Data Reduction for Instance-Based Learning Using Entropy-Based Partitioning

Seung-Hyun Son and Jae-Yearn Kim

Department of Industrial Engineering, Hanyang University,
17 Haengdang-Dong, Sungdong-Ku,
Seoul, 133-791, South Korea
shson@ihanyang.ac.kr, jyk@hanyang.ac.kr

Abstract. Instance-based learning methods such as the nearest neighbor classifier have proven to perform well in pattern classification in several fields. Despite their high classification accuracy, they suffer from a high storage requirement, computational cost, and sensitivity to noise. In this paper, we present a data reduction method for instance-based learning, based on entropy-based partitioning and representative instances. Experimental results show that the new algorithm achieves a high data reduction rate as well as classification accuracy.

1 Introduction

As competition among corporations intensifies and awareness of the importance of information grows, data mining methods that can extract useful information from large amounts of data are receiving increased interest. Information discovered through data mining can facilitate informed decision-making. Among data mining's several methods, classification techniques create models that distinguish data classes. A model is used to predict the class of objects whose class label is unknown. For example, a classification model may be built to categorize bank loan applications as either safe or risky, and used in customer confidence assessments by credit card companies, as well as in many marketing fields.

This paper presents a data reduction method for instance-based learning that is designed to improve classification accuracy through data reduction using entropy-based partitioning and representative instances. Also, through this method, the original data set is purged of irrelevant attributes and the number of instances is decreased.

Several data reduction methods for classification purposes have been proposed. Liu, Hussain, Tan, and Dash introduce a data reduction method that differentiates between attribute values [1], and Cano, Herrera, and Lozano introduce a combination of stratification and evolutionary algorithms [2]. Datta and kibler introduced the prototype learner, which finds representative instances in each partition after dividing by the attribute value of each class [3]. They proposed a symbolic nearest mean classifier, which uses k-means clustering to group instances of the same class [4]. Wai Lam's prototype generation filtering (PGF)

M. Gavrilova et al. (Eds.): ICCSA 2006, LNCS 3982, pp. 590–599, 2006.

algorithm operates by combining nearest instances and preferentially calculating the distance of each instance [5]. J.S. Sanchez's reduction by space partitioning (RSP) algorithms divide the training set into several subsets based on its diameter [6]. The diameter of a set is defined as the distance between its two farthest instances.

Most existing research methods use clustering techniques to create several partitions that maintain the homogeneity of the data [3, 4, 5, 6]. These clustering techniques calculate the distances of all instances repeatedly and create clusters. However, if the size of data increases, the computing time increases greatly. Also, these methods consider all attributes, including those that are irrelevant, which further increases computing time.

This paper presents an algorithm that accelerates partitioning as compared to existing methods and can remove irrelevant attributes. In addition, this new method can find the representative instances of each partition more quickly.

Section 2 of this paper introduces instance-based learning and measures of entropy and distance; Section 3 describes the procedure used by the proposed algorithm; Section 4 applies an example to explain the proposed algorithm; Section 5 presents experimental results; and finally Section 6 presents conclusions.

2 Preliminaries

This section introduces instance-based learning and measures for finding data partitioning and center instances.

2.1 Instance-Based Learning

The instance-based learning is a machine learning technique that has proven to be successful over a wide range of classification problems. The instance-based knowledge representation uses the instances themselves to represent what is learned, rather than inferring a rule set or decision tree and storing it instead.

The nearest neighbor algorithm is one of the most widely studied examples of instance-based learning methods [7, 8, 9]. This algorithm retains all of the training set and classifies unseen cases by finding the class labels of instances that are closest to them. It learns very quickly, because it only needs to read a training set without much further processing, and it generalizes accurately for many applications. Despite its high classification accuracy, however, it has a relatively high storage requirement and because it must search through all instances to classify unseen cases, it is slow to perform classification. Data reduction for instance-based learning can be used to obtain a reduced representation of the data, while minimizing the loss of information content.

2.2 Entropy Measure

Let S be a set consisting of s data instances. Suppose the class label attribute has m distinct values defining m distinct classes, C_i (for $i = 1, \ldots, m$). Let s_i

be the number of instances of S in class C_i. The expected information needed to classify a given instance is given by

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} p_i \log_2(p_i) \ , \qquad (1)$$

where p_i is the probability that an arbitrary instance belongs to class C_i and is estimated by $\frac{s_i}{s}$ [10].

Let attribute $A1$ have v distinct values, $\{a_1, a_2, \ldots, a_v\}$. Attribute $A1$ can be used to partition S into v subsets,$\{S_1, S_2, \ldots, S_v\}$, where S_j contains those instances in S that have value a_j of $A1$. Let s_{ij} be the number of instances of class C_i in a subset S_j. The entropy based on the partitioning into subsets by $A1$, is given by

$$E(A1) = \sum_{j=1}^{v} \frac{s_{1j} + \ldots + s_{mj}}{s} I(s_{1j}, \ldots, s_{mj}) \ . \qquad (2)$$

The term $\frac{s_{1j}+\ldots+s_{mj}}{s}$acts as the weight of the jth subset and is the number of instances in the subset divided by the total number of instances in S. The smaller the entropy value is, the greater the purity of the subset partitions.

2.3 Distance Measure

The distance measure used to find the center instance in each partition is the Euclidean distance(ED) and is given by

$$ED(x, y) = \sqrt{\sum_{i=1}^{a} d(x_i, y_i)^2} \ , \qquad (3)$$

where x and y are two instances, a is the number of attributes, and x_i refers to the ith attribute value, for instance x [10]. For numerical attributes, $d(x_i, y_i)$ is defined as their absolute difference (i.e., $|x_i - y_i|$). For categorical attributes, the distance between two values is typically given by

$$d(x_i, y_i) = 0 \quad if\ x_i = y_i,\ and\ 1\ otherwie\ . \qquad (4)$$

The center instance is based on the sum of the distances between instances in each partition, i.e., the center instance x^{ith} of each partition is given by

$$min\{\sum_{k=1}^{n} ED(x^{1st}, y^{kth}), \sum_{k=1}^{n} ED(x^{2nd}, y^{kth}), \ldots, \sum_{k=1}^{n} ED(x^{nth}, y^{kth})\} \ , \qquad (5)$$

where n is the number of instances in each partition, x^{ith} and y^{kth} are the ith instance and the kth instance, respectively. The center instance is decided based on the least Euclidean distance measure. These formulas are used at the step that locates the center instance.

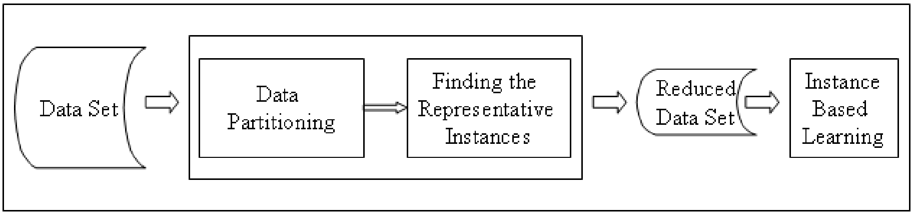


Fig. 1. Proposed algorithm process

3 Proposed Algorithm

The proposed algorithm consists of parts that seek the data partition and the representative instances. First, we calculate the entropy of each attribute. The data set is segmented preferentially via the attribute that has the smallest entropy. Using this method, data having homogeneity are gathered in the same partition, and each partition has the characteristics of the original data set. Second, we locate the representative instances using Euclidean distance. The representative instances consist of instances that represent the characteristics of each partition.

The procedure used by the proposed algorithm is as follows:

Steps 1–4: Data partitioning

Step 1. Calculate the entropy of all attributes using Equations (1) and (2). Select the attribute that has the lowest entropy. Partition the data set via the attribute's values.

Step 2. In each partition, calculate the entropy of the remaining attributes and redivide the partition via the attribute that has the smallest entropy value in each partition.

Step 3. This partitioning process continues until all partitions are pure, meaning all the class values are the same, or no further partitioning is possible.

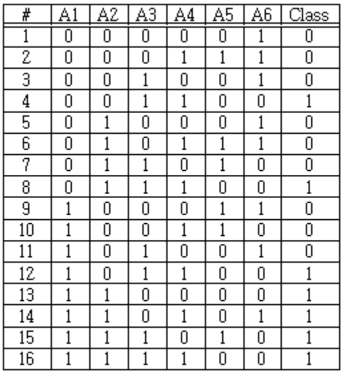
Step 4. Several partition sets are composed.

Steps 5–8: Finding the representative instances

Step 5. Find the center instance of each partition. The center instance is determined by using Equations (3), (4), and (5). In this case, not all attributes are considered to find the center instance; attributes that are used once at each partition can be ignored because they have the same value in each partition set, and attributes that are not used at the data partitioning stage are regarded as irrelevant attributes and are thus purged.

Step 6. In each partition, find the k nearest instances to the center instance; k is proportional to the number of instances in each partition.

Step 7. In each partition, find the representative instances. The representative instances consist of the union of the center instance and the k nearest instances to the center instance.



#	A1	A2	A3	A4	A5	A6	Class
1	0	0	0	0	0	1	0
2	0	0	0	1	1	1	0
3	0	0	1	0	0	1	0
4	0	0	1	1	0	0	1
5	0	1	0	0	0	1	0
6	0	1	0	1	1	1	0
7	0	1	1	0	1	0	0
8	0	1	1	1	0	0	1
9	1	0	0	0	1	1	0
10	1	0	0	1	1	0	0
11	1	0	1	0	0	1	0
12	1	0	1	1	0	0	1
13	1	1	0	0	0	0	1
14	1	1	0	1	0	1	1
15	1	1	1	0	1	0	1
16	1	1	1	1	0	0	1

Fig. 2. Example data set

Step 8. The reduced data set consists of the representative instances in each partition, and is used for the instance-based learning. The proposed algorithm process is shown in Figure 1.

4 Example

The example data set consists of 16 instances, 6 attributes, and 1 class, and is presented in Figure 2.
Steps 1–4: Data partitioning

Step 1. Calculate the entropy of all attributes. To calculate the expected information of attribute A1, we use Equation (1) as follows:

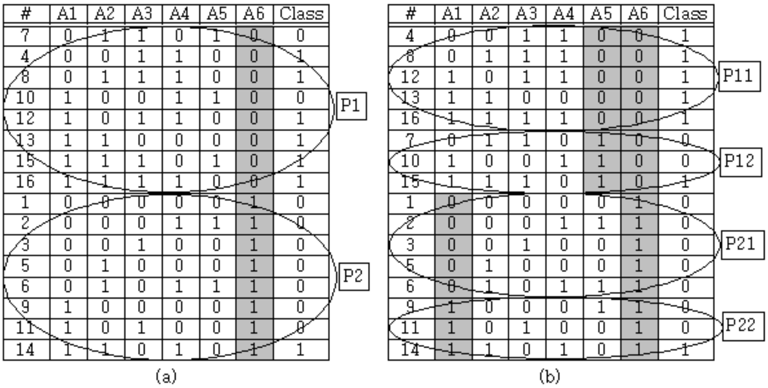
For $A1 = 0$:
$s_{11} = 6, s_{21} = 2$, $I(s_{11}, s_{21}) = -\frac{6}{8}log_2\frac{6}{8} - \frac{2}{8}log_2\frac{2}{8} = 0.31 + 0.50 = 0.81$
For $A1 = 1$:
$s_{12} = 3, s_{22} = 5$, $I(s_{12}, s_{22}) = -\frac{3}{8}log_2\frac{3}{8} - \frac{5}{8}log_2\frac{5}{8} = 0.53 + 0.42 = 0.95$

Using Equation (2), the entropy needed to classify a given sample if the samples are partitioned according to $A1$ is

$$E(A1) = \frac{8}{16}I(s_{11}, s_{21}) + \frac{8}{16}I(s_{12}, s_{22}) = (\frac{8}{16}) \times 0.81 + (\frac{8}{16}) \times 0.95 = 0.88.$$

Similarly, we can compute $E(A2) = 0.88$, $E(A3) = 0.88$, $E(A4) = 0.88$, $E(A5) = 0.82$, and $E(A6) = 0.60$.

Since $A6$ has the lowest entropy among the attributes, it is selected as the first attribute to use for partitioning. Next, divide the example data set by the values of attribute $A6$. The partitions are $P1$ and $P2$. After step 1, the result of partitioning is shown in Figure 3(a). The attribute values considered in the partition are marked in gray.



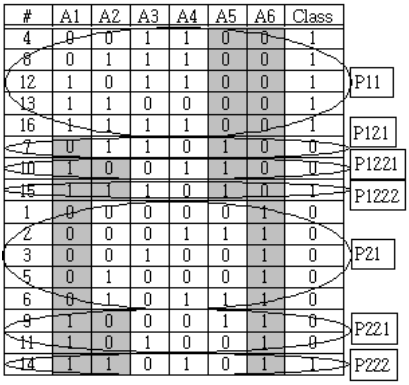
#	A1	A2	A3	A4	A5	A6	Class
7	0	1	1	0	1	0	0
4	0	0	1	1	0	0	1
8	0	1	1	1	0	0	1
10	1	0	0	1	1	0	0
12	1	0	1	1	0	0	1
13	1	1	0	0	0	0	1
15	1	1	1	0	1	0	1
16	1	1	1	1	0	0	1
1	0	0	0	0	0	1	0
2	0	0	0	1	1	1	0
3	0	0	1	0	0	1	0
5	0	1	0	0	0	1	0
6	0	1	0	1	1	1	0
9	1	0	0	0	0	1	0
11	1	0	1	0	0	1	0
14	1	1	0	1	0	1	1

(a)

#	A1	A2	A3	A4	A5	A6	Class
4	0	0	1	1	0	0	1
8	0	1	1	1	0	0	1
12	1	0	1	1	0	0	1
13	1	1	0	0	0	0	1
16	1	1	1	1	0	0	1
7	0	1	1	0	1	0	0
10	1	0	0	1	1	0	0
15	1	1	1	0	1	0	1
1	0	0	0	0	0	1	0
2	0	0	0	1	1	1	0
3	0	0	1	0	0	1	0
5	0	1	0	0	0	1	0
6	0	1	0	1	1	1	0
9	1	0	0	0	1	1	0
11	1	0	1	0	0	1	0
14	1	1	0	1	0	1	1

(b)

Fig. 3. (a) Partition sets after step1 (b) Partition sets after step 2



#	A1	A2	A3	A4	A5	A6	Class
4	0	0	1	1	0	0	1
8	0	1	1	1	0	0	1
12	1	0	1	1	0	0	1
13	1	1	0	0	0	0	1
16	1	1	1	1	0	0	1
7	0	1	1	0	1	0	0
10	1	0	0	1	1	0	0
15	1	1	1	0	1	0	1
1	0	0	0	0	0	1	0
2	0	0	0	1	1	1	0
3	0	0	1	0	0	1	0
5	0	1	0	0	0	1	0
6	0	1	0	1	1	1	0
9	1	0	0	0	1	1	0
11	1	0	1	0	0	1	0
14	1	1	0	1	0	1	1

P11
P121
P1221
P1222
P21
P221
P222

Fig. 4. Partition sets after data partitioning

Step 2. Partitions $P1$ and $P2$ are created. For each partition, calculate the entropy of the attributes that are not considered.

Because the entropy of attribute $A5$ is the smallest in partition $P1$, partition $P1$ is divided using attribute $A5$. Partition $P2$ is divided using attribute $A1$ for the same reason. The partitioning results after step 2 are shown in Figure 3(b). Partitions $P11$ and $P12$ are formed in partition $P1$, and partitions $P21$ and $P22$ are formed in partition $P2$.

Step 3. This partitioning process can continue until all partitions are pure or no further partitioning can be done.

Step 4. : Seven partition sets are composed in example: $\{P11$, $P121$, $P1221$, $P1222$, $P21$, $P221$, and $P222\}$. The results after the data partitioning are shown in Figure 4.

Steps 5–8: Finding the representative instances

Step 5. Find the center instance of each partition using Equation (5).

When finding the center instance for partition $P11$, only attributes $A1$ and $A2$ are used. Therefore, this method has the advantage of being able to locate the center instance faster, using only some and not all attributes. The center instance is decided based on the least Euclidean distance measure. For example, if we calculate the sum of the distances between instances for all instances in partition $P11$, we get the following:

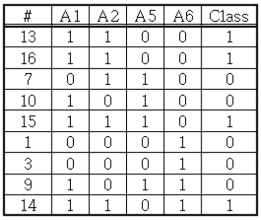
The sum of the distances with instance #4 and the remaining instances = $\sqrt{1} + \sqrt{1} + \sqrt{2} + \sqrt{2} = 2 + 2\sqrt{2}$,
The sum of the distances with instance #8 and the remaining instances = $\sqrt{1} + \sqrt{2} + \sqrt{1} + \sqrt{1} = 3 + \sqrt{2}$,
The sum of the distances with instance #12 and the remaining instances = $\sqrt{1} + \sqrt{2} + \sqrt{1} + \sqrt{1} = 3 + \sqrt{2}$,
The sum of the distances with instance #13 and the remaining instances = $\sqrt{2} + \sqrt{1} + \sqrt{1} + \sqrt{0} = 2 + \sqrt{2}$,
The sum of the distances with instance #16 and the remaining instances = $\sqrt{2} + \sqrt{1} + \sqrt{1} + \sqrt{0} = 2 + \sqrt{2}$.

Therefore, the center instance in partition $P11$ becomes instance #13 or instance #16.

Step 6. In each partition, find the k instances nearest to the center instance. k is predefined at the data partitioning stage. For example, *one* instance is selected in partition $P11$, and also *one* is selected in $P21$. And no *one* instance is selected in the remaining partitions because they have only *one* or *two* instances.

Step 7. Find the representative instances in each partition. For example, *two* instances are selected in partition $P11$: *one* center instance and *one* nearest instance. The result is the same in Partition $P21$. Only *one* center instance is selected as the representative instance in the remaining partitions.

Step 8. The reduced data set consists of the representative instances in each partition. For example, the final reduced data set is shown in Figure 5.



#	A1	A2	A5	A6	Class
13	1	1	0	0	1
16	1	1	0	0	1
7	0	1	1	0	0
10	1	0	1	0	0
15	1	1	1	0	1
1	0	0	0	1	0
3	0	0	0	1	0
9	1	0	1	1	0
14	1	1	0	1	1

Fig. 5. Final reduced data set

5 Experimental Results

This data reduction algorithm was empirically compared with the k-nearest neighbor(k-nn) algorithm and Wai Lam's PGF algorithm. Six data sets from

Table 1. Data sets

Data set	Number of instances	Number of attributes
Zoo	101	16
Audiology	226	63
Vote	435	17
Soybean	683	36
Credit	690	5
Mushroom	8124	22

the widely used UCI Database Repository were tested in the experiments [11]. The data sets used in the experiments are shown in Table 1. Experiments were performed on a PC with Pentium IV 3.0 GHz CPU and 512 MB of RAM. The implementation was done using Visual C++. For each data set, ten-fold cross-validation was used to estimate the average classification accuracy [10]. For each data set, we randomly partitioned the data into ten mutually exclusive subsets, $S_1, S_2, \ldots, S_{10}$, each of approximately equal size. Training and testing were performed 10 times. The classifier of the first iteration was trained on subsets $S_2, S_3, \ldots, S_{10}$ and tested on S_1, and the classifier of the second iteration was trained on subsets $S_1, S_3, \ldots, S_{10}$ and tested on S_2, and so on.

The classification accuracy estimate is the overall number of correct classifications from the ten iterations, divided by the total number of instances in the data set. The data reduction rate is the product of the number of instances and the number of attributes in the reduced data set, divided by the product of the number of instances and the number of attributes in the original data set. Note that higher classification accuracy and a better data reduction rate imply better performance.

The detailed performance of each algorithm for each individual data set can be found in Table 2. Symbols "—" indicate that k-nn algorithm retains 100% of the original size (i.e., data reduction rate is 0%). The proposed algorithm removes 93.64% of the original size on average and the average accuracy is 89.42%.

In the Zoo data set, using the proposed algorithm, irrelevant 7 attributes among 16 attributes were removed, and the data set consisted of 14 representative instances. The data reduction rate of the Zoo data set is shown in Figure 6.

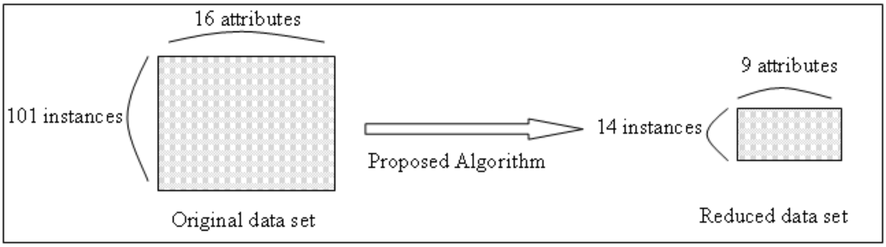


Fig. 6. Data reduction rate of the Zoo data set

Table 2. Classification accuracy(Accuracy) and data reduction rate(Size) for each data set

Data set		k-nn	PGF	Proposed Algorithm
Zoo	Accuracy	97.00	90.00	92.10
	Size	—	91.10	92.20
Audiology	Accuracy	76.10	67.20	78.50
	Size	—	87.00	90.10
Vote	Accuracy	95.50	92.60	92.80
	Size	—	93.90	94.00
Soybean	Accuracy	90.80	89.10	90.50
	Size	—	87.90	88.35
Credit	Accuracy	80.70	84,50	86.70
	Size	—	97.70	97.80
Mushroom	Accuracy	99.90	99.60	95.50
	Size	—	99.10	99.40
Average	Accuracy	89.67	87.17	89.42
	Size	—	92.78	93.64

Also, $2 - 7$ attributes among 16 attributes were used to find the representative instances in each partition, and only 3.8 attributes were used on average.

In the Audiology, Vote, and Soybean data set, our proposed algorithm resulted in greater accuracy in classification and better reduction than the PGF algorithm. In particular, the proposed algorithm reduced 63 attributes to 30 in the Audiology data set.

In the Credit data set, the data reduction rate was high (97.8%). An initial 690 instances were reduced to 15 instances.

In the Mushroom data set, many attributes were regarded as irrelevant and removed. An initial 8124 instances were reduced to 178 instances, and 22 attributes were pared to 6 attributes.

When the proposed algorithm was compared with the k-nn algorithm, classification accuracy demonstrated similar results. In the case of the credit data set, the classification accuracy of the proposed algorithm was higher (86.7%). Also, when the proposed algorithm was compared with the PGF algorithm, the classification accuracy and data reduction rate were high in most cases. Through entropy-based partitioning that computing time is less than agglomerative hierarchical clustering method like PGF algorithm, data reduction was also achieved more quickly.

6 Conclusions

We have presented a new data reduction method for instance-based learning that integrates the strength of instance partitioning and attribute selection. Reducing the amount of data for instance-based learning reduces data storage requirements, lowers computational costs, minimizes noise, and can facilitates a more rapid search.

Using the proposed algorithm, the initial data set is segmented into several partitions. Each partition is divided continuously based on entropy, and this partitioning process can continue until all partitions are pure or no further partitioning can be done. Finally, the homogeneity of instances in the same partition can be maintained. After dividing the initial data set, the attributes that are not used in the data partitioning stage are regarded as irrelevant and removed. Because irrelevant attributes can be removed, this method can find the representative instances of each partition more quickly than other methods.

Experimental results show that the proposed algorithm achieves a high data reduction rate as well as classification accuracy. The proposed algorithm can be employed to preprocess data used for data mining as well as in instance-based learning.

References

1. Liu, H., Hussain, F., Tan, C.L., Dash M.: Discretization: an enabling technique. Data Mining Knowledge Discovery. 6 (2002) 393-423
2. Cano, J.R., Herrera, F., Lozano M.: On the combination of evolutionary algorithms and strafitied strategies for training set selection in data mining. Applied Soft Computing, In Press, Correted Proof, (2005)
3. Datta, P., Kibler, D.: Learning prototypical concept description. Proceedings of the 12th International Conference on Machine Learning. (1995) 158-166
4. Datta, P., Kibler, D.: Symbolic nearest mean classifier. Proceedings of the 14th National Conference of Artificial Intelligence. (1997) 82-87
5. Lam, W., Keung, C.L., Ling C.X.: Learning good prototypes for classification using filtering and abstraction of instances. Pattern Recognition, Vol. 35. (2002) 1491-1506
6. Sanchez, J.S.: High training set size reduction by space partitioning and prototype abstraction. Pattern Recognition, Vol. 37. (2004) 1561-1564
7. Dasarath, B.V.: Nearest Neighbor Norms : NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos, CA (1991)
8. Wilson, D.R., Martinez, T.R.: Reduction Techniques for instance-based learning algorithms. Mach. Learning. 38 (2000) 257-286
9. Cano, J.R, Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in kdd: an experimental study. IEEE Transactions on Evolutionary Computation. 7 (6) (2003) 561-575
10. Han, J., Kamber M.: Data Mining : Concepts and Techniques. Morgan Kaufman (2001)
11. Merz, C.J., Murphy, P.M. : UCI Repository of Machine Learning Databases, Internet: http://www.ics.uci.edu/∼mlearn/MLRepository.html