Selecting Features and Objects for Mixed and Incomplete Data

Yenny Villuendas-Rey[1], Milton García-Borroto[2], and José Ruiz-Shulcloper[3]

[1] Ciego de Ávila University UNICA, C. de Ávila, Cuba
yennyv@bioplantas.cu
http://www.unica.cu
[2] Bioplantas Center, UNICA, C. de Ávila, Cuba
mil@bioplantas.cu
http://www.bioplantas.cu
[3] Advanced Technologies Applications Center, MINBAS, Cuba
jshulcloper@cenatav.co.cu
http://www.cenatav.co.cu/

Abstract. Selecting objects and features before classifying is a very important task, and can lead to big improvements in classifier accuracy and speed. There are many papers about this topic, but few of them consider the simultaneous or combined approach. In this paper, we present a new method for combined object and feature selection for databases with features not purely numeric or non-numeric. The experiments performed show that it attains the best tradeoff between object and feature reduction in 12 of 15 tested databases, without a significant impact in 1-NN accuracy.

Keywords: object selection, feature selection, supervised classification, classifier accuracy, mixed and incomplete data.

1 Introduction

Object selection for supervised classification has been widely studied in Pattern Recognition. Its goal is to obtain a reduced subset of objects with similar or improved classification accuracy. Eliminating redundant objects decreases the classifier computational cost, and removing erroneous or noisy objects achieves better performance. On the other hand, feature selection aims at obtaining a feature subset with similar or better behavior than the original set, by deleting irrelevant and redundant features. Several methods have been proposed for these tasks [1, 2].

As pointed out by Kuncheva and Jain [3] both object and feature selection aim at data reduction without a significant impact in the classification accuracy. Nevertheless, their semantics and search strategies are somewhat different, and we can consider them as complementary tasks. Their experiments showed that the jointly selection of features and objects may lead to better results than applying an object selection method followed by a feature selection method or vice versa. However, a few papers have explored the combined or simultaneous approaches.

J. Ruiz-Shulcloper and W.G. Kropatsch (Eds.): CIARP 2008, LNCS 5197, pp. 381–388, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Another relatively unexplored area in object and feature selection is related to problems where objects are simultaneously described by numeric and non-numeric features, and some missing data appears (Mixed and Incomplete Data, MID). In this kind of problems some Logical Combinatorial Pattern Recognition (LCPR) tools have been successfully used [4].

In this paper we introduce a novel method for combined object and feature selection in problems with MID. The proposed solution uses typical testors [5] for feature selection. A testor of a training matrix is a feature subset that does not confuse two subdescriptions of different classes in terms of these features. In the case of overlapping classes, this feature subset does not introduce new confusions. The typical testor (TT) is an irreducible testor, so any feature deletion introduces new confusions.

Despite the calculation of all typical testors of a given training matrix has exponential time complexity, algorithms like LEX [5] allow working with tens of features. Moreover, there are many real problems described in such amount of features. In our experimentation with 15 repository databases, we successfully compute all TT in a few minutes. The amount of typical testors in a training matrix is exponentially bounded, which can be a drawback for any TT-based method. However, in our experiments with databases ranging from 9 features and 286 objects to 36 features and 3196 objects, the number of TT vary from 1 to 45. We can also find this behavior in real world problems.

In our method, we use compact sets for object selection. A compact set (CS) is a connected component of the maximum similarity graph, so all the objects in the CS are very similar to each other. Compact sets have been successfully used for object selection in the Compact Set Editing method (CSE), a method specially designed for working with MID [6] with a good behavior in experiments.

We organize the paper as follows: Section 2 provides a review of the previous works, section 3 describes the proposed solution, section 4 presents the experimental results, and section 5 gives the conclusions.

2 Previous Works

In 1994 Skalak proposes the first algorithm for simultaneous feature and object selection (RMHC-FP) [7]. He uses a Random Mutation Hill Climbing algorithm, encoding features and objects in a binary string of length $|F|+|T|$, where the feature set is F and T the training object set. In the string, the value 0 represents the exclusion of a feature/object and the value 1 its inclusion. It generates an initial random string and iteratively applies a mutation operator while the classification accuracy is improved, or until it reaches a fixed number of iterations.

In 1999, Kuncheva and Jain [3] use a Genetic Algorithm (GA) for selecting simultaneously features and objects. They use binary chromosomes and the same encoding strategy of RMHC-FP. The selection strategy is elitist, so only the best chromosomes survive from one population to the next one. The authors use the following fitness function:

$$fitness = A_V - \alpha\left((sf + so)/\left(|F|+|T|\right)\right)$$

where A_V is the classifier accuracy with respect to a validation set V, α is a user defined parameter, sf is the number of selected features and so is the number of selected objects.

In 1999 Ishibushi and Nakashima [8] proposes another GA-based method with a different fitness function:

$$fitness = w_c NC - w_f sf + w_o so$$

where NC is the number of correctly classified objects of the training matrix, w_c is the weight associated with the accuracy, while w_f and w_o are the weights related with the number of features and objects selected, respectively. They also use two different mutation probabilities: a very small to mutate from 0 to 1 and a greater one to mutate from 1 to 0. This makes exclusions more probable than inclusions, allowing higher reduction in objects and features.

In 2000 Dasarathy proposes the first deterministic method for the combined selection of both features and objects [9]. His method employs a Sequential Backward Search (SBS) feature selection, using as fitness function a combined measure of the object reduction ratio and the classifier accuracy. For each SBS iteration, the algorithm projects the training sample using the proper feature subset, applies an object selection procedure and then calculates the accuracy and fitness value.

Rozsypal and Kubat propose in 2003 another GA-based method [10]. According to these authors, the method is more flexible and practical for large databases than previous methods, because it uses a different codification strategy: the value encoding. They use the following fitness function:

$$fitness = 1/\left(c_1 E_R + c_2 N_E + c_3 N_A\right)$$

where E_R is the number of misclassified objects of the training matrix, N_E is the number of selected objects and N_A is the number of selected features. The algorithm deletes objects with missing values and normalizes numeric features.

In 2005 Villuendas et al. [11] propose the first combined method for feature and object selection specially designed for MID. The SOFSA algorithm merges object-selected projections until it achieves at least the original accuracy. The algorithm calculates the original accuracy using all features and objects. Then it sorts the feature sets with respect to their informational weight. In some cases, it makes no reductions.

Recently, Ahn et al. [12] apply a GA to simultaneously select features and objects in a real world problem, with an encoding strategy similar to [3, 7]. They use the classification accuracy with respect to the test data as the fitness function for the GA.

Finally, we can conclude that most of the methods able to deal with MID are evolutionary, so they have an important random component. This causes that two different applications of the algorithm with the same data could have dramatically different results. Also, as pointed out by Kuncheva [3], GA can spend a long time to get a good solution. The deterministic solution proposed by Dasarathy is very time consuming, because it combines a time consuming object selection method with the SBS, a slow feature selection method. The other deterministic solution (SOFSA) may obtain no reductions at all.

3 Testor and Compact Set Based Combined Selection (TCCS) Method

We will explain the TCCS algorithm with an example, using a training matrix T of 5 objects, described by 5 features. Suppose we have 2 typical testors $\{x_1, x_2\}$ and $\{x_4\}$.

TCCS first projects the training matrix T using each typical testor. Then it applies the CSE method to each projection (Figure 1).

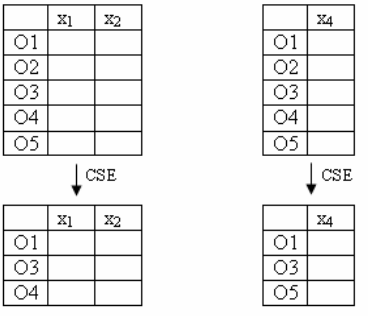

Fig. 1. Projecting and object selection in the TCCS method

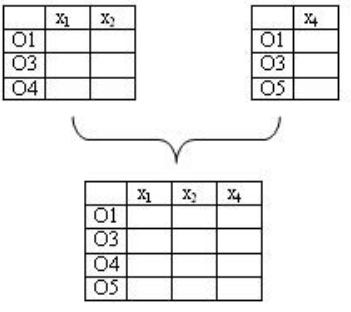

Fig. 2. Merging strategy of the TCCS method

The method computes the classifier accuracy with respect to the respective projected validation matrix V, and sorts the submatrixes in descending order according to the accuracy results. We consider the accuracy as a measure of the importance of the objects and features in the submatrix for the classifier. Therefore, if a current solution does not achieve the original accuracy it means that the procedure omitted some necessary object and/or features. That is why the algorithm iteratively merges (see figure 2) and applies CSE to the current solution until it reaches at least the original accuracy. It is important to note that a combination of two testors always leads to a new testor, so the merged solution keeps the discriminative power of the combined testors.

Formally, we can describe the algorithm as follows: Let be C a supervised classifier, T the training matrix and V the validation matrix.

Step 1. Calculate all typical testors TT in T.
Step 2. Calculate Acc_0, the initial accuracy of C trained with T classifying V.
Step 3. For each TT, create a submatrix S by projecting T using only the features in TT. Then, apply CSE method to S. Compute the accuracy of C, trained with S, classifying the respective projection of V.
Step 4. Sort the submatrixes in descending order according to the calculated accuracy, and use the first one as the initial solution.
Step 5. Calculate the accuracy of the current solution, and if it is greater than or equal to Acc_0, or all features are included, return the solution. Else, merge the current solution with the next submatrix, apply CSE to the result and repeat Step 5.

4 Experimental Results

We carry out experiments with the 15 UCI databases described in Table 1. We randomly split 5 times each database in Training (70%), Validation (20%) and Testing (10%), averaging the results.

Table 1. Description of the databases used in the experiments

UCI name	Objects	Numerical Features	Categorical Features	Missing Values
breast cancer	286	3	6	Yes
breast cancer Wisconsin	699	0	16	Yes
credit screening	690	6	9	Yes
echocardiogram	132	9	3	Yes
heart disease Cleveland	303	7	7	Yes
hepatitis	155	6	14	Yes
horse colic	368	10	18	Yes
hypothyroid	3772	7	23	Yes
import85	205	16	9	Yes
kdd Japanese vowels	640	12	0	No
kr-vs-kp	3196	0	36	No
mfeat morphological	2000	6	0	No
page blocks	5437	10	0	No
segment	2310	19	0	No
sick	3772	7	23	Yes

We compare the behavior of the proposed method with other simultaneous or combined approaches. The methods selected for the comparisons include some of the most cited or more recently published. They were the Skalak RMHC-FP, Kuncheva and Jain genetic algorithm (KJ-GA), Ichibuchi and Nakashima genetic algorithm (IN-GA), Ahn et al. genetic algorithm (AKH-GA), Dasarathy method (DM), and SOFSA. For RMHC-FP we substitute the City Block distance used in the original paper for HOEM [13], best suited for MID. We use the same function for the other methods and the same parameter values than in the original papers. Because of the high computational time complexity of both DS and IN-GA they exceed our maximum allowed execution time in some of the bigger databases (24 hours in a 1.8 GHz Dual Core processor with 2GB RAM), and they were not considered in the results. Our method takes less than a minute to execute in most databases, and has a maximum execution time of 30 minutes in *kr-vs-kp*.

As the main goal of a combined selection method is to reduce the training matrix dimensions with no significant impact in the classification accuracy, the better method is that which finds a better tradeoff between accuracy and reduction. To compare the behavior of these methods with respect to their tradeoff we build the Pareto frontier only with those methods having no significant accuracy degradation. The Pareto frontier is a common evaluation measure of multiobjective problems, and identifies those results for which no other result can simultaneously improves all the

objectives (Pareto Optimal). In this paper, a Pareto Optimal result is one that no other result has a lower retention in both objects and features simultaneously. That is, for keeping the accuracy any method with higher feature reduction has a lower object reduction and vice versa. To select the results with no significant accuracy degradation we made an independence t-test within a 0.05 confidence level.

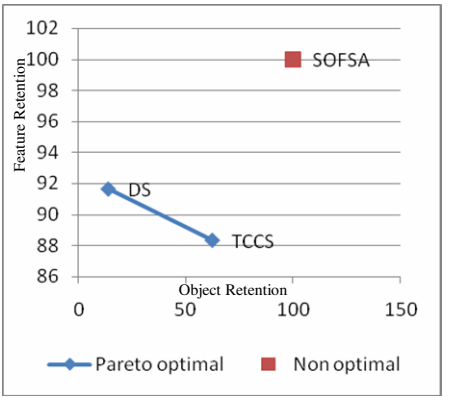

Fig. 3. Pareto frontier for *echocardiogram* database

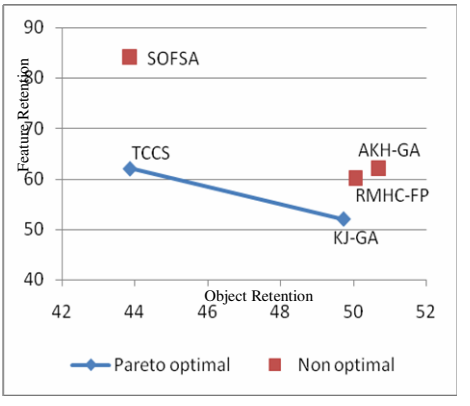

Fig. 4. Pareto frontier for *page-blocks* database

In Figure 3 and Figure 4, we show the Pareto frontiers for *echocardiogram* and *page-blocks* databases, respectively. The methods that do not appear in those figures had significant accuracy degradation. In Figure 3, DS, TCCS and SOFSA have no significant accuracy difference; but TCCS and DS have lower object and feature retention percents. Therefore, SOFSA is not Pareto optimal. On the other hand, DS and TCCS are both Pareto optimal because there is no method that can obtain lower object retention without higher feature retention and vice versa. We can follow an analogous reasoning for Figure 4.

In Table 2, we present the appearance rate of each method in the Pareto frontier for the 15 tested databases. Notice that TCCS significantly outperforms the other methods, achieving a good tradeoff between feature and object reduction in 80% of the databases. The closest method obtains a good tradeoff in less than 50% of the databases.

Table 2. Appearance rate in the Pareto frontier of each tested method

Method	Appearance rate
TCCS	12/15
SOFSA	7/15
AKH-GA	6/15
DS	4/15
KJ-GA	2/15
IN-GA	0
RMHC-FP	0

An explanation of these results can be the following. SOFSA maintains the original accuracy in five databases, but with reduction in neither features nor objects. It usually attains good reduction in objects, but TCCS outperforms it, because it uses a multiple edition scheme with the same algorithm. In five databases, TCCS outperforms the accuracy of all methods and the original classifier, reducing between 32% and 72% of objects, and between 9% and 70% of features. DM leads to an important reduction in objects but makes no significant reduction in features and in larger tested databases takes too much time to execute. Genetic strategies, on the other hand, had important degradations in accuracy.

After analyzing these results, we can conclude that our method achieves the best tradeoff between accuracy and reduction in both objects and features, being Pareto optimal in 80% of the tested databases.

5 Conclusions

In this paper, we introduce a new combined feature and object selection method for supervised classification. Our method is deterministic and does not use any evolutionary strategy, which is a distinctive characteristic with respect to most of the existing methods. It is based on the testor theory and compact sets, so it is especially well suited for Mixed and Incomplete Data. Based on the experimentation we can conclude that our method gets the best tradeoff between reduction rates in both objects and features with no significant drop in accuracy.

References

1. Bezdek, J.C., Kuncheva, L.I.: Nearest Prototype classifiers design: an experimental study. Technical Report, University of West Florida, pp. 1–37 (2004)
2. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. The Journal of Machine Learning Research 3, 1157–1182 (2003)
3. Kuncheva, L.I., Jain, L.C.: Nearest neighbor classifier: Simultaneous editing and feature selection. Pattern Recognition Letters 20, 1149–1156 (1999)
4. Ruiz-Shulcloper, J., Abidi, M.A.: Logical Combinatorial Pattern Recognition: A Review. In: Pandalai, S.G. (ed.) Recent Research Developments in Pattern Recognition. Transworld Research Networks, USA, pp. 133–176 (2002)
5. Santiesteban, Y., Pons-Porrata, A.: LEX: A new algorithm to calculate typical testors. Revista Ciencias Matemáticas 21, 118–126 (2003)
6. García-Borroto, M., Ruiz-Shulcloper, J.: Selecting Prototypes in Mixed Incomplete Data. In: Sanfeliu, A., Cortés, M.L. (eds.) CIARP 2005. LNCS, vol. 3773, pp. 450–459. Springer, Heidelberg (2005)
7. Skalak, D.B.: Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms. In: Eleventh International Machine Learning Conference, pp. 293–301. Morgan Kaufmann, New Brunswick (1994)
8. Ishibushi, H., Nakashima, T.: Evolution of reference sets in nearest neighbor classification. In: McKay, B., Yao, X., Newton, C.S., Kim, J.-H., Furuhashi, T. (eds.) SEAL 1998. LNCS (LNAI), vol. 1585, pp. 82–89. Springer, Heidelberg (1999)

9. Dasarathy, B.V.: Concurrent Feature and Prototype Selection in the Nearest Neighbor Decision Process. In: 4th World Multiconference on Systemics, Cybernetics and Informatics, vol. VII, pp. 628–633. Orlando, USA (2000)
10. Rozsypal, A., Kubat, M.: Selecting representative examples and attributes by a genetic algorithm. Intelligent Data Analysis 7, 291–304 (2003)
11. Villuendas-Rey, Y., García-Borroto, M., Medina-Pérez, M.A., Ruiz-Shulcloper, J.: Simultaneous features and objects selection for Mixed and Incomplete data. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 597–605. Springer, Heidelberg (2006)
12. Ahn, H., Kim, K.J., Han, I.: A case-based reasoning system with the two-dimensional reduction technique for customer classification. Expert Systems with Applications: An International Journal 32, 1011–1019 (2007)
13. Wilson, R.D., Martinez, T.R.: Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research 6, 1–34 (1997)