Incomplete Data Decomposition for Classification

Rafał Latkowski

Institute of Computer Science, Warsaw University
ul. Banacha 2, 02–097 Warsaw, Poland
rlatkows@mimuw.edu.pl

Abstract. In this paper we present a method of data decomposition to avoid the necessity of reasoning on data with missing attribute values. The original incomplete data is decomposed into data subsets without missing values. Next, methods for classifier induction are applied to such sets. Finally, a conflict resolving method is used to combine partial answers from classifiers to obtain final classification. We provide an empirical evaluation of the decomposition method with use of various decomposition criteria.

1 Introduction

In recent years a great research effort has been made to develop methods inducing classifiers for data with missing attribute values. Some approaches making possible to handle missing attribute values have been developed within the roughsets framework [7,14]. In those approaches a modification of indiscernibility relation is considered to handle missing attribute values. The other approach presented in *LEM1* and *LEM2* methods [4,5] is to modify an algorithm that search for covering set of decision rules. In this paper we present a method of data decomposition to avoid the necessity of reasoning on data with missing attribute values and without modification of the inductive learning algorithm itself.

The decomposition method was developed to meet certain assumptions. The primary aim was to find a possibility to adapt many existing, well known classification methods that are initially not able to handle missing attribute values to the case of incomplete data. The secondary aim was to cope with the problem of incomplete information systems without making an additional assumption on independent random distribution of missing values and without using data imputation methods [3,4]. Many real world applications have showed that appearance of missing values is governed by very complicated dependencies and the application of arbitrary method for data imputation can increase error rate of the classifier.

The decomposition method tries to avoid the necessity of reasoning on data with missing attribute values. The original incomplete data is decomposed into data subsets without missing values. Next, methods for classifier induction are applied to such sets. Finally, a conflict resolving method is used to combine

J.J. Alpigini et al. (Eds.): RSCTC 2002, LNAI 2475, pp. 413–420, 2002.

partial answers from classifiers to obtain final classification. In this paper we are focused on the selecting the efficient decomposition criteria for classification. We provide an empirical evaluation of the decomposition method in comparison to the Quinlan's C4.5 method [11,12].

2 Preliminaries

In searching for concept approximation we are considering a special type of information systems — decision tables $\mathbb{A} = (U, A \cup \{d\})$, where $d : U \rightarrow V_d$ is a decision attribute. In a presence of missing data we may consider the attributes $a_i \in A$ as a functions $a_i : U \rightarrow V_i^*$, where $V_i^* = V_i \cup \{*\}$ and $* \notin V_i$. The special symbol "$*$" denotes absence of regular attribute value and if $a_i(x) = *$ we say that a_i is not defined on x. We can interpret $a_i : U \rightarrow V_i^*$ as a *partial* function in contrast to $a_i : U \rightarrow V_i$ interpreted as a *total* function.

In such tables we can search for patterns of regularities in order to discover knowledge hidden in data. We would like to focus here on searching for regularities that are based on the presence of missing attribute values. A standard tool for describing a data regularities are *templates* [10,9]. The concept of template require some modification to be applicable to the problem of incomplete information table decomposition.

Definition 1. *Let $\mathbb{A} = (U, A \cup \{d\})$ be a decision table and let $a_i \in V_i$ be a* total descriptor*. An object $u \in U$ satisfies a total descriptor $a_i \in V_i$, if the value of the attribute $a_i \in A$ for this object u is not missing in $\mathbb{A}$, otherwise the object u does not satisfy total descriptor.*

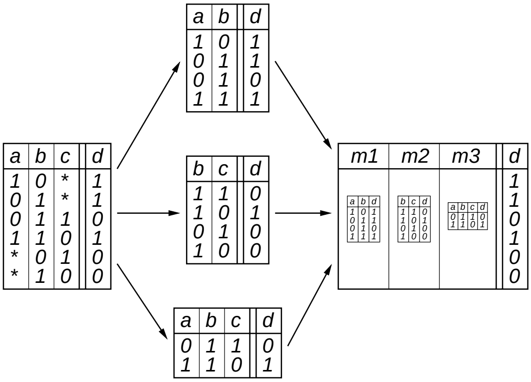
Definition 2. *Let $\mathbb{A} = (U, A \cup \{d\})$ be a decision table. Any conjunction of total descriptors $(a_{k_1} \in V_{k_1}) \wedge \ldots \wedge (a_{k_n} \in V_{k_n})$ is called a* total template*. An object $u \in U$ satisfies total template $(a_{k_1} \in V_{k_1}) \wedge \ldots \wedge (a_{k_n} \in V_{k_n})$ if values of attributes $a_{k_1}, \ldots, a_{k_n} \in A$ for the object u are not missing in $\mathbb{A}$.*

Total templates are used to discover regular areas in data that contain no missing values. Once we have a total template, we can identify it with a subtable of original data table. Such a subtable consists of attributes that are elements of total template and contains all objects that satisfy this template. With such a unique assignment of total templates and complete subtables of original data we can think of the data decomposition as a set of total templates.

3 Method Description

The decomposition method consist of two phases. In the first step the data decomposition is done. In the second step classifiers are induced and combined with a help of a conflict resolving method.

In the data decomposition phase original decision table with missing attribute values is partitioned to a number of decision subtables with complete



a	b	c	d
1	0	*	1
0	1	*	1
0	1	1	0
1	1	0	1
*	0	1	0
*	1	0	0

a	b	d
1	0	1
0	1	1
0	1	0
1	1	1

b	c	d
1	1	0
1	0	1
0	1	0
1	0	0

a	b	c	d
0	1	1	0
1	1	0	1

m1	m2	m3	d
			1
			1
			0
			1
			0
			0

Fig. 1. The incomplete data is decomposed into complete subtables. Then, a conflict resolving method is applied.

object descriptions. Such a data decomposition should be done in accordance to regularities in real-world interest domain. We expect that the decomposition could reveal patterns of missing attribute values with a similar meaning for investigated real-world problem. Ideally the complete subtables that are result of the decomposition should correspond to natural subproblems of the whole problem domain.

The considered decomposition itself becomes the problem of covering data table with templates, as investigated in [10,9]. A standard approach to cover data table with templates is to iteratively generate the best template for objects that remains uncovered. The algorithm starts from the full set of objects. Than the (sub)optimal template is generated according to chosen criterion. In our experiments we used a dedicated, effective genetic algorithm to generate a suboptimal template. All objects that satisfy the generated template are removed and the process is continued until the set of uncovered objects becomes empty. The set of templates generated by this algorithm covers all objects from original decision table. We can treat covering set of total templates as the result of decomposition.

Subsets of original decision table must meet some requirements in order to achieve good quality of inductive reasoning as well as to be applicable in case of methods that cannot deal with missing attribute values. We expect the decision subtables corresponding to templates are exhaustively covering the input table. They should contain no missing attribute values. It is obvious that the quality of inductive reasoning depends on a particular partition and some partitions are better than others. We should construct the template evaluation criteria for templates defining decision subtables relevant to the approximated concept.

Once we have data decomposed into complete decision subtables we should merge partial classifiers to one global classifier. This is the second step of the decomposition method. Answer of classifiers induced from decision subtables are combined by a conflict resolving method. In presented experiments a Quinlan's *C4.5* method was used to induce classifiers from the decision subtables. This method was chosen to be able to compare missing attribute values handling built in C4.5 with the decomposition method that does not relay on any other missing attribute values handling. The empirical evaluation provided by Grzymała-Busse in [4] and by Quinlan in [11] suggest that C4.5 has very effective mechanism for missing attribute values handling. The initial experiments showed that application of voting conflict resolving for the decomposition method is not enough to achieve good results. Partially this is a consequence of possible *positive region* [6,13] reduction in subtables of original data. Objects that are covered by small number of total templates and contain many missing values were often incorrectly classified. In experiments we use an inductive learning method to resolve conflicts. The expressiveness of regular classifier makes it possible to combine partial answers induced from inconsistent decision subtables in much more sophisticated way. To be consequent also here the C4.5 method was used for conflict resolving.

Briefly we can summarize the decomposition method as follows:

1. Create a temporary set $\mathbb{T}$ of objects being a copy of the original decision table and repeat 2-3 until the temporary set $\mathbb{T}$ is empty;
2. Generate the best total template according to chosen criterion;
3. Remove objects from the temporary set $\mathbb{T}$ that are covered by generated template;
4. Create complete decision subtables that correspond to generated set of templates;
5. Induce classifiers over complete decision subtables;
6. Induce, from answers of the classifiers based on subtables, the top classifier used as a conflict resolving method.

4 Decomposition Criteria

Common approach to measure adequateness of a template for decomposition of a particular data set is to define a function q which describes overall quality of the investigated template. Then, the best template is understood as a template with the best value of such a quality function [9]. To achieve good results we should select quality function q very carefully and in accordance to nature of the optimized problem.

A standard approach to measure template quality is to define a quality function using *width* and *height* of a template [10,9]. The *template height* is the number of objects that satisfy a template and the *template width* is the number of attributes that are elements of a template. To obtain a quality function q of a template we have to combine width and height to get one value. A usual formula that combines these two factors is

$$q = w \cdot h. \tag{1}$$

We can also add a simple mechanism to control the importance of each factor

$$q = w^{\alpha} \cdot h, \tag{2}$$

where $\alpha > 0$. If we apply $\alpha > 1$ the importance of the width, thus importance of the size of available object description, increases and the number of necessary templates to cover original decision table is higher. The empirical results showed, however, that α does not have significant impact on overall classification quality.

The quality function based only on width and height is not always enough to classify objects better than the C4.5 method with native missing attribute values handling. The empirical evaluation demonstrated that in data exist many templates with similar width and height, but with different potential for the data decomposition.

We can estimate the template quality by measuring *homogeneous degree* of indiscernibility classes [9]. Such a measure corresponds to the quality of the classification by prime implicants [6]. We measure the homogeneous degree within the indiscernibility classes

$$G = \frac{\sum_{i=1}^{K} max_{c \in V_d} card(\{y \in [x^i]_{IND} : d(y) = c\})}{K^2}, \tag{3}$$

where K is the number of indiscernibility classes $[x^1]_{IND}, \ldots, [x^K]_{IND}$. We can easily incorporate the G factor into the quality function

$$q = w \cdot h \cdot G^{\alpha}, \tag{4}$$

where α controls influence of the G factor to the whole quality value.

The second measure is similar to the *wrapper* approach in the feature selection [1]. Instead estimating the template quality we can use the predictive accuracy of the data subset. The classifier itself is executed on decision subtable determined by the total template and the number of correct answers is counted.

$$P = \frac{\textit{number of correct answers}}{\textit{number of objects}}. \tag{5}$$

Also this factor can be easily incorporated into the quality function

$$q = w \cdot h \cdot P^{\alpha}. \tag{6}$$

The predictive accuracy turned out to be applicable to the template evaluation even without width and height i.e.

$$q = P. \tag{7}$$

5 Empirical Evaluation

There were carried out some experiments in order to evaluate the decomposition method with various template evaluation functions. A genetic algorithm was used for generation of the best template with respect to the selected decomposition criterion. Results were obtained from the average of classification quality from 100 times repeated five-fold Cross-Validation (CV5) evaluation. This testing method was introduced to assure preciseness in measuring the classification

Table 1. Comparison of the decomposition method that use various template evaluation criteria with the C4.5 method.

	C4.5	$w \cdot h$	$w \cdot h \cdot G$	$w \cdot h \cdot G^8$	$w \cdot h \cdot P$	$w \cdot h \cdot P^8$	P
att	52.55 ±0.12	54.94 ±0.14	59.48 ±0.13	59.28 ±0.12	55.77 ±0.13	61.94 ±0.09	63.33 ±0.09
ban	62.14 ±0.25	65.82 ±0.15	65.57 ±0.16	65.34 ±0.17	68.51 ±0.15	74.91 ±0.14	76.30 ±0.15
cmc2	45.72 ±0.10	44.92 ±0.11	42.23 ±0.10	42.37 ±0.10	47.28 ±0.09	51.33 ±0.09	51.41 ±0.10
dna2	86.84 ±0.06	80.73 ±0.09	80.48 ±0.07	80.43 ±0.08	86.20 ±0.08	88.39 ±0.08	89.07 ±0.06
hab2	71.54 ±0.14	68.07 ±0.15	74.40 ±0.16	74.45 ±0.15	69.14 ±0.13	74.67 ±0.10	75.98 ±0.10
hep	80.12 ±0.22	75.88 ±0.27	77.14 ±0.25	77.94 ±0.25	79.53 ±0.16	85.29 ±0.16	86.59 ±0.15
hin	70.47 ±0.09	69.96 ±0.09	66.54 ±0.10	68.63 ±0.12	70.16 ±0.10	71.10 ±0.08	70.53 ±0.10
hyp	95.82 ±0.01	96.72 ±0.02	95.23 ±0.01	95.23 ±0.02	96.76 ±0.01	96.81 ±0.01	97.09 ±0.01
pid2	60.81 ±0.12	61.98 ±0.13	61.73 ±0.12	62.14 ±0.10	62.19 ±0.13	67.11 ±0.09	68.29 ±0.08
smo2	60.75 ±0.07	56.14 ±0.07	69.52 ±0.11	69.53 ±0.15	57.92 ±0.10	68.95 ±0.03	69.66 ±0.02
$\sum$		-11.6%	+5.56%	+8.58%	+6.7%	+53.74%	+61.49%

quality as well as the number of generated templates. The *C4.5* method was used as a classifier and tests were performed with different decomposition approaches as well as without using decomposition method at all. The *WEKA* software system [2], which contains re-implementation of Quinlan's C4.5 Release 8 algorithm in Java, was utilized in experiments. Data sets from *StatLib* [8] were used for evaluation of the decomposition method. Data sets contain missing values in the range from 14.1% to 89.4% of all values in data.

Table 1 presents the results of the decomposition method. In the first column there are the results of the C4.5 method. In the following columns the results of the decomposition method are presented with various template quality function described in the header of each column. The big numbers represents the average accuracy of a classification method while the small numbers represents the standard deviation of results. The sum at the bottom row corresponds to a difference of the classification accuracy in comparison to the C4.5 method.

The decomposition method performs better than the C4.5 method, especially when the predictive quality is included in the template quality function. We should consider that evaluation of the predictive quality is very time-consuming, even in spite of partial result caching and other optimizations. The homogeneous degree is much more easier to compute, however, the results not always overcome the C4.5 method.

Table 2 presents the average number of generated templates with its standard deviation. The number of templates corresponds to the number of subclassifiers being result of the data decomposition. As we can see there are no strong general correlation between the number of templates and classifier accuracy. For some data sets the better classification is related to the increase of the number of templates while for the other data sets better accuracy is achieved without any increase of the number of templates.

Table 2. Comparison of the number of subtables (templates) used in the decomposition method.

	$w \cdot h$	$w \cdot h \cdot G$	$w \cdot h \cdot G^8$	$w \cdot h \cdot P$	$w \cdot h \cdot P^8$	P
att	3.96 ±0.02	4.18 ±0.02	9.23 ±0.05	3.97 ±0.03	3.88 ±0.03	4.97 ±0.07
ban	8.92 ±0.24	7.20 ±0.05	5.67 ±0.05	8.18 ±0.05	10.62 ±0.08	23.10 ±0.18
cmc2	2.00 ±0.00	4.97 ±0.01	5.93 ±0.02	2.15 ±0.02	4.11 ±0.04	5.25 ±0.05
dna2	1.06 ±0.01	1.01 ±0.01	1.01 ±0.00	2.54 ±0.02	3.55 ±0.04	7.08 ±0.04
hab2	3.65 ±0.02	5.00 ±0.00	4.73 ±0.02	3.67 ±0.02	3.08 ±0.03	2.33 ±0.04
hep	4.02 ±0.03	3.89 ±0.02	3.61 ±0.03	4.15 ±0.02	5.50 ±0.05	8.77 ±0.10
hin	3.83 ±0.03	13.72 ±0.06	21.66 ±0.08	4.91 ±0.04	8.77 ±0.07	13.22 ±0.11
hyp	2.00 ±0.00	5.95 ±0.01	5.82 ±0.02	2.02 ±0.01	2.14 ±0.02	7.53 ±0.08
pid2	2.98 ±0.01	2.99 ±0.01	2.00 ±0.00	2.98 ±0.01	3.26 ±0.03	4.81 ±0.05
smo2	2.00 ±0.00	2.20 ±0.02	1.59 ±0.03	1.26 ±0.02	1.26 ±0.02	2.06 ±0.04

6 Conclusions

The decomposition method turned out to be an efficient tool for adapting existing methods to deal with missing attribute values in decision tables. It can be applied to various algorithms for classifier induction to enrich them with capabilities of incomplete information systems processing. The time-consuming predictive quality evaluation can be replaced now with easier to compute measures of the template quality. The further research will focus on application of rule-based inductive learning with uniform conflict resolving method at the subtables and the whole system level. We believe that decomposition done in accordance to the natural structure of analyzed data can result in classifier close to the common sense reasoning.

Acknowledgments. I wish to thank professor Andrzej Skowron for a great support while writing this paper. This work was partially supported by the Polish State Committee for Scientific Research grant No. 8T11C02519.

References

1. M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(3), 1997.
2. E. Frank, L. Trigg, and M. Hall. *Weka 3.3.2, Waikato Environment for Knowledge Analysis.* http://www.cs.waikato.ac.nz/ml/weka, The University of Waikato, Hamilton, New Zealand, 2002.
3. Y. Fujikawa and T. B. Ho. Scalable algorithms for dealing with missing values. 2001.
4. J. W. Grzymała-Busse and M. Hu. A comparison of several approaches to missing attribute values in data mining. In W. Ziarko and Y. Y. Yao, editors, *Proceedings of 2nd International Conference on Rough Sets and Current Trends in Computing, RSCTC-2000*, volume 2005 of *LNAI*, pages 180–187. Springer, 2000.

5. J. W. Grzymała-Busse and A. Y. Wang. Modified algorithms LEM1 and LEM2 for rule induction from data with missing attribute values. In *Proceedings of 5th Workshop on Rough Sets and Soft Computing (RSSC'97) at the 3rd Joint Conference on Information Sciences*, pages 69–72, Research Triangle Park (NC, USA), 1997.
6. J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron. Rough sets: A tutorial. In S. K. Pal and A. Skowron, editors, *Rough Fuzzy Hybridization. A New Trend in Decision Making*, pages 3–98, Singapore, 1999. Springer.
7. M. Kryszkiewicz. Properties of incomplete information systems in the framework of rough sets. In L. Polkowski and A. Skowron, editors, *Rough Sets in Knowledge Discovery 1: Methodology and Applications*, pages 422–450. Physica-Verlag, 1998.
8. M. Meyer and P. Vlachos. *StatLib — Data, Software and News from the Statistics Community.* http://lib.stat.cmu.edu/, Carnegie Mellon University, Pittsburgh, PA, 1998.
9. S. H. Nguyen. *Regularity Analysis and its Application in Data Mining.* PhD thesis, Warsaw University, Faculty of Mathematics, Computer Science and Mechanics, 1999.
10. S. H. Nguyen, A. Skowron, and P. Synak. Discovery of data patterns with applications to decomposition and classification problems. In L. Polkowski and A. Skowron, editors, *Rough Sets in Knowledge Discovery*, volume 2, pages 55–97, Heidelberg, 1998. Physica-Verlag.
11. J. R. Quinlan. Unknown attribute values in induction. In A. M. Segre, editor, *Proceedings of the Sixth International Machine Learning Workshop*, pages 31–37. Morgan Kaufmann, 1989.
12. J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufman, San Mateo, 1993.
13. A. Skowron and C. Rauszer. The discernibility matrices and functions in information systems. In R. Słowiński, editor, *Intelligent Decision Support. Handbook of Applications and Advances in Rough Sets Theory*, pages 331–362, Dordrecht, 1992. Kluwer.
14. J. Stefanowski and A. Tsoukiàs. Incomplete information tables and rough classification. *International Journal of Computational Intelligence*, 17(3):545–566, August 2001.