To appear in *Proc. KDD-97*

A dataset decomposition approach to data mining and machine discovery

Blaž Zupan[1], Marko Bohanec[1], Ivan Bratko[1,2], Bojan Cestnik[3,1]

[1] Institute Jožef Stefan, Jamova 39, SI-1000 Ljubljana, Slovenia
[2] Faculty of computer and information science, University of Ljubljana, Slovenia
[3] Temida, d.o.o., Ljubljana, Slovenia
blaz.zupan@ijs.si, marko.bohanec@ijs.si, ivan.bratko@fri.uni-lj.si, bojan.cestnik@ijs.si

Abstract

We present a novel data mining approach based on decomposition. In order to analyze a given dataset, the method decomposes it to a hierarchy of smaller and less complex datasets that can be analyzed independently. The method is experimentally evaluated on a real-world housing loans allocation dataset, showing that the decomposition can (1) discover meaningful intermediate concepts, (2) decompose a relatively complex dataset to datasets that are easy to analyze and comprehend, and (3) derive a classifier of high classification accuracy. We also show that human interaction has a positive effect on both the comprehensibility and classification accuracy.

Introduction

When dealing with a complex problem, a good strategy is to decompose it to less complex and more manageable subproblems. This has an obvious parallel in data analysis: instead of analyzing a complete dataset, decompose it to smaller, more manageable datasets that can be analyzed independently.

In this paper, we propose a dataset decomposition approach that is restricted to classification datasets that define a single target concept. Such datasets consist of instances (examples), each being described by a set of attributes and a class. Given an initial dataset of some target concept, the decomposition induces a definition of the target concept in terms of a hierarchy of intermediate concepts and their definitions.

The dataset decomposition method is based on function decomposition (Curtis 1962). Let a dataset E_F with attributes $X = \langle x_1, \ldots, x_n \rangle$ and class variable y partially represent a function $y = F(X)$. The goal is to decompose this function into $y = G(A, H(B))$, where A and B are subsets of attributes, and $A \cup B = X$. Functions G and H are partially represented by datasets E_G and E_H, respectively. The task is to determine E_G and E_H so that their complexity (determined by some complexity measure) is lower than that of E_F, and so that E_G and E_H are consistent with E_F. Such a decomposition also discovers a new intermediate concept $c = H(B)$. Since the decomposition can be applied recursively on E_G and E_H, the result is a hierarchy of concepts.

Central to each decomposition step is the selection of a partition of attributes X to sets A and B. We propose a method that selects this partition so that the joint complexity of the resulting E_G and E_H is minimized. Although such decomposition can be completely autonomous, the comprehensibility of the discovered concepts may be increased if the user is involved in partition selection. We refer to such an approach as supervised decomposition.

The decomposition aims at the discovery of (1) meaningful intermediate concepts, (2) useful concept hierarchy, and (3) small and manageable datasets that describe each concept in the hierarchy. The resulting datasets can be further analyzed independently, but due to reduced complexity, the analysis task is expected to be easier than that for the original dataset.

Single-step dataset decomposition

The core of the decomposition algorithm is a *single-step decomposition* which, given a dataset E_F that partially specifies a function $y = F(X)$ and a partition of attributes X to sets A and B denoted by $A|B$, decomposes F into $y = G(A, c)$ and $c = H(B)$. This is done by constructing the datasets E_G and E_H that partially specify G and H, respectively. X is a set of attributes $x_1, \ldots, x_m$, and c is a new, intermediate concept. A is called a *free set* and B a *bound set*, such that $A \cup B = X$ and $A \cap B = \emptyset$. E_G and E_H are discovered in the decomposition process and are not predefined.

Consider a dataset from Table 1 that partially describes a function $y = F(x_1, x_2, x_3)$, where x_1, x_2, and x_3 are attributes and y is the target concept. y, x_1, and x_2 can take the values `lo`, `med`, `hi`; x_3 can take the values `lo`, `hi`.

Suppose the task is to derive the datasets E_G and E_H for the attribute partition $A|B = \langle x_1 \rangle | \langle x_2, x_3 \rangle$. The dataset is first represented by a *partition matrix*, which is a tabular representation of the dataset E_F with all combinations of values of attributes in A as row labels and of B as column labels (Table 2). Partition matrix entries with no corresponding instance in E_F are denoted with "-" and treated as *don't-care*.

x_1	x_2	x_3	y
lo	lo	lo	lo
lo	lo	hi	lo
lo	med	lo	lo
lo	med	hi	med
lo	hi	lo	lo
lo	hi	hi	hi
med	med	lo	med
med	hi	lo	med
med	hi	hi	hi
hi	lo	lo	hi
hi	hi	lo	hi

Table 1: Set of instances that partially describe the function $y = F(x_1, x_2, x_3)$.

Each column in the partition matrix denotes the behavior of F when the attributes in the bound set are constant. Columns that exhibit the same behavior, i.e., have pairwise equal row entries or at least one row entry is don't-care, are called *compatible* and can be labeled with the same value of c. The decomposition aims at deriving the new intermediate concept variable c with the smallest set of values, i.e., finding the proper labeling of partition matrix columns using the smallest set of labels. The problem is formulated as a graph coloring problem and solved by a polynomial heuristic method (Perkowski et al. 1995).

x_2 / x_1 x_3	lo / lo	lo / hi	med / lo	med / hi	hi / lo	hi / hi
lo	lo	lo	lo	med	lo	hi
med	-	-	med	-	med	hi
hi	hi	-	-	-	hi	-
c	1	1	1	2	1	3

Table 2: Partition matrix with column labels (c) for the partition $\langle x_1 \rangle | \langle x_2, x_3 \rangle$ and dataset from Table 1.

From the labeled partition matrix, it is easy to derive new datasets E_G and E_H. For E_H, the attribute set is B. Each column in partition matrix provides an instance in dataset E_H whose class equals to the column label. E_G is derived as follows. For any value of c and combination of values of attributes in A, $y = G(A, c)$ is determined by finding an instance $e_i \in E_F$ in a corresponding row and in any column labeled with the value of c. If such an entry exists, an instance with attribute set $A \cup \{c\}$ and class $y = F(e_i)$ is included in E_G.

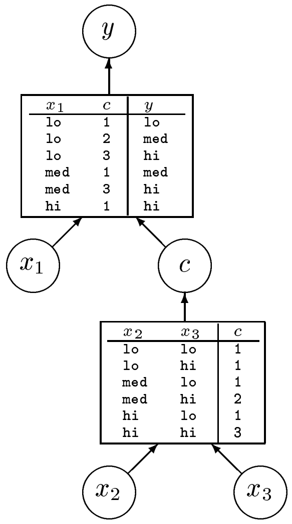

Figure 1: Decomposition of the dataset from Table 1.

Figure 1 shows E_G and E_H of our example dataset. Note that the new datasets are less complex than the original one, and are furthermore much easier to interpret: c corresponds to $\mathrm{MIN}(x_2, x_3)$ and y to $\mathrm{MAX}(x_1, c)$.

Single-step decomposition can also detect redundant attributes. Let an initial set of attributes X be partitioned to $B = \langle x_j \rangle$ and $A = X \setminus \langle x_j \rangle$. If $\nu(A|B) = 1$, then the corresponding function $c = H(x_j)$ is constant, and x_j can be removed from the dataset.

Overall decomposition method

Given a dataset E_F that partially defines a function $y = F(X)$, where $X = \langle x_1, \ldots, x_n \rangle$, it is important to identify an appropriate attribute partition $A|B$ in the single step decomposition. The partition selection can affect both the complexity and comprehensibility of the resulting datasets. In (Zupan *et al.* 1997), the authors proposed different *partition selection measures* of which in this paper we mention and use only the simplest one: partition matrix column multiplicity $\nu(A|B)$. Thus, the decomposition favors the partitions which yield the intermediate concepts with the smallest value sets. To limit the time complexity of the method, only the partitions having a few attributes in the bound set are considered by the algorithm.

In this paper, we advocate for the interaction of the user throughout the decomposition process. Given the initial dataset, all candidate partitions are examined and those with the best partition selection measure are presented to the user. The user selects the most favorable partition, which is used for decomposition. To further engage the user, we let him decide whether to decompose a dataset or leave it as it is. Because of user's involvement we refer to such a process to as *supervised decomposition*. Compared to unsupervised decomposition (Zupan *et al.* 1997), we expect a positive effect on comprehensibility.

The described method is implemented as a system called HINT (Hierarchy INduction Tool). The system runs on common UNIX platforms.

Case study: housing loans allocation

The method was experimentally evaluated on a real-world dataset taken from a management decision support system for allocating housing loans (Bohanec, Cestnik, & Rajkovič 1996). This system was developed for the Housing Fund of the Republic of Slovenia and used since 1991 in 13 floats of loans.

In each float, the basic problem is to allocate the available funds to applicants. Typically, there are several thousands of applicants and their requested amount exceeds the available resources. Therefore, the applicants must be ranked in a priority order in accordance with the criteria prescribed in the tender. Each applicant is ranked into one of five priority classes. The criteria include: applicant's *housing* conditions, current *status*, and his *social* and *health* conditions.

The evaluation of loan priority is carried out by a hierarchical concept model (Figure 2). For each internal concept in the structure, there is a decision rule that determines the aggregation of concepts. Both the structure and the rules were developed manually by experts using a multi-attribute decision making shell DEX (Bohanec & Rajkovič 1990).

For the evaluation of the decomposition method, we took applicants' data from one of the floats carried out in 1994. There were 1932 applicants in that float. Each data record contained 12 two to five-valued attributes. Due to the discreteness of attributes, the 1932 records provided 722 unique dataset instances. These instances covered only 3.7% of the complete attribute space. Each instance was classified using the original evaluation model and the resulting unstructured dataset was analyzed by the decomposition.

First, the attributes were tested for redundancy. The attributes `cult_hist` and `fin_sources` were found redundant and removed from the dataset. These two attributes may affect the priority under some special circumstances, e.g., house is a cultural monument or the applicant has granted additional financial sources. These were not present in the dataset.

The resulting dataset was examined for decomposition. All possible partitions with bound sets of 2

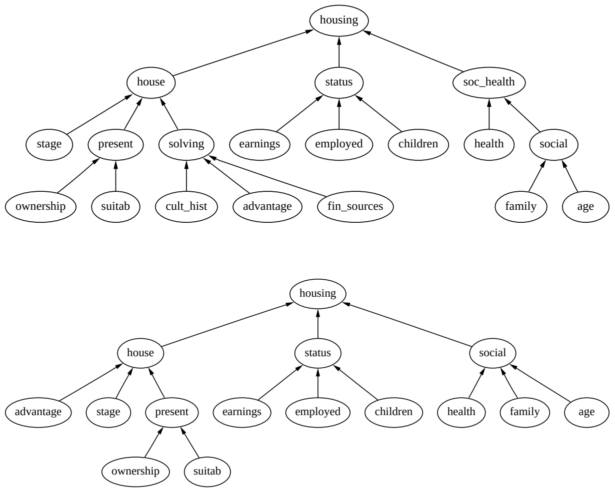

Figure 2: Original (top) and discovered (bottom) concept hierarchy for housing loans allocation.

or 3 attributes were examined. From these, according to partition selection measure (column multiplicity ν), HINT proposed only the best candidates with $\nu = 3$. Among 120 possible bound sets of 3 attributes, there were 11 bound sets that minimized ν. Among these, the domain expert chose the bound set ⟨`earnings,employed,children`⟩ as the most favorable as it constituted a comprehensible intermediate concept of applicants' current `status`. The decomposition process was continued similarly, resulting in intermediate concepts `social` (social and health condition of the applicant), `present` (suitability of applicant's present housing) and `house` (overall housing conditions). Figure 2 shows the resulting concept structure. Apart from the two missing redundant attributes it is very similar to the structure actually used in the management decision support system. We consider this similarity of concept structures as a significant indicator of success of our method.

Next, the resulting datasets were examined. These are considerably less complex than the initial one: while the initial dataset contained 722 instances, the most complex resulting dataset (`housing`) has only 38 instances while all other datasets include less than 20 instances. In total, the resulting decomposed datasets include only 108 instances. In addition, the decomposed datasets use significantly less attributes. It was observed that all the datasets were comprehensible and consistent with the expert's expectations.

To assess the benefit of user's interaction, we used HINT in unsupervised mode that automatically discovered the concept structure. Assessed by the expert, it was found that some less intuitive intermediate concepts had been developed. For example, the decomposition combined `employed` and `advantage`, which is difficult to interpret as a useful concept.

The generalization quality of decomposition was assessed by 10-fold cross validation: the initial dataset was split to 10 subsets, and 10 experiments were performed taking a single subset as a test set and the instances in the remaining subsets as a training set. HINT used either the structure as developed in supervised mode, or was run in the unsupervised mode on the training sets. The classification accuracies were 97.8% and 94.7%, respectively. For comparison, we used C4.5 decision tree induction tool (Quinlan 1993) and obtained the accuracy of 88.9%. These results clearly indicate that for this dataset the decomposition outperformed C4.5. It is further evident that the supervised method resulted in a classifier that was superior to that developed without user's interaction.

Related work

The proposed decomposition method is based on the function decomposition approach to the design of digital circuits (Curtis 1962). The approach was recently advanced by research groups of Perkowski, Luba, and Ross (Perkowski et al. 1995; Luba 1995; Ross *et al.* 1994). Given a Boolean function partially specified by a truth table, their methods aim to derive switching circuits of low complexity.

Within machine learning, an approach that relies on a given concept structure but learns the corresponding functions from the training sets is known as structured induction (Michie 1995). Its advantages are comprehensibility and high classification accuracy.

The method presented in this paper shares the motivation with structured induction, while the core of the method is based on boolean function decomposition. In comparison with related work, the present paper is original in the following aspects: new method for handling multi-valued attributes, supervised decomposition, paying strong attention to discovery of meaningful concept hierarchies, and experimental evaluation on a data-mining problem.

Conclusion

A new data analysis method based on dataset decomposition is proposed. The method is restricted to classification datasets that define a single target concept and develops its description in terms of a hierarchy of intermediate concepts and their definitions. In this way, we obtain datasets that are less complex than the initial dataset and are potentially easier to interpret.

We have assessed the applicability of the approach in the analysis of non-trivial housing loans allocation dataset. The method was able to assist in the discovery of the concepts structure very similar to the one that is actually used for the evaluation of housing loans applications in practice. The decomposition resulted in datasets that were significantly less complex than the initial one, and represented meaningful concepts. The total size in terms of data elements of the decomposed dataset tables was only 4.45% of the size of the original dataset table. It was further shown that the decomposition is a good generalizer and for our dataset outperformed a state-of-the-art induction tool C4.5.

The decomposition approach as presented in this paper is limited to consistent datasets with discrete attributes and classes. However, recently developed noise and uncertainty handling mechanisms and an approach to handle continuously-valued datasets (Demšar *et al.* 1997) facilitate more general data-analysis tasks that are planned for the future. Another interesting issue for further work is to extend the approach to handle non-classification datasets. Possible applications of this type include data-base restructuring and discovery of functional dependencies.

References

Bohanec, M., and Rajkovič, V. 1990. DEX: An expert system shell for decision support. *Sistemica* 1(1):145–157.

Bohanec, M.; Cestnik, B.; and Rajkovič, V. 1996. A management decision support system for allocating housing loans. In Humphreys, P., et al., eds., *Implementing System for Supporting Management Decisions.* Chapman & Hall. 34–43.

Curtis, H. A. 1962. *A New Approach to the Design of Switching Functions.* Van Nostrand, Princeton, N.J.

Demšar, J.; Zupan, B.; Bohanec, M.; and Bratko, I. 1997. Constructing intermediate concepts by decomposition of real functions. In *Proc. European Conference on Machine Learning*, 93–107. Springer.

Luba, T. 1995. Decomposition of multiple-valued functions. In *Int. Symp. on Multiple-Valued Logic*, 256–261.

Michie, D. 1995. Problem decomposition and the learning of skills. In Lavrač, N., and Wrobel, S., eds., *Machine Learning: ECML-95*, Notes in Artificial Intelligence 912. Springer-Verlag. 17–31.

Perkowski, M. A., et al. 1995. Unified approach to functional decompositions of switching functions. Technical report, Warsaw University of Technology and Eindhoven University of Technology.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers.

Ross, T. D.; Noviskey, M. J.; Gadd, D. A.; and Goldman, J. A. 1994. Pattern theoretic feature extraction and constructive induction. In *Proc. ML-COLT '94 Workshop on Constructive Induction and Change of Representation.*

Zupan, B.; Bohanec, M.; Bratko, I.; and Demšar, J. 1997. Machine learning by function decomposition. In *Proc. 14th International Conference on Machine Learning.*