheme and the procedures proposed here.

It is important to note that, in terms of reduction rate, the iterative MaxNCN eliminates much more prototypes than any other scheme. Nevertheless, it also obtains the worst classification accuracy. On the contrary, IRSP4 shows the highest accuracy but the lowest reduction percentage. Thus, looking for balancing between accuracy and storage reduction, one can observe that the best options are Hart's, Chen's and the plain MaxNCN approach.

In particular, MaxNCN provides an average accuracy of 69.52% (only 4% less than IRSP4, which is the best option in accuracy) with an average reduction rate of 78.24% (approximately 20% higher than that of IRSP4). Results given by Chen's algorithm are similar to those of the MaxNCN procedure, both in accuracy and reduction percentage.

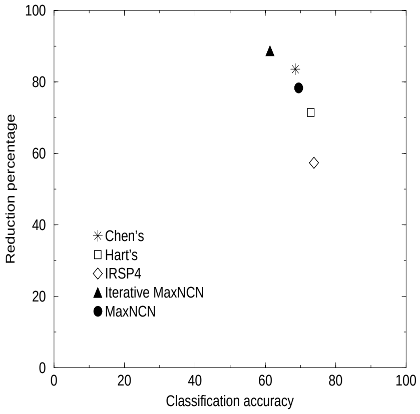

Fig. 2. Averaged accuracy and reduction rates.

In order to assess the performance relative to these two competing goals simultaneously, Fig. 2 represents the normalised Euclidean distance between each (accuracy, reduction) pair and the origin (0% accuracy, 0% reduction), in such a way that the "best" technique can be deemed as the one that is farthest from the origin. Thus, it is possible to see that the proposed MaxNCN approach along with Hart's and Chen's algorithms represent a good trade-off between accuracy and reduction rate.

Finally, it is to be noted that several alternatives to the algorithms here introduced have also been analysed, although all them had a behaviour similar to that of MaxNCN. For example, we defined a simple modification in which, instead of using an original prototype as representative of a neighbouring group, it computes the respective centroid of the NCN. Another alternative consisted of using the NN instead of the NCN, but the corresponding performance was systematically worst than that of MaxNCN.

6 Conclusions

In this paper, a new approach to TS size reduction has been introduced. This algorithm primarily consists of replacing a group of neighbouring prototypes that belong to a same class by a single representative. This group of prototypes is built by using the NCN, instead of the NN, of a given point because in general, those cover a bigger region than the one defined by the NN.

From the experiments carried out, it is apparent that the plain MaxNCN provides a well balanced trade-off between accuracy and TS size reduction rate, in clear contrast to the behaviour of the iterative version, which results in maximum reduction percentage and very poor accuracy performance.

An extension to the algorithms here proposed would consist of including a *consistency test* before removing a prototype from the TS. By this condition, we would try to keep the discriminating power and consequently, to increase the classification accuracy of the resulting condensed set.

References

1. D.W. Aha, D. Kibler, and M.K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.
2. M.C. Ainslie and J.S. Sánchez. Space partitioning for instance reduction in lazy learning algorithms. In *2nd Workshop on Integration and Collaboration Aspects of Data Mining, Decision Suport and Meta-Learning*, pages 13–18, 2002.
3. J.C. Bezdek and L.I. Kuncheva. Nearest prototype classifier designs: an experimental study. *International Journal of Intelligent Systems*, 16:1445–1473, 2001.
4. C.L. Chang. Finding prototypes for nearest neighbor clasifiers. *IEEE Trans. on Computers*, 23:1179–1184, 1974.
5. B.B. Chaudhuri. A new definition of neighbourhood of a point in multi-dimensional space. *Pattern Recognition Letters*, 17:11–17, 1996.
6. C.H. Chen and A. Józwik. A sample set condensation algorithm for the class sensitive artificial neural network. *Pattern Recognition Letters*, 17:819–823, 1996.
7. B.V. Dasarathy. *Nearest neighbor (NN) norms: NN pattern classification techniques.* IEEE Computer Society Press, Los Alamitos, CA, 1990.
8. B.V. Dasarathy. Minimal consistent subset (mcs) identification for optimal nearest neighbor decision systems design. *IEEE Trans. on Systems, Man, and Cybernetics*, 24:511–517, 1994.
9. P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach.* Prentice Hall, Englewood Cliffs, NJ, 1982.
10. P. Hart. The condensed nearest neigbor rule. *IEEE Trans on Information Theory*, 14:505–516, 1968.
11. C.J. Merz and P.M. Murphy. *UCI Repository of Machine Learning Databases.* Dept. of Information and Computer Science, U. of California, Irvine, CA, 1998.
12. J.S. Sánchez, F. Pla, and F.J. Ferri. On the use of neighbourhood-based non-parametric classifiers. *Pattern Recognition Letters*, 18:1179–1186, 1997.
13. D.L. Wilson. Asymptotic properties of nearest neigbor rules using edited data sets. *IEEE Trans. on Systems, Man and Cybernetics*, 2:408–421, 1972.
14. D.R. Wilson and T.R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38:257–286, 2000.