Databases Reduction Simultaneously by Ordered Projection

Isabel Nepomuceno[1], Juan A. Nepomuceno[1],
Roberto Ruiz[1], and Jesús S. Aguilar–Ruiz[2]

[1] Department of Computer Science, University of Sevilla, Sevilla, Spain
{isabel, janepo, rruiz}@lsi.us.es
[2] Area of Computer Science, University Pablo de Olavide, Sevilla, Spain
jsagurui@upo.es

Abstract. In this paper, a new algorithm *Database Reduction Simultaneously by Ordered Projections* (RESOP) is introduced. This algorithm reduces databases in two directions: editing examples and feature selection simultaneously. Ordered projections techniques have been used to design RESOP taking advantage of symmetrical ideas for two different task. Experimental results have been made with UCI Repository databases and the performance for the latter application of classification techniques has been satisfactory.

1 Introduction

Nowadays the huge amount of information produced in different disciplines implies that manual analysis of data is not possible. Knowledge discovery in databases (KDD) deal with the problem of reducing and analyzing data with the use of automated analysis techniques. Data mining process is the previous process of extracting trends or patterns from data in order to transform data in useful and understandable information.

Data mining algorithms must work with databases with thousands of attributes and thousands of examples in order to extract trends or pattern from data. Databases preprocessing techniques are used to reduce the number of examples or attributes as a way of decreasing the size of the database with which we are working. There are two different types of preprocessing techniques, **editing**: reduction of the number of examples by eliminating some of them or finding representatives patterns or calculating prototypes and, secondly, **feature selection**: eliminating non-relevant attributes. Today's standard technology motivates more powerful methods which embed two different tasks at the same time.

In this paper, we propose an algorithm to embed horizontal and vertical database reduction simultaneously, that is, editing and feature selection at the same time. In section two the algorithm is presented. Section three shows experimental results carry out with the method proposed. Finally, conclusions are presented.

2 RESOP Algorithm

RESOP, *Reduction Database Simultaneously by Ordered Projection*, carries out editing of examples and features selection simultaneously using ideas of two algorithms based on ordered projections: EPO, see [1], and SOAP, see [2]. Techniques based on ordered projections use the idea of projecting every example over each attribute in order to create a partition in subsequence. Each subsequence is composed with examples of the same class. The aim is to built a partition of the space of examples in order to evaluate what example can be eliminated and what attribute is more relevant.

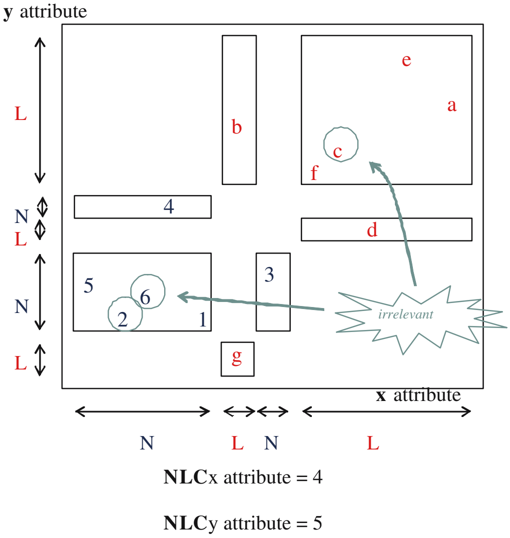

Fig. 1. RESOP algorithm applied over a two-dimensional database. Examples c, 6, 2 are eliminated and x attribute with NLC=4 is better to classify than y attribute with NLC=5.

RESOP idea is illustrated in Figure 1. A database with two attributes, x and y, and examples with two possible labels, letters or numbers, is given. Every example is projected over each axis: the valor over the corresponding attribute for each example is considered. We obtain in x axis the sequence $\{[5, 2, 6, 4, 1], [g, b], [3], [f, c, d, e, a]\}$ with label $\{N, L, N, L\}$. In y axis we obtain the sequence $\{[g], [1, 2, 6, 5, 3], [d], [4], [f, c, b, a, e]\}$. First attribute, x, has 4 label changes (NLC=4) $\{N, L, N, L\}$ and the second one, y, NLC=5, $\{L, N, L, N, L\}$. Firstly for vertical reduction, best attributes in order to classify are attributes with the lowest NLC. The generate ranking of examples is first attribute x and y the second one. Secondly, in order to horizontal reduction, examples which will be eliminated are examples which are not necessary to define the regions (i.e., to define classification rules): 2, 6 and c.

The algorithm has two different parts to handle: continuous and discrete attributes. See algorithm 1. Algorithm 2 is necessary to calculate the Number of Label Changes (NLC) for each attribute. λ parameter is used in order to relax the condition of editing for databases with a huge number of attributes and few examples, see [3].

3 Experimental Methodology

In this section, we present experiments carried out and the methodology used. The behavior of our algorithm (editing and selection of attributes) has been studied in several data sets available from UCI Repository [1], see [4].

Table 1. Comparison of the percentage of correctly classified instances for C4.5, IB1 and NB algorithms over the different medium size databases obtained with RESOP. PR is the percentage of data retention after the reduction. *na* is not available data after reduction process.

				IBK		**J48**		**NB**	
				Original	RESOP	Original	RESOP	Original	RESOP
Data	Size	λ	PR	ER	ER	ER	ER	ER	ER
ads	3279 × 1558	1	7.89	90.95	89.15	93.52	93.09	93.86	92.54
		0.95	na	na	na	na	na	na	na
		0.85	na	na	na	na	na	na	na
hypothyroid	3772 × 29	1	16.11	91.07	91.7	99.44	92.29	95.49	92.23
		0.95	9.41	91.07	87.54	99.44	89.63	95.49	89.66
		0.85	na	na	na	na	na	na	na
isolet	1559 × 617	1	50.00	84.22	68.36	74.67	63.43	81.58	76.04
		0.95	50.00	84.22	68.36	74.67	63.43	81.58	76.04
		0.85	50.00	84.22	68.36	74.67	63.43	81.58	76.04
letter	20000 × 16	1	49.18	95.53	86.51	87.54	81.45	64.08	52.87
		0.95	47.73	95.53	86.48	87.54	81.16	64.08	51.73
		0.85	43.34	95.53	86.34	87.54	81.41	64.08	50.21
mushroom	8124 × 22	1	14.47	99.79	95.09	99.8	94.74	0	82.92
		0.95	na	na	na	na	na	na	na
		0.85	na	na	na	na	na	na	na
musk2	6598 × 166	1	46.04	62.77	59.1	72.18	68.5	37.29	74.89
		0.95	15.98	62.77	52.87	72.18	66.82	37.29	72.69
		0.85	1.73	62.77	15.99	72.18	18.46	37.29	21.88
splice-2	3190 × 60	1	44.52	65.71	52.32	91.22	47.08	93.76	55.33
		0.95	na	na	na	na	na	na	na
		0.85	na	na	na	na	na	na	na

Data sets used are partitioned using ten-fold-cross-validation procedure. The algorithm runs for each data set. The original and the reduced data set are used as a training set in a classification algorithm and the percentage of correctly classified instances is measured. C4.5, IB1 and NB are used as classification algorithms. The purpose is to study the relevance of our method and the greater percent of correctly classified instances when the classification method is applied on the reduced data set. The reduction method algorithm is executed taken into account that a ranking of attributes is produced selecting the best 50% of this attributes and removing the remainder.

[1] http://www1.ics.uci.edu/ mlearn/MLRepository.html

In order to proof the goodness of our approach, in Table 1 results of the classification using C4.5, IB1 and NB techniques are shown. In this table we modify λ parameter. This parameter allow us to control the level of reduction of examples. The main objective is to compare the performance of our reduction method when the λ parameter adjust the number of instances to delete. The ER, error rate, for the original database and the different reduced databases is presented. ER is the percent of correctly classified instances produced when classifiers algorithms are applied. Finally PR is the percentage of data retention after the reduction. We must consider how the error changes when the database is reduced considerably. Our aim is to keep or increase the ER value after reduction database.

4 Conclusions

In this paper a new technique for reducing databases in two directions simultaneously is presented. On the one hand removing examples (editing examples) or vertical reduction, and on the other hand removing attributes (feature selection) or horizontal reduction. The method is based on using techniques of ordered projection, see [1,2], in order to reduce simultaneously examples and attributes.

Result obtained are satisfactory in order to evaluate the goodness of the proposal. Take into account today's standard technology, the method is very interesting from data mining techniques application point of view.

Future works will focus on making a comparison with order similar methods which edits examples and selects attributes, and studding the behavior of the algorithm with huge databases as for example microarrays.

References

1. Riquelme, José C.; Aguilar-Ruiz, Jesús S.; Toro, Miguel: Finding representative patterns with ordered projections Pattern Recognition 36 (2003), pp. 1009-1018.
2. Ruiz, R.; Riquelme, Jose C.; Aguilar-Ruiz, Jesus S.: NLC: A Measure Based on Projections 14th International Conference on Database and Expert Systems Applications, DEXA 2003Lecture Notes in Computer Science, Springer-VerlagPrague, Czech Republic, 1-5 September, (2003).
3. Jesús S. Aguilar-Ruiz, Juan A. Nepomuceno, Norberto Díaz-Díaz, Isabel A. Nepomuceno-Chamorro: A Measure for Data Set Editing by Ordered Projections. IEA/AIE (2006), pp. 1339-1348.
4. Blake, C.; Merz, E.K.: UCI repository of machine learning databases. (1998).

Algorithm 1 RESOP - database Reduction Simultaneously by Ordered Projections

```
INPUT $D$: data base
OUTPUT $D$: data base reduced, $k$ parameter
begin
  for all example $e_i \in D, i \in \{1, ..., n\}$ do
    weakness($e_i$) : = 0
  end for
  for all continuous attribute $a_j, j \in \{1, ..., m_1\}$ do
    $D_j := QuickSort(D_j, a_j) \quad in \quad incr. \quad order$
    $D_j = ReSort(D_j)$
    for all example $e_i \in E_j, i \in \{1, ..., n\}$ do
      if $e_i$ is not border then
        $weakness(e_i) := weakness(e_i) + 1$
      end if
      NLC ($a_j$) : = NumberLabelChanges(D,j)
    end for
  end for
  for all discrete attribute $a_j, j \in \{1, ..., m_2\}$ do
    for all value $v_i^j \in a_j$ do
      $V := \{e | value(e, a_j) = v_i^j\}$
      Let $\overline{e}$ be an example such that weakness($\overline{e}$) $= min_{i \in V}\{weakness(e)\}$
      for all $e_i \in V$ except $\overline{e}$ do
        $weakness(e_i) := weakness(e_i) + 1$
      end for
    end for
    NLC ($a_j$) : = NumberLabelChanges(D,j)
  end for
  for all example $e_i \in D, i \in \{1, ..., n\}$ do
    if $weakness(e_i) \geq m \cdot \lambda$ then
      remove $e_i$ from D
    end if
  end for
  NLC Attribute Ranking
  Select the k first attributes
```

Algorithm 2 NLC - Number Label of Ghanges

```
INPUT D, m: data base, number of attributes
OUTPUT nlc: number of label changes
  if att(u[j],i) $\in$ subsequence of the same value then
    nlc : = nlc + Change Same Value
  else
    if label(u[j]) <> lastLabel then
      nlc : = nlc + 1
    end if
  end if
```