Financial data set analysis - hierarchical testing with GUHA method

David Coufal**
Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2, 182 07 Praha 8, Czech Republic
e-mail: coufal@cs.cas.cz

Summary. The paper deals with structure analysis of financial data set released within PKDD’99 discovery challenge. The aim of the analysis is to design a decision tree, which can be used either as structure description of analyzed data or for prediction of certain, from banking point of view interesting, facts. In a design process of a tree the GUHA method is employed as a main part of here proposed hierarchical testing algorithm. The experiments show that this algorithm is reasonable and applicable well with good results.

1 Introduction

In the year 1999 we participated in discovery challenge for financial data set announced within the 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases - PKDD’99, held in Prague, September 15-18. The approach used in our paper [1] employed GUHA method for analysis of given data. As a result of this analysis we have obtained a set of hypotheses revealing properties (or their combinations) supporting/not supporting the fact of good or bad loan payment policy. Such information is of course interesting, however, in certain situations we would like to have some decision guidelines how to determine if the asking client is potentially good or bad. See an example, in [1] we have revealed that presence of AvgM-sanction-interest-no property coheres with a good payment policy. On the other hand presence of order-household-no property coheres with a bad payment policy. Now what should we do with a client which satisfy simultaneously both these properties? In this example we have only two properties ”going against each other”, however, in [1] we have published 20 properties (Tables 1,2) supporting/not supporting good or bad loan payment policy. Hence the mixture of mutually excluding properties can be in the case of a particular client very complex.

The above considerations focused us on idea of creation of some tool, which can automatically decide if the asking client is good or bad. The chosen approach for this task’s solving is based on here proposed algorithm of

** This work was supported by GA CR 201/00/1489
hierarchical testing of hypotheses with GUHA method. Application of this
algorithm leads to design of a decision tree. Such a tree is then asked tool for
automated decisions taking.

The rest of the paper is composed in the following way. In the next sec-
tion, basics of GUHA methods are reviewed. The third section explains the
concept of hierarchical testing of hypotheses. The fourth section deals with
application of this algorithm on financial data set and shows the results of
this application. The paper is concluded by the fifth section.

2 Basics of GUHA method

In this section we review shortly basics of GUHA method. The review is given
in a very limited way, therefore a reader interested in more details is referred
to [2–4].

Source database and data matrix: Let’s start with the notion of source
database. Source database is formally a matrix with $N$ rows and $M$ columns,
where the rows correspond to objects and the columns to variables defined
on these objects. Each variable $V^j, j = 1, \ldots, M$ has its value $V^j(o_i)$ for
each object $o_i, i = 1, \ldots, N$. Value $V^j(o_i)$ then represents the value in $i$th row and
$j$th column of source database. The source database is in fact the data set
we are given to process by GUHA method.

Variables of source database can be *nominal or ordinal*. For nominal vari-
able the set of its possible values (the range of variable) is finite, usually
not ordered. The elements of this set are called classes. For ordinal variable
the set of its possible values is a linearly ordered set, possibly not finite. For
purpose of this paper only subsets of real numbers are considered.

The basic object the GUHA works on is dichotomized source database,
which is a matrix of zeros and ones. This matrix, called as *data matrix*, is
created on base of source database by process of variables *categorization*.
During this process, each variable is associated with a set of its categories. A
category of a variable is then a subset of the set of variable’s possible values.
For nominal variable each class represents one category. For ordinal variable
a category is given by an interval defined in the set of variable’s possible
values.

The rows of data matrix represent the objects of source database and the
columns represent categories. Which categories are defined for data matrix is
user depend task and in fact it is driven by the purpose of overall task solved
by GUHA. Denote by $C^j$ some category derived on base of $j$th variable
$V^j$. For each object $o_i$ and each category $C^j$ of data matrix, there is 1 in
the $i$th row and category’s respective column if and only if $V^j(o_i) \in C^j$. If
$V^j(o_i) \notin C^j$, then there is 0.
Hypotheses formation and testing: A general form of hypothesis in GUHA method is $A \approx B$. The first part $A$ is called as antecedent, the second part $S$ as succedent. To form the whole hypothesis, antecedent and succedent are bind by a quantifier $\approx$. The antecedent and succedent (together called as cedents, when they need not be distinguished) are actually boolean propositions formed on base of categories employing standard and $\&$ and negation $\neg$ boolean connectives. Denote as literal $L$ term $C$ or $\neg C$, where $C$ is a category, then cedent has a general form

$$L_1 \& L_2 \& \ldots \& L_{n-1} \& L_n ,$$

number $n$ is length of cedent. Since cedents are boolean propositions we are able to determine for particular object $o_i$ and particular cedent if the proposition given by this cedent has the value 1 (it is satisfied by $o_i$) or it has value 0 (it is not satisfied by $o_i$). For given antecedent $A$ and succedent $S$ we can proceed such a testing for all objects of source database. The information about cedents behavior then can be expressed in the form of so called four fold table (ff-table):

$$\begin{array}{c|c|c|c}
S & \neg S \\
\hline
A & a & b \\
\neg A & c & d \\
\end{array}$$

where $a$ is the number of objects both satisfying $A$ and $S$; $b$ is the number of objects satisfying $A$ and not satisfying $S$; $c$ is the number of object not satisfying $A$ and satisfying $S$; and, in the end, $d$ is the number of objects both not satisfying $A$ and not satisfying $S$. Note that $a + b + c + d = N$.

The semantics of a quantifier is given by its associated function. An associated function is a boolean function operating on ff-tables (on values $a, b, c, d$). Consider some data matrix, then for each hypothesis $H : A \approx S$ we can determine its ff-table. On base of values of this ff-table, value of associated function of quantifier $\approx$ can be computed. If this value is 1, then we say that hypothesis $H$ is valid in data matrix, if value is 0 then we say that the hypothesis $H$ is not valid. We mention here only one quantifier, many others can be defined.

**founded almost implication quantifier — FIMPL:** $\Rightarrow_{cp, base}$

The associated function of FIMPL quantifier has two optional parameters $cp$ and $base$, restricted to inequalities $0 < cp \leq 1; base \geq 1$. On base of values $a$ and $b$ of ff-table the prob statistic is computed, $\text{prob}(a,b) = \frac{a}{a+b}$.

The associated function then has the form

$$\Rightarrow_{cp, base} (a,b) = \begin{cases} 1 & \text{prob}(a,b) \geq cp \text{ and } a \geq base \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$^{1}$ Note, that although the associated function is generally a function of values $a, b, c, d$, in this case, it depends only on values $a, b$. 
The meaning of FIMPL quantifier is the following. For particular antecedent \( A \) and particular succedent \( S \) value of FIMPL quantifier’s associated function is 1 if and only if at least 100 \( \cdot \text{cp} \% \) of objects satisfying \( A \) also satisfy \( S \), and the number of these objects is greater or equal to value of parameter base.

Now when we know what is the notion of hypothesis, together with the notion of its validity or not validity, we can explain the process of hypotheses testing. This process is based on so called task definition. During the task definition a rough template for tested hypotheses is specified. That is, specification of maximal length of antecedent and succedent - \( n^{\text{max}}_A, n^{\text{max}}_S \) is given. Further, two sets of categories \( \text{Cat}_A, \text{Cat}_S \) are determined. On base of these specifications broad set of relevant antecedents and succedents is given. A relevant antecedent is a cedent (1) of length \( n_A \leq n^{\text{max}}_A \), where literals are from the set \( \text{Cat}^A \). Similarly for relevant succedent with \( n^{\text{max}}_S \) and \( \text{Cat}^S \). The last step of task definition is the choice of certain quantifier together with the values of its parameters. Relevant hypothesis is then a hypothesis formed from some relevant antecedent and some relevant succedent bind together with the chosen quantifier. Each possible relevant hypothesis is then tested for its validity. When hypothesis is valid then it is revealed by GUHA.

Obviously, the GUHA method can be reasonably used only if it is implemented in the form of computer program. Currently there is implementation denoted as GUHA + available at http://www.cs.cas.cz/ics/software.html web page.

3 Hierarchical testing algorithm

In this section we introduce the concept of hierarchical testing of hypotheses with GUHA method, which consequently forms the base of decision tree design process.

Assume that we are given by a set \( \mathcal{T} \) of training data for some classification task, \( \mathcal{T} = \{ (x^t, c^t) \} \), \( x^t = (x^t_1, \ldots, x^t_n) \), \( x^t_1 \in X_1, \ldots, x^t_n \in X_n, c^t \in Y \), \( t \) is index within the set \( \mathcal{T} \). Therefore, for each given point \( x^t \) we have class \( c^t \) into which it is classified. Such a training set \( \mathcal{T} \) can be transformed into a source database in the following way. Each point \( (x^t, c^t) \) is seen as object \( o_t \) of source database. On these objects, values of \( n + 1 \) variables \( V^{X_1}, \ldots, V^{X_n}, V^Y \) are defined. For particular object \( o_t \), we have \( V^{X_t}(o_t) = x^t_1, \ldots, V^{X_n}(o_t) = x^t_n, V^Y(o_t) = c^t \). This forms a source database with \( |\mathcal{T}| \) rows\(^2\) and \( n + 1 \) columns.

Consequently, after the source database is created, respective categories for each variable can be defined. In the case of \( V^{X_1}, \ldots, V^{X_n} \) variables, the categories definition is user depend, for variable \( V^Y \) each category is given by one class from range of \( Y \), because \( Y \) is a nominal variable. Definition of categories then induces a data matrix created on base of training data set \( \mathcal{T} \).

\(^2\) \( |\cdot| \) is a cardinality of a set
On such a data matrix GUHA analysis can be performed according to
task definition given as: \( n^{\text{max}}_S \leq n; \) \( n^{\text{max}} = 1; \) \( \text{Cat}_S^A \) consists of all categories defined for variables \( V_{X_1}, \ldots, V_{X_n}; \) \( \text{Cat}_S^S \) consists of categories defined for variable \( V_Y. \) As quantifier, FIMPL quantifier is used with some chosen (see
discussion below) values of \( cp \) and \( \text{base} \) parameters.

Now, what does it mean when a hypothesis \( H : A \implies_{cp, \text{base}} S \) is revealed? It
means, that if an object \( o_i \in T \) satisfies antecedent \( A \) of this hypothesis,
then there is at least \( cp \) probability that it also satisfies its succeedent \( S. \)
Let \( H^A \) denote set of all objects satisfying antecedent of \( H. \) The number of
these objects is given by a sum \( a + b \) of hypothesis \( \text{fit-table} \) values. Objects
\( o_i \in H^A \) then can be according to this hypothesis \( H \) classified to class given
by category forming hypothesis succeedent \( S. \) However, this classification is
reasonable only when a value of parameter \( cp \) is high.

On base of this considerations we can approach to algorithm of hierar-
chical testing (HT) of hypotheses with FIMPL quantifier. Let a data matrix
(initial data matrix \( D\mathcal{M}_0 \)) is constructed as specified above, set \( \mathcal{H} = \emptyset, i = 1, \)
and perform the following loop:

1. process GUHA task with specifications as above, for \( i = 1 \) on initial data
   matrix \( D\mathcal{M}_0, \) for \( i > 1 \) on reduced data matrix \( D\mathcal{M}_{i-1}; \)
   if there is no hypothesis revealed, go into step 3;
   else choose from revealed hypotheses the hypothesis with the maximal
   value \( a + b \) (if there are several such hypotheses choose some),
   denote this
hypothesis as \( H_i, \) add it into the set \( \mathcal{H} \) and continue to step 2;
2. exclude from data matrix \( D\mathcal{M}_{i-1} \) objects satisfying antecedent of
   hypothesis \( H_i \) chosen in preceding step 1 - this forms reduced data matrix
   \( D\mathcal{M}_i, \) go to step 3;
3. if reduced data matrix \( D\mathcal{M}_i \) is empty or there is no hypothesis revealed
   according to step 1 end loop;
   else set \( i = i + 1 \) and go back to step 1;

Table 1. HT algorithm

When the loop is finished, we obtain a set of hypotheses \( \mathcal{H}, \) suppose it
is not empty, \( \mathcal{H} = \{ H_1, H_2, \ldots, H_m \}, \) \( H_i : A_i \implies_{cp, \text{base}} S_i. \) Each hypo-
thesis \( H_i \in \mathcal{H} \) determines a set of objects \( H_i^A \) which can be classified into the
respective class given by succeedent \( S_i. \) From logic of HT algorithm the
natural hierarchy of revealed hypotheses issues. This hierarchy coincides with
the ordering according to index \( i, i = 1, \ldots, m; \) and, in fact, together with forms of particular hypotheses, determines structural description of initial
data matrix. This description is the best represented in the graphical form:
\[
\begin{array}{c}
\text{start } \rightarrow A_1 \rightarrow A_2 \rightarrow \cdots \rightarrow A_m \rightarrow \text{end} \\
\downarrow \quad \downarrow \quad \downarrow \\
S_1 \quad S_2 \quad S_m
\end{array}
\]

**Figure 1.** Structural description given by HT algorithm

What does this schema mean? Clearly, with respect to operation of HT algorithm, the schema given in Fig.1 represents the following procedure. Take an object of initial data matrix \(DM_0\) (start position) and test if it satisfies antecedent \(A_1\) of first hypothesis, if it is so, conclude that this object belongs to class given by succedent \(S_1\). If the object does not satisfy \(A_1\), proceed to test if it satisfies antecedent \(A_2\). If \(A_2\) is satisfied, conclude that object belongs to \(S_2\) class. If \(A_2\) is not satisfied proceed to test \(A_3\) and so on. The procedure of testing and possible classification is performed in direction of increasing index \(i\), until some classification is taken or object passes without classification (end position).

The structural description represented by schema in Fig.1 can be also viewed as a decision tree. Utilizing this tree, an outside object of the same type as the objects of initial data matrix \(DM_0\), can be classified according to information extracted from \(DM_0\).

In the rest of this section we will discuss some properties characterizing set of HT’s revealed hypotheses \(\mathcal{H}\). These properties will be discussed especially with respect to choice of values of FIMPL quantifier’s parameters, which are used in GUHA testing within the HT algorithm. Let us consider the following properties of \(\mathcal{H}\) set:

1. How many objects of source database the union of sets \(H_i^A\) covers?
2. What is the minimal value from \(\text{prob}(a_i, b_i)\) values?
3. What is the number \(m\) of hypotheses in the set \(\mathcal{H}\)?

Clearly, when we take the set of hypotheses \(\mathcal{H}\) as some rough structural description of source database with respect to classification of its objects, then above properties have the following meaning.

In the case of the first property we want the union \(\bigcup H_i^A\) to cover (include) maximal number of objects of source database\(^3\). This demand is reasonable, because we would like to have description of source database with maximal evidence about its objects, or, from the decision tree point of view, we would like to pick out maximal information from source database for purpose of decision tree design.

In the case of the second property we investigate how reliable are hypotheses of set \(\mathcal{H}\), i.e., how reliable are decisions about the classification according to particular hypothesis \(H_i\). Naturally, our demand is to have the reliability as high as possible.

---

\(^3\) Terminology remark: Obviously, the objects of data matrix and source database are the same. Since data matrix is derived on base of source database, we will refer its objects as objects of source database.
In the case of the third property we investigate the complexity of our structural description, the more hypotheses in $\mathcal{H}$ we have, the more complex our description is. Of course, we would like to have as simple as possible description.

The satisfaction of above properties is concurrent and it is mainly affected by specification of $cp$ and $base$ value of FIMPL quantifier.

First consider $cp$ parameter. When its value is high then we have assured that property 2 is satisfied well, because for each $H_i$, $\text{prob}(a_i, b_i) \geq cp$ holds. However, for high value of $cp$ we can obtain set of hypotheses not covering enough (poor satisfaction of property 1) the objects of source database. The extreme case is when we obtain empty set $\mathcal{H}$. When the value of $cp$ is low, then we get $\mathcal{H}$ usually with good satisfaction of property 1, but the hypotheses are not reliable.

The value of $base$ parameter is not so crucial as the value of $cp$ is. If we take $base = 1$, then we enable, with respect to $cp$ value, good satisfaction of property 1, however the complexity of description according to property 3 increases. Further, with low value of $base$ parameter, hypotheses with poor statistical justification are revealed. On the other hand, high values of $base$ parameter negatively affects good satisfaction of property 1.

From the above discussion it is seen that to find compromise solution for FIMPL quantifier’s parameters setting is hard. Actually, this setting is the result of trials and errors process. Usually we demand to have reliable description (property 2), i.e., we demand high value of $cp$ (say 0.9), and then we test the covering of resulting set $\mathcal{H}$ with respect to some low value of $base$ parameter. If the covering is bad then we lower the value of $cp$ and rerun HT algorithm to get new set $\mathcal{H}$. The value of $base$ parameter is usually increased when in $\mathcal{H}$ we have a lot of hypotheses with low value of $a$ parameter.

Instead of further analysis of this problem we show the real application of HT algorithm resulting in a decision tree’s design. The demonstration is performed on the case of financial data set.

4 Financial data set

The financial data set was released for PKDD’99 discovery challenge purpose and it is available for download, together with its detail description, at http://lisp.vse.cz/pkdd99 web site.

In paper [1] we have analyzed this set utilizing GUHA method from Fisher quantifier’s point of view. As it was already stated in the first section, there is another possible view of analysis. It issues from an effort for structural description of data with respect to loan status classification of clients. This structural analysis can be performed via hierarchical testing algorithm and it results to design of decision tree. There are two possible view of this tree. The first one is to see tree as structural description of financial data set. The second one is to use this tree as tool enabling automatic classification of in
future asking clients, i.e., it enables to perform some kind of prediction. The prediction is then based on experience picked out from analyzed financial data set. This second view is our preferred one.

To analyze financial data set for decision tree design purpose, we have used the same data preprocessing as it was used in [1]. Since we will design a decision tree for loan status prediction, we could utilize only the data relating to accounts with granted loans. From this reason initial data matrix used for hierarchical testing consists of 682 objects (accounts). For each object, values of 69 descriptive variables were given or computed, with respective categorization defined, see [1]. The most important variable of these variables is the variable loan-status. This variable has two categories defined, loan-status-good and loan-status-bad. Satisfying of respective category then classify particular account.

To demonstrate our method, we have performed two experiments of hierarchical testing with different settings of input parameters. The settings are summarized in Table 2.

<table>
<thead>
<tr>
<th>experiment</th>
<th>( n_0^{\text{max}} )</th>
<th>( n_0^{\text{max}} )</th>
<th>( cp )</th>
<th>( \ln \sigma )</th>
<th>( m )</th>
<th>cover</th>
</tr>
</thead>
<tbody>
<tr>
<td>experiment 1</td>
<td>2</td>
<td>1</td>
<td>0.90</td>
<td>1</td>
<td>4</td>
<td>682</td>
</tr>
<tr>
<td>experiment 2</td>
<td>2</td>
<td>1</td>
<td>0.95</td>
<td>10</td>
<td>11</td>
<td>642</td>
</tr>
</tbody>
</table>

Table 2. Parameters of HT algorithm

The last two columns of this table refer to parameters of resulting sets \( \mathcal{H} \). The column \( m \) refers to number of particular hypotheses the respective \( \mathcal{H} \) set consists of. The cover column gives covering of respective sets, according to property 1. Let us now discuss these two experiments in more details.

**Experiments:** In the Table 3 we have explicit representation of found four hypotheses for first experiment.

<table>
<thead>
<tr>
<th>antecedent</th>
<th>succedent</th>
<th>( a )</th>
<th>( b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>AvgM-sanction-interest-yes</td>
<td>loan-status-good</td>
<td>603</td>
<td>50</td>
</tr>
<tr>
<td>card-type-no</td>
<td>loan-status-bad</td>
<td>23</td>
<td>2</td>
</tr>
<tr>
<td>acc-dist-at-(100000, 300000)</td>
<td>loan-status-good</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>acc-year-93</td>
<td>loan-status-bad</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3. Hypotheses of experiment 1
From this table we can see that although maximal length of antecedent \( n_A^\text{max} = 2 \) was specified, only the hypotheses with antecedent’s length of \( n_A = 1 \) were revealed. The reliability of particular hypotheses is greater than 90%, which can be easily confirmed using values of \( a, b \) parameters given in last two columns. Covering, as it is given in Table 2, is absolute, i.e., all objects of source database are covered. However, the last hypothesis is from statistical point of view not acceptable. The some protest can be given against third hypothesis, however in the less strength.

When we transform Table 3 into the form of schema given in Fig.1., i.e., into the form of decision tree, we get schema as it is given in Fig.2. The using is the same as it was explained for Fig.1. Take account of asking client and test if this account satisfies antecedent of first hypothesis. If it is so, then go into the right and conclude that it is potentially good client. If the antecedent of first hypothesis is not satisfied, test the antecedent of second hypothesis and so on. The possible results of the whole procedure are two. Either you classify a tested client, i.e., in hierarchy of hypotheses there is an antecedent which is satisfied by client’s account, or client remains unclassified because no antecedent is satisfied.

\[
\begin{align*}
& \text{start} \\
& \downarrow \\
& 1. \text{AvgM-sanction-interest-no} \rightarrow \text{loan-status-good} \\
& \downarrow \\
& 2. \text{card-type-no} \rightarrow \text{loan-status-bad} \\
& \downarrow \\
& 3. \text{acc-dist-a4-(100000,300000]} \rightarrow \text{loan-status-good} \\
& \downarrow \\
& 4. \text{acc-year-93} \rightarrow \text{loan-status-bad} \\
& \downarrow \\
& \text{end}
\end{align*}
\]

**Figure 2.** Decision tree for experiment 1

Decision tree given in Fig.2. is relatively good from above discussed three properties point of view. It covers the whole source database, particular hypotheses are reliable in 90% and \( \mathcal{H} \) consists of only four simply hypotheses.

Within the second experiment we tries to get even reliable description with comparison to experiment 1. To reach this aim we have increased value of \( cp \) parameter to \( cp = 0.95 \), see Table 2. When we performed HT algorithm with \( cp = 0.95 \) and \( base = 1 \) we got 18 hypotheses, however much of them with small values of \( a \) parameter, so we increased value of \( base \) parameter to \( base = 10 \), and rerun HT algorithm. The result of this run was revealing of 11 hypotheses presented in Table 4.
<table>
<thead>
<tr>
<th>antecedent</th>
<th>succedent</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. order-household-yes</td>
<td>l.s.-good</td>
<td>421</td>
<td>20</td>
</tr>
<tr>
<td>2. card-type-yes &amp; AvgM-sanction-interest-no</td>
<td>l.s.-good</td>
<td>65</td>
<td>2</td>
</tr>
<tr>
<td>3. loan-year-98 &amp; AvgM-sanction-interest-no</td>
<td>l.s.-good</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td>4. loan-duration-48 &amp; AvgM-Ibalance &gt; 40000</td>
<td>l.s.-good</td>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>5. owner-dist-a13 &gt; 3 &amp; AvgM-sanction-interest-yes</td>
<td>l.s.-bad</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>6. user-yes</td>
<td>l.s.-good</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>7. loan-pay-(2000, 5000] &amp; loan-year-94</td>
<td>l.s.-good</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>8. MinM-amount-sign ≤ -20000 &amp; AvgM-household &gt; 0</td>
<td>l.s.-bad</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>9. loan-pay-(2000, 5000] &amp; owner-dist-a5 &gt; 52</td>
<td>l.s.-good</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>10. acc-dist-a5-(17, 52] &amp; MaxM-Ibalance-(50000, 80000]</td>
<td>l.s.-good</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>11. acc-dist-a7 &gt; 6 &amp; AvgM-Ibalance ≤ 30000</td>
<td>l.s.-bad</td>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4. Hypotheses of experiment 2

\[
\text{start} \\
\downarrow \\
1. \text{order-household-yes} \rightarrow \text{l.s.-good} \\
\downarrow \\
2. \text{card-type-yes} \& \text{AvgM-sanction-interest-no} \rightarrow \text{l.s.-good} \\
\downarrow \\
3. \text{loan-year-98} \& \text{AvgM-sanction-interest-no} \rightarrow \text{l.s.-good} \\
\downarrow \\
4. \text{loan-duration-48} \& \text{AvgM-Ibalance} > 40000 \rightarrow \text{l.s.-good} \\
\downarrow \\
5. \text{owner-dist-a13} > 3 \& \text{AvgM-sanction-interest-yes} \rightarrow \text{l.s.-bad} \\
\downarrow \\
6. \text{user-yes} \rightarrow \text{l.s.-good} \\
\downarrow \\
7. \text{loan-pay-(2000, 5000]} \& \text{loan-year-94} \rightarrow \text{l.s.-good} \\
\downarrow \\
8. \text{MinM-amount-sign} \leq -20000 \& \text{AvgM-household} > 0 \rightarrow \text{l.s.-bad} \\
\downarrow \\
9. \text{loan-pay-(2000, 5000]} \& \text{owner-dist-a5} > 52 \rightarrow \text{l.s.-good} \\
\downarrow \\
10. \text{acc-dist-a5-(17, 52]} \& \text{MaxM-Ibalance-(50000, 80000]} \rightarrow \text{l.s.-good} \\
\downarrow \\
11. \text{acc-dist-a7} > 6 \& \text{AvgM-Ibalance} \leq 30000 \rightarrow \text{l.s.-bad} \\
end
\]

Figure 2. Decision tree for experiment 2
Almost all revealed hypotheses are of antecedent's length \( n_A = 2 \). Covering of set \( H \) is 642. Clearly, now the covering is not absolute, which is a "tax" for increased reliability of hypotheses and rising value of base parameter. The decision tree corresponding to Table 4 is given in Fig. 2.

When we take a look at revealed hypotheses, we can see that main properties relating to good loan payment policy are \( \text{AvgM-sanction-interest-no, order-household-yes, and user-yes} \). On the other hand, properties relating to bad loan payment policy are \( \text{AvgM-sanction-interest-yes, card-type-no, owner-dist-13 > 3} \) (unemployment rate in 1996). These results coincides with the ones published in [1].

In the end discuss the acc-year (hyp. 4 of first experiment) and loan-year (hyp. 3,7 of second experiment) variables revealing in above decision trees. It is clear that in the case of a new asking client, categories acc-year-93 and loan-year-98/94 will be hard or cannot be satisfied, i.e., they are useless in the context of decision tree. To avoid this situation, decision trees should be constructed on base of source database without these variables. This can be easily done. The reason for retaining these variables in our experiments is that designed decision trees give structure description of analyzed financial data set. From this point of view, it is possibly interesting to know that bad or good payment policy is associated with particular year of account's or loan's establishment.

5 Conclusions

In this paper algorithm of hierarchical testing (HT) of hypotheses was proposed. The algorithm is based on GUHA method testing of hypotheses with FIMPL quantifier. On base of experimental results it was shown that HT algorithm is the suitable tool for design of decision trees. These trees can be consequently used either as structural description of analyzed data or for prediction purposes. The suggested application of GUHA method can brings new ideas into the problematic of decision trees construction. On the other hand this problematic can have positive impact on further development of the method itself.

References