A function-decomposition method for development of hierarchical multi-attribute decision models

Marko Bohanec (marko.bohanec@ijs.si)
Jožef Stefan Institute, Ljubljana, Slovenia
Blaž Zupan (blaz.zupan@ijs.si)
Jožef Stefan Institute
Ivan Bratko (ivan.bratko@ninurta.fri.uni-lj.si)
Faculty of Computer and Information Sciences, Ljubljana, and Jožef Stefan Institute
Bojan Cestnik (bojan.cestnik@ijs.si)
Temida, d.o.o., Ljubljana, and Jožef Stefan Institute

Summary

We present a novel method for the development of hierarchical multi-attribute decision models from a given unstructured set of decision examples. The method develops a hierarchical structure by discovering new aggregate attributes and their descriptions. Each new aggregate attribute is described by an example set whose complexity is lower than the complexity of the initial set. The method is based on function decomposition. Models can be developed either with or without human interaction. The method is experimentally evaluated on a real-world housing loans allocation problem. This case study shows that the decomposition can discover a meaningful and transparent decision model of high classification accuracy. We show that human assistance has a positive effect on both the comprehensibility and classification accuracy.

Keywords: multi-attribute decision making, hierarchical models, function decomposition, discovery, data-driven modeling, data-mining

1 Introduction

In decision support systems, we use models to predict the outcome of decision choices we might make (Mallach 94). For multi-attribute decision problems, i.e., decision-making situations in which the alternatives are described by several attributes that cannot be optimized simultaneously, we develop multi-attribute decision models. Most commonly, such models are developed in a hierarchical fashion, starting from some general but imprecise goal statement, which is gradually refined into more precise sub and sub-sub goals (Stewart 92). A typical example of this approach is Saaty's (93) Analytic Hierarchy Process.

The development of hierarchical multi-attribute decision models is difficult, especially when the decision problem itself is difficult and involves several tens of attributes. In most cases, the models are developed manually in a tiresome and lengthy process in which the designers (decision analysts, decision makers, experts, knowledge engineers) use their knowledge about the problem and employ their skill and experience. On the other hand, the computers nowadays provide relatively inexpensive and available means to collect data, and there is a growing volume of data about decisions already made. This data may contain useful information for decision support, discovery of underlying principles, and different analysis tasks.

In this paper we propose a method that develops a hierarchical multi-attribute decision model using decision *examples* that may be taken either from an existing database of past decisions, or provided explicitly by the decision-maker. Each example is described by a set of attributes and its utility. The method is restricted to decision problems with nominal attributes and utility. Given an initial set of examples, the method develops a corresponding model in terms of a hierarchy of attributes and their definitions. The development proceeds either with or without human interaction.

The proposed method is based on *function decomposition*, an approach originally developed for the design of digital circuits (Ashenhurst 52; Curtis 62). Let a set of decision examples E_F with *attributes* $X=\langle x_1,\ldots,x_n\rangle$ and *utility* variable y partially represent a utility 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 such that $A\cup B=X$, and functions G and H are partially represented by sets of examples E_G and E_H, respectively. The task of decomposition is to determine E_G and E_H so that their complexity (determined by some complexity measure) is, if possible, 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 *aggregate attribute* (hereafter referred to as *concept*) $c=H(B)$. Since the decomposition can be applied recursively on E_G and E_H, the result in general is a *hierarchy* consisting of attributes (terminal nodes) and concepts (internal nodes). For each concept in the hierarchy, there is a corresponding set of examples (such as E_H) that describes the dependency of that concept on its immediate descendants in the hierarchy.

Central to each decomposition step is the selection of a partition of attributes X to sets A and B. This is guided by a *partition selection measure* that assesses the joint complexity of the resulting E_G and E_H. The decomposition selects the partition that minimizes this measure. Although such decomposition can be completely autonomous, the comprehensibility of the discovered structure may be improved if the user is involved in the partition selection process: few best partitions are presented to the user, who selects the best candidate and assigns a label to the new concept. We refer to such an approach as *supervised decomposition*.

The decomposition aims at the discovery of (1) decision model hierarchy, (2) meaningful concepts, and (3) small and manageable sets of examples that describe each concept in the model. The decomposition method also has a generalization property, and the obtained hierarchical model can be used to classify (evaluate) new alternatives. When used in the supervised mode, the method assists the decision-maker in recognizing and organizing the concepts embedded in data, so it can also be regarded as a data-mining tool.

The paper is organized as follows. Section 2 describes the decomposition method. In section 3, the method is experimentally evaluated on a real-world problem of housing loans allocation. The issues specifically addressed are: comprehensibility, the benefit of user's interaction in the decomposition process, and classification accuracy of the developed model. Section 4 overviews the related work. The paper is concluded by a summary and possible directions of further work.

2 Decomposition method

This section describes the decomposition method. First, a single-step decomposition is presented which, given a set of decision examples E_F, decomposes it to consistent sets E_G and E_H. This is followed by the description of overall decomposition algorithm that, given an initial set of examples, iteratively applies the single-step decomposition to derive a hierarchy of concepts. Each concept in the hierarchy is described by its own set of examples. In each iteration, the decomposition algorithm also deals with the problem of attribute selection.

2.1 Single-step decomposition

The core of the decomposition algorithm is a *single-step decomposition* which, given a set of examples E_F that partially specifies a function $y=F(X)$, and a partition of attributes X to sets A and B (denoted $A|B$), decomposes F into $y=G(A,c)$ and $c=H(B)$. This is done by constructing sets of examples 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 concept. The partition $A|B$ is composed of a *free* set A and *bound* set B such that $A\cup B=X$ and $A\cap B=\emptyset$. E_G and E_H are developed in the decomposition process and are not predefined in any way.

Let us describe the single-step decomposition by an example. Suppose there is a set E_F (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. The variables y, x_1, and x_2 can take the values `lo`, `med`, `hi`, and x_3 can take the values `lo`, `hi`.

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

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

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

Suppose that we want to derive E_G and E_H for the attribute partition $A|B=\langle x_1\rangle|\langle x_2,x_3\rangle$. For this purpose, the initial set of examples is first represented by a *partition matrix*, which is a tabular representation of E_F with all combinations of values of attributes in A as row labels and of B as column labels. Each example from E_F has its corresponding entry in the matrix. Partition matrix entries with no corresponding example in E_F are denoted by '-' and treated as a *don't-care*. For our example set (Table 1) and the above attribute partition, the partition matrix is given in Table 2.

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 a don't-care, are called *compatible* and can be labeled with the same value of c. For instance, the first two columns in Table 2 are compatible: their entries in the first row are equal, and at least one entry is '-' in the remaining rows.

The single-step decomposition aims at deriving the new concept variable c having the smallest set of possible values. For that purpose, we construct an *incompatibility graph* whose vertices correspond to partition matrix columns. Two vertices are connected if the corresponding columns are incompatible. To find the values of c, the incompatibility graph is colored: the coloring will assign different labels to the vertices that represent mutually incompatible columns, while the compatible columns may share the same color.

An optimal coloring identifies the minimal number of groups of mutually compatible columns. This number is called *column multiplicity* and denoted by $\nu(A|B)$. Column multiplicity equals to the lowest number of possible values to be used for the new concept variable c. Since optimal graph coloring is an NP-hard problem, we use a heuristic method of polynomial complexity (Perkowski, et al. 95). For our example partition matrix, the incompatibility graph is given in Figure 1. Note that three colors are required to color this graph.

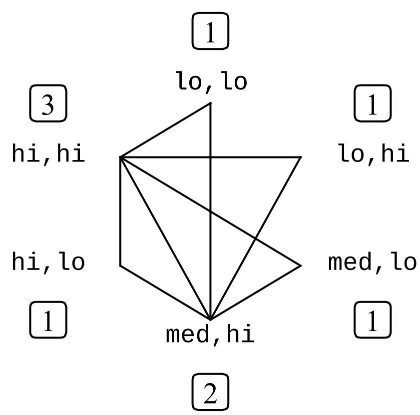

Figure 1: Incompatibility graph with assigned colors (labels) for the partition matrix from Table 2.

After the coloring of the incompatibility graph, each column of the partition matrix is assigned a label, which corresponds to an abstract value of the new concept variable c. From such an annotated partition matrix, the new sets E_G and E_H can be derived. For E_H, the attribute set is B. Each column in partition matrix is an example in E_H . The label (color) of the column becomes the class value of that example.

E_G is derived as follows. For each value of c and combination of values of attributes in A, $y=G(A,c)$ is determined by looking for an example $e_i \in E_F$ in the corresponding row and in any column labeled with the value of c. If such an example exists, it is included in E_G using the attributes $A \cup \{c\}$ and class $y=F(e_i)$.

Figure 2a shows E_G and E_H for the decomposition of Table 2. Note that the new sets are less complex than the initial set E_F, and are thus much easier to interpret: it is easy to see that c corresponds to $\min(x_2,x_3)$ and y to $\max(x_1,c)$. The corresponding interpretation of the three possible values of c is: `1=lo`, `2=med`, `3=hi`.

There exist two other non-trivial partitions of the same attribute set: $\langle x_2\rangle|\langle x_1,x_3\rangle$ and $\langle x_3\rangle|\langle x_1,x_2\rangle$. The corresponding decompositions are shown in Figures 2b and 2c, respectively. Compared to the decomposition using the partition $\langle x_1\rangle|\langle x_2,x_3\rangle$, they result in overall larger sets E_G and E_H and introduce intermediate concepts with more values (4 and 5, respectively, instead of 3). Moreover, the resulting sets are harder to interpret. Among the three attribute partitions it is therefore preferable to decide for the first one.

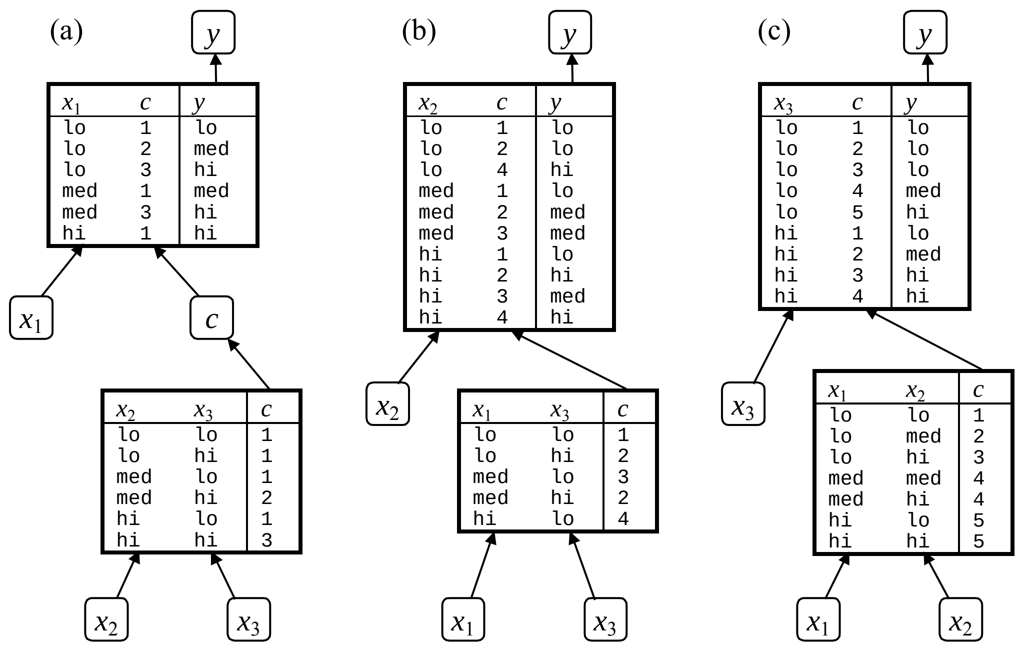

Figure 2: Three different decompositions of Table 1.

The decomposition algorithm has a generalization property. Each undefined entry (‘-’) of the partition matrix in row a and column b is generalized if there exists a corresponding example e_i that belongs to the same row a and to a column labeled with the same label as column b. For example, the entry in row `hi` and column ⟨`lo,hi`⟩ in Table 2 is generalized to `hi` because the column ⟨`lo,hi`⟩ has the same label as columns ⟨`lo, lo`⟩ and ⟨`hi, lo`⟩.

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

2.2 Overall decomposition method

Given a set of examples E_F that partially defines a utility function $y=F(X)$, where $X=\langle x_1,\ldots,x_n\rangle$, it is particularly important to find an appropriate attribute partition $A|B$ of the set X for the single-step decomposition. As illustrated in the previous section, the partition selection can affect both the complexity and comprehensibility of the resulting example sets. Zupan (97) proposed several *partition selection measures* of which in this paper we mention and use only the simplest one: column multiplicity $\nu(A|B)$. In this case, the decomposition method favors the attribute partitions that yield the intermediate concepts with the smallest sets of possible values. For our example, this criterion prefers the decomposition from Figure 2a, which indeed results in the simplest and most comprehensible description of the target concept.

To limit the time complexity of the method, we propose to consider only the partitions with small bound sets. In the case study presented in section 3, partitions with only two or three attributes in the bound sets were investigated. Such a constraint results in intermediate concepts with only a few attributes, which may have a positive effect on comprehensibility and the size of generated sets of examples.

In this paper we advocate for the interaction of the user throughout the decomposition process. Given an initial set of examples, all candidate partitions are examined and those with the best partition selection measure are presented to the user. The user then decides for the most favorable partition, i.e., the partition whose bound set consists of inter-related attributes that can constitute a meaningful intermediate concept. The selected partition is used by the single-step decomposition to derive two new sets of lower complexity. These two sets are then further investigated for decomposition. To further engage the user in the decomposition process, we let them decide whether or not to decompose a given set of examples.

Because of user's involvement in partition selection and selection of the sets to decompose, we refer to such a process to as *supervised decomposition*. Compared to *unsupervised* decomposition (Zupan, et al. 97), we expect better comprehensibility, especially in the cases when the initial examples sparsely cover the attribute space.

2.3 Implementation

The decomposition method is implemented in the C language as a system called HINT (Hierarchy INduction Tool). The system runs on several UNIX platforms, including HP-UX, SGI Iris, and SunOS. The definition of domain names and examples, and the guidance of the decomposition is managed by a script language.

3 Experimental evaluation: housing loans allocation

Experimental evaluation of the method was carried out using a real-world database used in a management decision support system for allocating housing loans (Bohanec, et al. 96). This system was developed for the Housing Fund of the Republic of Slovenia and used since 1991 in 13 floats of loans with a total value of approximately 90 million ECU.

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 financial resources. Therefore, the applicants must be ranked in a priority order for the distribution of resources in accordance with the criteria prescribed in the tender. Each applicant is ranked into one of five priority classes. The criteria may vary from tender to tender, but typically include:

1. applicant's *housing* conditions in terms of the ownership and suitability of present housing, the way of solving their problem, and the stage of solving;
2. applicant's *status* in terms of earnings, employment and the number of children;
3. *social* and *health* conditions.

In the system, the evaluation of loan priority is carried out by a hierarchical multi-attribute model whose structure is presented in Figure 3. For each internal concept in the structure, there is a decision rule defined for the aggregation of concepts. Both the structure and the rules were developed manually by the experts using a multi-attribute decision making shell DEX (Bohanec, Rajkovič 90).

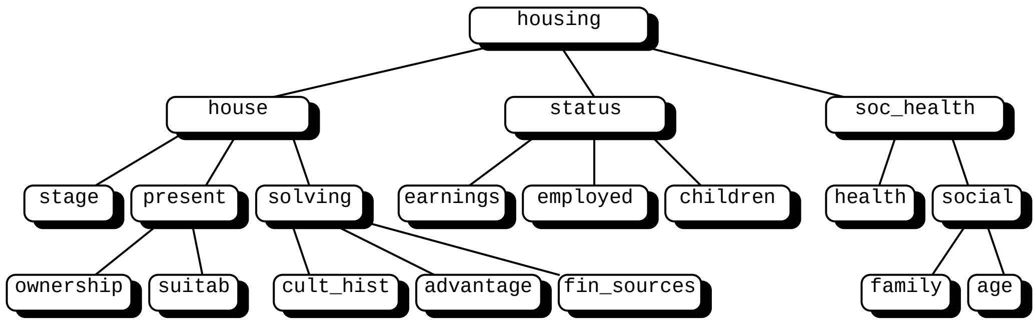

Figure 3: Original model structure for housing loans allocation.

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. In addition to some general data, such as the name and address of the applicant, each data record contained 12 two to five-valued attributes that were essential for the determination of loan priority. Due to the discreteness of attributes, the 1932 records provided 722 unique examples. These examples cover only 3.7% of the complete attribute space.

The primary goal of the experimental evaluation was to try to *reconstruct* the original decision model using only the available applicants' data, supplemented by the already known decisions about their loan priority. For this purpose, each example was classified by the original evaluation model and the resulting unstructured database was submitted to the decomposition method. Both unsupervised and supervised decompositions were carried out. The resulting models were interpreted, compared to the original one, and analyzed in terms of comprehensibility and classification accuracy. Finally, the generalization quality of the method was assessed by a cross-validation method.

3.1 Supervised decomposition

In the first stage of the analysis, the attributes were tested for redundancy. The attributes `cult_hist` and `fin_sources` were found redundant and removed from the database. The reason for redundancy is that these two attributes affect the loan priority only under some very special circumstances, which did not occur in the database. For example, `cult_hist` applies only to renewing a house that is a cultural or historical monument, and there were no such houses in that float.

The resulting set of examples was examined for decomposition. All possible partitions with bound sets of 2 or 3 attributes were examined. From these, according to the partition selection measure (column multiplicity ν), HINT proposed only the best candidates with ν=3. Among the 120 possible bound sets of 3 attributes, there were 11 bound sets that minimized column multiplicity:

```
suitab    advantage employed     earnings employed family
advantage stage     employed     earnings children health
advantage employed  health       employed children health
advantage employed  family       employed health   family
earnings  employed  children     employed health   age
earnings  employed  health
```

Among these, it was considered by the domain expert that the underlined bound set is the most favorable as it constitutes the comprehensible intermediate concept of applicants' current `status`. The resulting concept structure is given in Figure 4b.

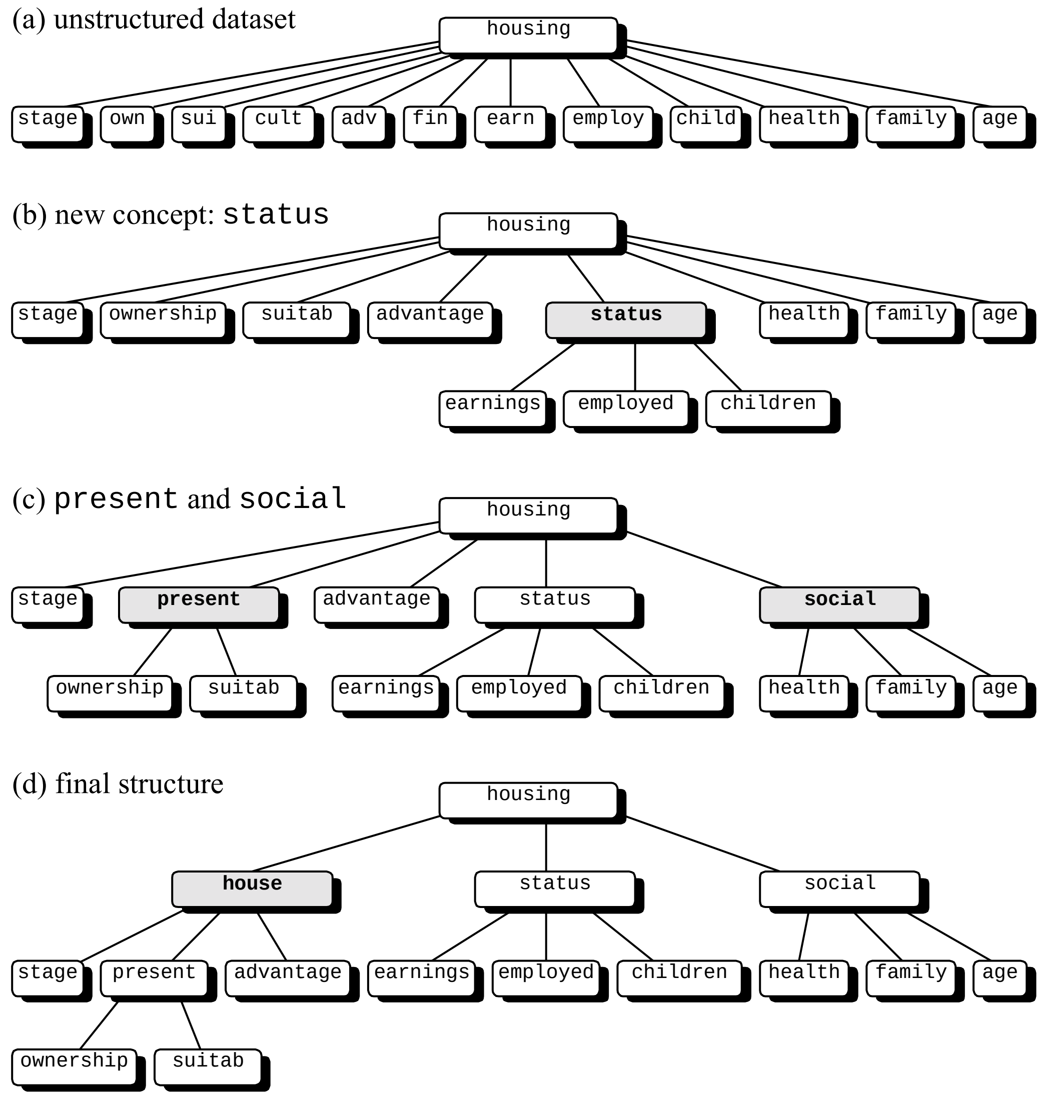

Figure 4: Evolving concept structure for housing allocation decision model.

Next, the new set of examples describing the concept `housing` was examined. In a similar fashion, three best candidate bound sets were considered among the total of 56 possible partitions:

ownership	suitab	advantage
suitab	advantage	stage
health	family	age

Again, the most favorable bound set is underlined, which was recognized as social and health condition of the applicant and formed a new 4-valued intermediate concept `social`.

The decomposition process continued similarly resulting in intermediate concepts `present` (suitability of applicant's present housing) and `house` (overall housing conditions). Overall, the process resulted in a concept structure presented in Figure 4d. Apart from the two excluded redundant attributes, the resulting concept structure is very similar to the structure actually used in the management decision support system to determine loan priority. Thus, when evaluated by the domain expert, the reconstruction was considered successful.

In each decomposition step, the selection of the most favorable bound set may not be straightforward. In our case, this was especially true for the first decomposition step, where the number of candidate partitions was quite high. The technique we employed was a gradual elimination of less favorable partitions. However, in the following decomposition steps the number of candidate bound sets was substantially lower, which made the selection easier. The decreased number of candidates was due to the lower number of attributes and better coverage of the attribute space.

3.2 Unsupervised decomposition

To assess the benefit of user's interaction in the decomposition process, we used HINT in unsupervised mode that automatically discovered the concept structure (Figure 5). When this was assessed by the expert, it was found that in addition to subtrees that were identical or similar to the ones in the original model, some less intuitive intermediate concepts were developed. For example, the unsupervised decomposition combined `employed` and `health` into `c2`, which was found difficult to be interpreted as a useful concept. Therefore, the overall solution was not as satisfactory as the one obtained by supervised decomposition.

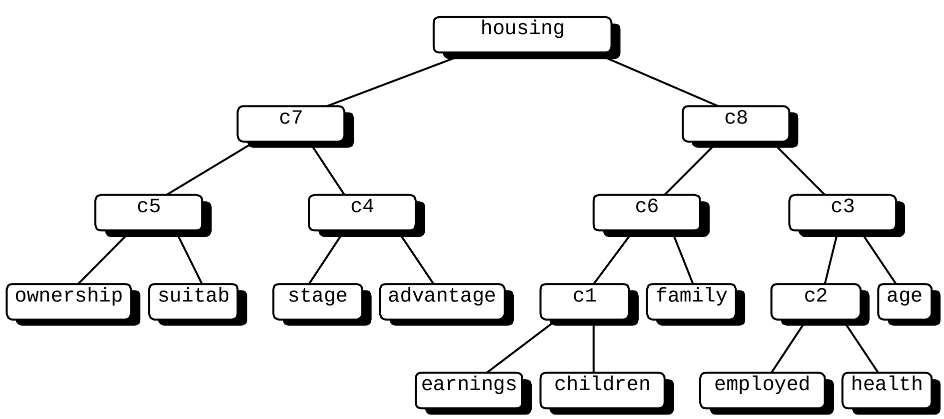

Figure 5: Model structure developed by unsupervised decomposition.

3.3 Interpretation of models

The sets of examples that describe the concepts in the resulting structure are considerably less complex than the initial one: while the initial set contains 722 examples, the most complex resulting set (`housing`) has only 38 examples, and all the remaining sets include less than 20 examples. In total, the resulting decomposed sets include only 108 examples, which is a considerable reduction in comparison with the initial set. In addition, the decomposed sets use significantly less attributes.

In the interpretation of the resulting sets it was observed that all were quite comprehensible. For example, even from the raw set it was easy to see that `status` depends monotonically on `earnings`, `employed`, and `children`. An even better interpretation was provided by a set of tools within DEX (Bohanec, Rajkovič 90), which include decision rule induction methods and visualization tools (Rajkovič, Bohanec 91). For all sets of examples it was found out that they relevantly and consistently with the expert's expectations represent the discovered concepts.

3.4 Cross-validation

The generalization quality of decomposition was assessed by 10-fold cross validation. The initial set of examples was split to 10 subsets, and 10 experiments were performed taking a single subset as a test set and the examples in the remaining subsets as a training set. HINT used either the structure as developed above (Figure 4d) or was used in the unsupervised mode on the training sets (i.e., 90% of original data). For comparison, we have also used C4.5, a machine learning program that induces decision trees from examples (Quinlan 93) and is considered a state-of-the-art generalization tool in machine learning. Note, however, that C4.5 does not develop hierarchical decision models. Rather, it uses a different representation (decision trees) and does not explicitly develop new concepts.

The obtained classification accuracies (Table 3) clearly indicate that for this problem 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. However, it has to be noted that the testing domain is, due to its original hierarchical structure, biased towards HINT.

Table 3: Classification accuracies in % by 10-fold cross validation.

HINT (developed structure)	HINT (unsupervised)	C4.5
97.8 ± 1.8	94.7 ± 2.5	88.9 ± 3.9

4 Related work

The problem of criteria identification and their structuring in terms of a decision model is central to multi-attribute decision making (Keeney, Raiffa 76), decision analysis (Phillips 86), and related fields. The research reported here was motivated by a practical need for a method that would automate and/or assist the decision-maker in developing a multi-attribute model from decision examples. The representation of decision models developed by the proposed method closely resembles that used in a multi-attribute decision support expert system shell DEX (Bohanec, Rajkovič 90).

The decomposition method is based on the function decomposition approach to the design of digital circuits by Ashenhurst (52) and Curtis (62). Their approach was recently advanced by Perkowski, et al. (95), Luba (95), and Ross, et al. (94). Given a

Boolean function partially specified by a truth table, their methods aim to derive switching circuits of low complexity.

The problem of developing hierarchical models from examples has been also studied within machine learning. There, the decomposition approach was first used by Samuel (67) in checkers playing programs. His methods relied on a given concept structure but learned the corresponding functions from the training sets. Another technique that uses a predefined structure, known as structured induction (Michie 95), was independently developed by Shapiro (87) and originally used for the classification of a fairly complex chess endgame. It was shown that the obtained solutions were both comprehensible and of high classification accuracy.

The method presented in this paper is thus closely related to three primary research areas: it shares the motivation with multi-attribute decision making and structured induction, while the core of the method is based on Ashenhurst-Curtis function decomposition. In comparison with related work, this paper is original in the following aspects: adaptation of the function decomposition approach to the development of multi-attribute decision models, new method for handling multi-valued attributes, supervised decomposition, emphasis on generalization effects of decomposition, paying strong attention to the discovery of meaningful concepts, and experimental evaluation on a real-world decision problem.

5 Conclusion

A novel method for the development of hierarchical multi-attribute decision models is proposed. Using an unstructured set of decision examples, the method develops a hierarchical structure of concepts and their definitions. The resulting model generalizes the decision examples and can serve for the evaluation of new alternatives. The method is implemented in a system called HINT.

The development of models can be carried out either autonomously or in the interaction with the decision-maker. In the latter case, the method turns into a data-mining tool for data structuring and analysis: the decomposition assists in the identification of concepts, organizing them into a hierarchy and deriving the concept representations by example sets. In this process, the original data set is decomposed into a number of less complex data sets that are easier to interpret and analyze.

We have assessed the applicability of the approach in a real-world housing loans allocation problem. It was demonstrated that the method was able to reconstruct almost completely the “right” decision model that was available for this problem. The reconstruction was carried out using 722 distinct decision examples taken from a database of applicants. Although these examples sparsely covered the attribute space, the method succeeded in deriving a model of high comprehensibility and classification accuracy. The comparison of models developed by supervised and unsupervised decomposition revealed that human assistance had a positive effect on both the comprehensibility and classification accuracy. It was further shown that the decomposition is a good generalizer and for this problem outperformed a state-of-the-art induction tool C4.5.

The decomposition approach as presented in this paper is limited to consistent sets of examples using discrete attributes and utility. However, recently developed noise and uncertainty handling mechanisms (Zupan 97), and an approach to handle continuously-valued attributes (Demšar, et al. 97) will enable HINT to be used in more general model developing tasks that are planned for the future.

Another important issue for further work is related to the interpretation of derived example sets. Currently, each concept in the developed decision model is defined by a set of examples, whose investigation and interpretation is left to the decision-maker. Unless the set is particularly small, such an interpretation may be difficult, resulting in a less comprehensible decision model. Fortunately, there is a number of existing methods that could be used to assist in the interpretation, for example regression, machine learning and visualization. We will attempt to extend the proposed approach by a systematic selection of such interpretation methods and correspondingly enhance HINT's capabilities.

6 References

R. L. Ashenhurst (1952). The decomposition of switching functions. Technical report, Bell Laboratories BL-1(11), p. 541–602.

M. Bohanec, B. Cestnik, and V. Rajkovič (1996). A management decision support system for allocating housing loans. In P. Humpreys, L. Bannon, A. McCosh, and P. Migliarese, editors, *Implementing System for Supporting Management Decisions*, p. 34–43. Chapman & Hall, London.

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

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

J. Demšar, B. Zupan, M. Bohanec, and I. Bratko (1997). Constructing intermediate concepts by decomposition of real functions. In M. van Someren and G. Widmer, editors, *Machine Learning*: ECML-97, p. 93–107. Springer-Verlag.

R. L. Keeney and H. Raiffa (1976). *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley.

T. Luba (1995). Decomposition of multiple-valued functions. In *25th Intl. Symposium on Multiple-Valued Logic*, p. 256–261, Bloomigton, Indiana.

E. G. Mallach (1994). *Understanding Decision Support Systems and Expert Systems*. Irwin.

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

A. Perkowski, T. Luba, S. Grygiel, P. Burkey, M. Burns, N. Iliev, M. Kolsteren, R. Lisanke, R. Malvi, Z. Wang, H. Wu, F. Yang, S. Zhou, and J. S. Zhang (1995). Unified approach to functional decompositions of switching functions. Technical report, Warsaw University of Technology and Eindhoven University of Technology.

L. D. Phillips (1986). Decision analysis and its applications in industry. In G. Mitra, editor, *Computer Assisted Decision Making*, p. 189–197. Elsevier (North-Holand).

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

V. Rajkovič and M. Bohanec (1991). Decision support by knowledge explanation. In H. G. Sol and J. Vecsenyi, editors, *Environments for Supporting Decision Processes*, p. 47–57. North Holland.

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

A. Samuel (1967). Some studies in machine learning using the game of checkers II: Recent progress. *IBM J. Res. Develop*., 11, p. 601–617.

T. L. Saaty (1993). *Multicriteria Decision Making: The Analytic Hierarchy Process*. RWS Publications.

A. D. Shapiro (1987). *Structured induction in expert systems*. Turing Institute Press in association with Addison-Wesley Publishing Company.

T. J. Stewart (1992). A critical survew on the status of multiple criteria decision making theory and practice. *Omega, Intl. J. of Mgmt. Sci*, 20(5/6), p. 569–586.

B. Zupan (1997). *Machine learning based on function decomposition*. Ph.D. Thesis. University of Ljubljana, Faculty of Computer and Information Sciences.

B. Zupan, M. Bohanec, I. Bratko, and J. Demšar (1997). Machine learning by function decomposition. In *Proc. International Conference on Machine Learning ICML-97*, Nashville, TN.