Simultaneous Features and Objects Selection for Mixed and Incomplete Data

Yenny Villuendas-Rey[1], Milton García-Borroto[2],
Miguel A. Medina-Pérez[1], and José Ruiz-Shulcloper[3]

[1] University of Ciego de Ávila, Cuba
{migue, yennyv}@bioplantas.cu
http://www.unica.cu
[2] Bioplants 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. In this paper a new simultaneous editing and feature selection method for the Most Similar Neighbor classifier is proposed. It is designed for databases with objects described by features no exclusively numeric or categorical. It is based on Testor Theory and the Compact Set Editing method, mixing edited projections until a good accuracy is achieved. Experimental results with several databases show a good performance compared to previous methods and the classifier using the original sample.

1 Introduction

The Nearest Neighbor classifier [1] is widely used on supervised classification. This classifier stores a training set (T) and each new object is assigned to the class of its closest object on T. According to this rule, a distance function is necessary for comparing objects.

In many applications we usually deal with Mixed and Incomplete Data (MID), because objects are described by features not purely numeric or categorical. Also, missing values could be present. In these cases, we usually have a similarity measure that is not an opposite of a distance function: it is not positive defined or does not assure the triangle inequality or is not symmetric. So, the Nearest Neighbor rule should be extended to the Most Similar Neighbor rule (MSN), allowing any similarity function. Though, it has the main disadvantages of the Nearest Neighbor rule: significant computational costs (similarities with all the objects in the training sample T must be assessed) and high storage requirements (all T must be stored).

Researches to reduce these drawbacks have been focused mainly on two separated approaches: reducing the number of objects and selecting a feature subset. Nevertheless, few studies have been addressed to solve the two problems jointly [2-5].

In this paper we introduce a new method for solving both problems with MID over non-metric spaces. In section 2 we summarize some jointly approaches reported in the literature. We also underline some of the main drawbacks of these methods. In

J.F. Martínez-Trinidad et al. (Eds.): CIARP 2006, LNCS 4225, pp. 597 – 605, 2006.

section 3 our method is described, and section 4 contains the experimental results. Finally, conclusions and future works are presented.

2 Related Works

Just a few authors (Skalak [5], Kuncheva [4], Ishibushi [3] and Dasarathy [2]) have faced the problems of simultaneously (or combined) feature and object selection for the NN classifiers.

Skalak [5] uses a Random Mutation Hill Climbing algorithm. It consists on the random generation of a binary string, which represents the inclusion/exclusion of an object or feature in the current solution. At each step, an element of the string is mutated, generating a new one. Accuracy with 1-NN is computed, and the best solution (in a fixed number of iterations) is selected.

Kuncheva [4] uses a Genetic Algorithm. In this case, the chromosome is a binary string of length $|F|+|T|$ (where F is the feature set and T is the training sample) which represents the selected features and objects. A value 1 in a bit of the string means the inclusion of this feature/object in the selected sample, 0 otherwise.

The fitness function used by the Genetic Algorithm is:

$$fitness = A_{1-NN}(V) - \alpha\left(\frac{sf + so}{|F| + |T|}\right) \tag{1}$$

where A_{1-NN} is the 1-NN accuracy in a validation sample V, sf is the number of the selected features, so is the number of the selected objects and α a constant parameter (user defined).

Analogously, Ishibushi [3] employs a Genetic Algorithm, but with a different fitness function:

$$fitness = w_{1-NN} * nwc(T) - w_f * sf - w_o * so \tag{2}$$

where nwc is the number of well classified objects in a training sample T, sf is the number of the selected features, so is the number of the selected objects and w_{1-NN}, w_f and w_o are the user defined weights associated with the accuracy, feature count and object count respectively.

The chromosome encoding of the algorithm is similar to the strategy used by Kuncheva.

These approaches have an important random component, so two different application of the algorithm with the same data could have dramatically different results. Besides, the results reached by these methods lack of a comprehensive meaning in the problem domain. Also, as pointed out by Kuncheva [4], Genetic Algorithms spent a long time to get to a good solution.

Dasarathy [2] proposes the application of a Sequential Backward Search (SBS), introduced by Kittler [6]. The SBS uses as Optimization Function a combined measure of the reduction ratio and the accuracy of the classifier with respect to (wrt)

the edited sample. This sample is obtained applying the Relative Neighborhood Graph based Editing (RNG-Edit) [7] and the Minimal Consistent Subset (MCS) [8].

This method have a big computational cost because their components (SBS, RNG-Edit and MCS), are all time consuming.

3 The Proposed Method

Specially designed to feature subsets selection with MID is the typical testors computation [9]. Typical testors are feature subsets that have two main properties: they do not confuse objects of different classes and are irreducible. So, a typical testor is a highly discriminative feature subset, with a clear meaning for specialists such as physicians, geologist, etc.

Typical testors have two main drawbacks: the computational cost and, in some databases, the amount of typical testors.

An object selection method designed to deal with MID is the Compact Set Editing (CSE) [10]. This algorithm has the desired property that is subclass consistent. So, the inner subclass structure of the training sample is preserved. CSE calculates the maximum similarity graph and make an ascending sort of the vertexes according to their indegree and outdegree. The nodes are iteratively discarded, starting from the firs, taking some actions to guarantee the subclass consistency.

The proposed algorithm is designed to deal with mixed and incomplete data, finding a new sample by reducing the numbers of features and objects in a simultaneous way. Besides, it tries to keep the 1-MSN classifier accuracy wrt a validation set as high as possible, using the selected sample.

The Simultaneous Object and Feature Selection Algorithm (SOFSA) proceeds as follows: firstly, we calculate the typical testors (TT) using the LEX algorithm [11],

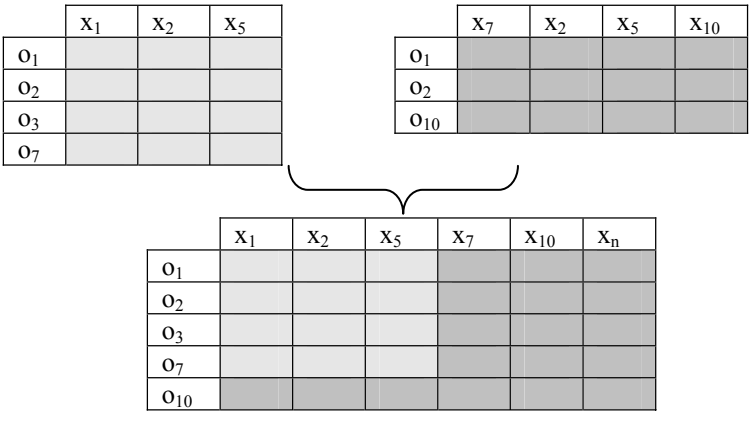

Fig. 1. Joint subsamples procedure used by SOFSA. All the features and objects of the parents are in the new subsample.

which has a good computational behavior for this task. Then, the TT are sorted according to their informational weights. Objects in the training sample are projected by the features presented in the first TT and CSE is applied to this projection. If accuracy wrt the validation sample is greater than the initial accuracy, calculated using all features and objects, a solution is returned, else the next TT is considered. In each previous step, we obtain a subsample. These subsamples generate a new subsample formed by the union of objects and features of each of them. See Fig 1.

Note that the process could be repeated for finding more accurate solutions than the initial one, but with more features and / or objects; until all projections are explored. In the cases that no solution improves the initial accuracy, we can choose an optimization function [2-4] that depends on accuracy result, object reduction ratio and feature reduction ratio. For example, we could use:

$$fitness = w_{1-NN} * A_{1-NN}(V) - w_f * \frac{sf}{|F|} - w_o * \frac{so}{|T|} \tag{3}$$

where A_{1-NN} is the classifier accuracy wrt the validation set, sf is the number of the selected features, so is the number of the selected objects and w_{1-NN}, w_f and w_o are the user defined weights associated with the accuracy, object reduction ratio and feature reduction ratio respectively.

Algorithm Parameters:

T	Training Sample
V	Validation Sample
CSE	Compact Set based Editing method
$\varepsilon_M^{Clasif}(V)$	Accuracy function wrt V, using $Clasif$ as classifier and M as training sample
ΩM	Partial objects descriptions of sample M using only the features in Ω

To compute the relevance of a feature x, we use the equation in [12]:

$$\rho(x) = \alpha * P(x) + \beta * L(X) \tag{4}$$

where $P(x)$ and $L(x)$ are the frequency informational weight and the length informational weight of feature x, respectively.

The frequency informational weight of x is given by:

$$P(x) = \frac{\tau_x}{\tau} \tag{5}$$

where τ_x is the amount of typical testors in which x appears and τ is the total typical testors amount.

The length informational weight of feature x is given by:

$$L(x) = \frac{\sum_{t \in TT_x} \frac{1}{|t|}}{|TT_x|} \tag{6}$$

where TT_x is the family of all typical testors in which x appears.

In our experiments, we use $\alpha = \beta = 0.5$ following [12]

SOFSA Algorithm:

1. $TT \leftarrow LEX(T)$ — Typical testors of T are computed using the LEX algorithm.
2. *forEach tt in TT* — The informational weight of each typical testor is calculated as the sum of the relevance of the features in the testor.
   $Weight_{tt} \leftarrow \sum_{x \in tt} \rho(x)$
3. $TT \leftarrow Sort(TT)$ — The typical testors are sorted according to their weights.
4. $InitialAcc \leftarrow \varepsilon_T^{1-MSN}(V)$ — The accuracy of 1-MSN classifier is computed.
5. $Solution \leftarrow []$
6. $found \leftarrow false$
7. *while not found* — While no solution is found, the algorithm proceeds to project the objects of the training sample by the features of the typical testor $\Omega \in TT$.
   $\Omega \leftarrow next(\Omega) \in TT$
   $Edited \leftarrow CSE(\Omega T)$ — Then, the CSE method is applied to obtain an edited sample.
   $Solution \leftarrow Join(Solution, Edited)$ — The edited sample is joined with the last solution, that is, all objects in the edited sample and all features in the typical testor are added to the solution.
   If $\varepsilon_{Solution}^{1-MSN}(V) > InitialAcc$ then — If the accuracy of the 1-MSN using the new solution is greater than initial accuracy,
   return Solution — a new solution is found and returned.
   $InitialAcc \leftarrow \varepsilon_{Solution}^{1-MSN}(V)$
   $found \leftarrow true$
8. End

4 Experimental Results

Traditional methods for selecting features and objects simultaneously assume metric spaces. In this paper they were extended to deal with MID in non-metric spaces, for

making numerical comparisons. For brevity, we name the extensions as: eDasarathy, eKuncheva, eIchibushi and eRMHC-FP respectively.

The experimental results were made using 4 databases from UCI, with Mixed and Incomplete Data. The description of these databases is shown in Table 1. In the experiments the databases were divided in Training (10% of the total of objects), Validation (20%) and Testing (70%).

Table 1. Description of the used databases

UCI name	Objects	Numerical Features	Categorical Features	Missing Values
Credit-screening	690	6	9	67
Hepatitis	155	6	14	167
Heart	270	6	7	0
Import-85 (symboling)	205	16	10	59

The similarity function used in our experiments was:

$$\Gamma(o,p)=\frac{\left|\left\{r\in F \mid C_r(o_r,p_r)=1\right\}\right|}{|F|} \tag{7}$$

where F is the set of all features, C_r the comparison criterion for the feature r; o_r and p_r are the values of the feature r of the objects o y p, respectively.

The Boolean similarity comparison criteria for numeric features were:

$$C_r(o_r,p_r)=\begin{cases}0 & \text{if } (o_r=?)\vee(\mathrm{p}_r=?)\vee(|o_r-p_r|\geq\sigma)\\ 1 & \text{otherwise}\end{cases} \tag{8}$$

where "?" denote a missing value, and σ is the standard deviation of the values of the feature.

For categorical data we use the criterion:

$$C_r(o_r,p_r)=\begin{cases}0 & \text{if } (\mathrm{o}_r=?)\vee(\mathrm{p}_r=?)\vee(o_r\neq p_r)\\ 1 & \text{otherwise}\end{cases} \tag{9}$$

The results for each used database are shown in tables 2-5. The accuracy function used was the ratio of well classified objects and the total amount of objects. The difference between accuracy without editing and accuracy after editing is shown in the Delta (Δ) column. As the databases are split randomly, we repeated the process 3 times to reduce the influence of randomness in partition, showing the mean in tables.

Table 2. Experimental results for database credit-screening. Accuracy without editing: 0.7428.

Method	Object Reduction (%)	Feature Reduction (%)	Edited Accuracy	Delta (Δ)
SOFSA	33.40	29.17	0.7590	0.0125
eDasarathy	83.61	6.67	0.7514	0.0086
eRMHC-FP	42.62	68.89	0.6707	-0.0722
eKuncheva	73.22	77.78	0.6071	-0.1357
eIshibushi	62.30	66.67	0.7377	-0.0052

In database credit-screening, the best method according to classifier accuracy is SOFSA, followed by eDasarathy. Both outperform the original accuracy of the 1-MSN classifier using only a reduced number of features and objects. According to reduction in both features and objects, eKuncheva shows the best performance; although the classifier accuracy is degrade in about 10%. Finally, eIshibushi shows a good reduction with a slightly drop in accuracy.

Table 3. Experimental results for database hepatitis. Accuracy without editing: 0.7161.

Method	Object Reduction (%)	Feature Reduction (%)	Edited Accuracy	Delta (Δ)
SOFSA	27.50	36.84	0.6938	-0.0223
eDasarathy	85.42	5.26	0.6615	-0.0547
eRMHC-FP	47.92	43.86	0.6797	-0.0365
eKuncheva	58.33	71.93	0.6094	-0.1068
eIshibushi	56.25	54.39	0.5000	-0.2161

In database hepatitis, no method outperforms classifier accuracy. SOFSA and eRMHC-FP give the best results according to accuracy. Both methods slightly degrade classifier accuracy, using a reduced number of features and objects. According to reduction, eKuncheva and eIshibuchi show the best performance, but the classifier accuracy is dropped in approximately 10% and 20% respectively.

Table 4. Experimental results for database heart. Accuracy without editing: 0.7619.

Method	Object Reduction (%)	Feature Reduction (%)	Edited Accuracy	Delta (Δ)
SOFSA	26.09	30.77	0.7619	0
eDasarathy	88.41	5.13	0.7904	0.0285
eRMHC-FP	44.93	51.28	0.6761	-0.0857
eKuncheva	52.17	79.49	0.6000	-0.1619
eIshibushi	52.17	61.54	0.6571	-0.1047

On database Heart, the best method according to accuracy results is eDasarathy. This method outperforms the unselected 1-MSN. SOFSA maintains original accuracy, reducing the 30% approximately of features and objects. All the other methods show a significant drop in accuracy. According to reduction, eKuncheva and eIshibushi both reduce about 50% to 70% of features and objects, with a drop of 10% in accuracy.

Table 5. Experimental results for database import-85. Accuracy without edititng: 0.4603.

Method	Object Reduction (%)	Feature Reduction (%)	Edited Accuracy	Delta (Δ)
SOFSA	8.33	37.00	0.4583	-0.0060
eDasarathy	84.03	4.00	0.4048	-0.0556
eRMHC-FP	45.14	42.67	0.4087	-0.0516
eKuncheva	27.08	36.00	0.2778	-0.1825
eIshibushi	27.08	36.00	0.3452	-0.1151

Finally, on database import-85, no method improves the classifier accuracy. SOFSA shows a little degradation, followed by eRMHC-FP. This last method shows the best results according to reduction in features and objects. The eRMHC-FP method obtains about a 45% of reduction in both features and objects.

The experimental results show that our method has a good performance over most of the databases. It reduces approximately 25 percent the number of features and objects with none or little degradations in classifier accuracy. In the numerical experiments it outperforms the other methods, except for database Heart, although in the database import-85 no significant reduction could be done.

eDasarathy had a stable behavior over all databases, but with insignificant reductions in the number of features. Random Mutation Hill Climbing and Genetic Algorithms show a good performance in some cases and in others the classifier performance is dramatically degraded. In these last three methods the reduction percent is always around half of features and objects.

5 Conclusions

Simultaneously (or combined) feature and object selection for improving the classifiers accuracy is a very important task in supervised classification problems. It was faced by several authors, but assuming metric spaces for 1-NN classifier. No procedure has been developed for MID in non-metric spaces for MSN classifiers. In this paper the methods introduced for metric spaces were extended for dealing with MID in non-metric spaces. Besides, a new algorithm is proposed to select jointly features and objects for the 1-MSN classifier. This is the first simultaneous features and objects selection algorithm specially designed to deal with databases containing objects described by features no exclusively numeric or categorical.

In most databases a good performance is showed by the proposed method, with respect to our extensions of earlier reported methods. Improvements or little degradations in the 1-MSN accuracy with a reduced number of features and objects are obtained by the method in all databases.

Several solutions could be returned by the method, which could be selected based on certain optimization criterion. These solutions have better accuracy, but the reduction in the number of objects and/or features is lesser.

All solutions returned by the method are formed by a highly discriminative feature subset, with a clear meaning for specialists. Also, the selected objects preserve the inner structure of the domain classes. Furthermore, this is a deterministic procedure, so we get unique solutions in two different application of the algorithm with the same data.

References

1. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley & Sons, N.Y. (1973)
2. Dasarathy, B.V.: Concurrent Feature and Prototype Selection in the Nearest Neighbor Decision Process. 4th World Multiconference on Systemics, Cybernetics and Informatics, Vol. VII, Orlando, USA (2000) 628/633
3. Ishibushi, H., Nakashima, T.: Evolution of reference sets in nearest neighbor classification. LNCS **1585** (1999) 82-89
4. Kuncheva, L.I., Jain, L.C.: Nearest neighbor classifier: Simultaneous editing and feature selection. Pattern Recognition Letters (1999)
5. Skalak, D.B.: Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms. Eleventh International Conference on Machine Learning (1994)
6. Kittler, J.: Feature set search algorithms. In: Chen, C.H. (ed.): Pattern recognition and signal proccessing. Sijthoff and Noordhoff, The Netherlands (1978)
7. Toussaint, G.T.: Proximity Graphs for Nearest Neighbor Decision Rules: Recent Progress. 34 Symposium on Computing and Statistics INTERFACE-2002, Montreal, Canada (2002)
8. Dasarathy, B.D.: Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. IEEE Transactions on systems, man and cybernetics. **24** (1994) 511-517
9. Ruiz-Shulcloper, J., Abidi, M.A.: Logical combinatorial pattern recognition: A Review. In: Pandalai, S.G. (ed.): Recent Research Developments in Pattern Recognition. Transword Research Networks, USA (2002) 133-176
10. García-Borroto, M., Ruiz-Shulcloper, J.: Selecting Prototypes in Mixed Incomplete Data. Lecture Notes in Computer Science **3773** (2005) 450-459
11. Santiesteban, Y., Pons, A.: Un nuevo algoritmo para el cálculo de los testores típicos. Revista de Ciencias Matemáticas **21** (2003) 85-95
12. Lazo-Cortés, M., Ruiz-Shulcloper, J.: Determining the feature informational weight for non-classical described objects and new algorithm to calculate fuzzy testors. Pattern Recognition Letters **16** (1995) 1259-1265