Mining Categorical Concept Hierarchies in Large Databases*
Been-Chian Chien and Su-Yu Liao
Department of Information Engineering, I-Shou University
Kaohsiung 840, Taiwan, R.O.C
cbc@isu.edu.tw

ABSTRACT
A concept hierarchy is a kind of concise and general form of concept description that organizes relationships of data and expresses knowledge in databases. In this paper, we propose an approach to generate concept hierarchies automatically for a given data set with nominal attributes based on rough entropy. The proposed method reduces the number of attributes after each process of generating concept level. Hence, the algorithm is efficient and scaleable for large databases. We also give an algorithm to find and evaluate rules from the generated concept hierarchy. The experiments show that the concept hierarchy mined by the proposed approach contains important and meaningful concepts effectively. The results of hierarchy structures can thus be applied to discover knowledge with other different purposes.

Keywords: Knowledge Discovery, Rough Set, Concept Hierarchy

1. INTRODUCTION
Knowledge discovery in databases (KDD) is one of the important research topics in both area of machine learning and databases. The objective of KDD is to examine how to extract knowledge automatically from huge amounts of data. The discovered knowledge from a large practical database can be applied to many applications, such as decision-making, marketing, information management, and so on. For different applications, the different types of knowledge lead to lots of methodologies in the development of data mining techniques [4]. For example, the Apriori algorithm for mining association rules [1], the classification learning algorithm for decision trees construction [14] and the unsupervised agglomerative algorithm for generating clusters [8]. Regardless of the mining approaches involved, we can imagine the various types of discovered knowledge as the concepts learned from databases based on different viewpoints.

Figure 1. A concept hierarchy of animal world.

Figure 2. A concept hierarchy.

For representing knowledge as concepts explicitly, a clear and meaningful concept description is required. Concept hierarchies are a kind of concise and general form of concept description widely used. Many knowledge representations are variants of concept hierarchies in different ways of expression. For transaction databases, as an example, association rules are the set of rules that describe the relative relationships in a partial ordering hierarchy of large item sets. Decision tree is another type of concept hierarchy used for generating classification rules and classifying the categories of data. The other type of concept hierarchy, like clusters, is a one-level concept grouping data with high similarity into classes.

A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts on one or a set of attribute domains. Such mappings may organize the set of concepts in partial order according to a general-to-specific ordering and form a tree-like structure (a hierarchy, a taxonomy) as depicted in Figure 1, mapping low-level concepts at Level 4 to more general concepts at Level 0. The most general concept can be an universal concept, whereas the most specific concepts correspond to the specific values of attributes in the database. Each node in the tree of concept hierarchy represents a concept which helps to express knowledge and data relationships in a database as a concise, high level term [5]. Formally, suppose that a hierarchy \( H \) is defined on a set of domains \( D_1, \ldots, D_k \), we have \( H_l : (D_1 \times \cdots \times D_k) \Rightarrow H_{l-1} \Rightarrow \cdots \Rightarrow H_0 \), where \( H_l \) denotes the set of concepts at the primitive level, \( H_{l+1} \) denotes the concepts at one level higher than those at \( H_l \), and \( H_0 \) represents the most general concept on the top level denoted as “ANY”, as Figure 2 illustrates.

It is reasonable for different users that distinct concept hierarchies can be constructed on the same attribute(s) based on users’ viewpoints or preferences. For constructing different types of concept hierarchies, many methods for numerical or categorical data have been developed. We summarize the methods of generating concept hierarchies briefly in the

* This work was supported in part by the National Science Council of Taiwan, R.O.C under Grant NSC-91-2213-E-214-018
following. The first kind of methods is to specify concept hierarchies by users, experts or data analysts according to a partial or total ordering of attributes explicitly at the schema level. A user or expert can easily define a concept hierarchy by semantics of schema or his/her need. Another kind of methods is to define a portion of a concept hierarchy by clustering explicit data in databases. However, it is essentially unrealistic to define an entire concept structure by explicit value enumeration manually in a large database. Thus, most of methods of this kind focus on automatic generation of concepts by grouping data by clustering techniques. The related researches include Cluster/2 by Michalski and Stepp [9], COBWEB by Fisher [3], hierarchical and parallel clustering by Hong and Mao [6], etc. The third kind of methods is that given a set of attributes, the system can then try to generate the partial ordering of attributes automatically so as to construct a meaningful concept hierarchy [4]. Without knowledge of data semantics, a concept hierarchy will be generated from the distinct values of attributes at the lowest level of hierarchy; then concepts are generated in higher levels. Some researches on generating concept hierarchies in a single numerical attribute [5] and finding the partial ordering in a set of categorical attributes [7] have been proposed. However, the problem for finding a complete hierarchical ordering from an arbitrary set of categorical attributes in a large database efficiently and meaningfully is still interesting and important.

The aim of this paper is to automatically construct a meaningful concept hierarchy by their partial ordering relationships among attributes in databases. For determining the partial ordering of categorical attributes to obtain the concept hierarchy, an efficient and effective methodology of data analysis is needed. We introduce an information measure called rough entropy, which is based on the rough set partitions. It can be proved that the values of rough entropy are monotonously increasing in accord with the generalization in higher level of concept hierarchy. We then propose an efficient algorithm making use of rough entropy to find a weakly relevant attribute, grouping the most relevant concepts and removing the weakly relevant attributes recursively to generate the concept hierarchy for a given set of attributes in a large database. A rule generation algorithm and two experiments are given to show the generated concept hierarchy is meaningful and effective.

The paper is organized as follows: Section 2 introduces the concepts of rough set theory and rough entropy. In Section 3, we presented the proposed algorithm based on rough entropy for generating concept hierarchies on categorical attributes. In Section 4, we proposed a flexible algorithm for generating classification rules. In Section 5, we show the performance of the proposed method by two experiments. Finally, conclusions and future work are discussed in Section 6.

2. ROUGH SET AND ROUGH ENTROPY

The rough set theory was first proposed by Zdzislaw Pawlak in 1982 [10][13]. Rough set is used to deal with the identification of common attributes in data sets [11][12]. It has been broadly and successfully applied to knowledge discovery in databases. This theory provides a powerful foundation to reveal and discover important structures in data and to classify complex objects. An attribute-oriented rough sets technique can reduce the computational complexity of learning processes and eliminate the unimportant or irrelevant attributes so that the knowledge can be learned from large databases efficiently. In an information system, some categories of objects cannot be distinguished in terms of the available attributes. They may be just defined roughly or approximately. The rough set theory deals with an information system represented in a form of a table that consists of objects and attributes. The idea of rough sets is based on the establishment of equivalence classes on the given data set \( U \) and supports two approximations called lower approximation and upper approximation. The lower approximation of a concept \( X \) contains the equivalence classes that are certain to belong to \( X \) without ambiguity. The upper approximation of a concept \( X \) contains the equivalence classes that cannot be described as not belonging to \( X \). We give out the formal definitions of the rough set theory. Let \( S = (U, A) \) denote an information system where \( U \) means a non-empty finite set of objects with a non-empty finite set of attributes \( A \). For all \( a \in A \) and each \( B \subseteq A \), we define a binary indiscernibility relation \( R_A(B) \) as follows:

**Definition 1:** Let \( x, y \in U \), we say that objects \( x \) and \( y \) are indiscernible if the equivalence relation \( R_A(B) \) is satisfied on the set \( U \):

\[
R_A(B) = \{(x, y) \mid x, y \in U, \forall a \in B, a(x) = a(y)\}.
\]

**Definition 2:** \( apr(U, R_A) \), is called an approximation space. The object \( x \in U \) belonging to one and only one equivalence class. Let \([x_B]\) denote an equivalence class of \( R_A(B) \) and \([U]\) denote the set of all equivalence classes \([x_B]\) for \( x \in U \). We have \([x_B] = \{y \mid x R_A(B) y, x, y \in U\}\) and \([U]\) = \([x_B] \mid x \in U\).

**Definition 3:** Let \( S = (U, A) \), \( x_i \in U \) and \( B \subseteq A \). The rough entropy of information based on a subset of attributes \( B \) is

\[
E(B) = -\sum_{i=1}^{n} \left[ \frac{|[x_i]_B| \log_2 \frac{1}{|[x_i]_B|}}{n} \right],
\]

where \([x_i]_B\) denotes the cardinality of the equivalence class of \([x_i]\) and \( n \) is the number of objects in \( U \).

**Theorem 1:** Let \( S = (U, A) \), \( B, B' \subseteq A \). If \([x_i]_B \subseteq [x_i]_{B'}\), then \( E(B) \leq E(B') \).

Proof. The values of rough entropy

\[
E(B) = -\sum_{i=1}^{n} \left[ \frac{|[x_i]_B| \log_2 \frac{1}{|[x_i]_B|}}{n} \right],
\]

\[
E(B') = -\sum_{i=1}^{n} \left[ \frac{|[x_i]_{B'}| \log_2 \frac{1}{|[x_i]_{B'}|}}{n} \right].
\]

Since \([x_i]_B \subseteq [x_i]_{B'}\), we have \([x_i]_B \leq [x_i]_{B'}\),

\[
\sum_{i=1}^{n} |[x_i]_B| \log_2 |[x_i]_B| \leq \sum_{i=1}^{n} |[x_i]_{B'}| \log_2 |[x_i]_{B'}|,
\]

\[
-\sum_{i=1}^{n} |[x_i]_B| \log_2 \frac{1}{|[x_i]_B|} \leq -\sum_{i=1}^{n} |[x_i]_{B'}| \log_2 \frac{1}{|[x_i]_{B'}|}.
\]

Hence,

\[
E(B) \leq E(B').
\]

**Corollary 1:** Let \( S = (U, A) \), \( B, B' \subseteq A \). If \( B' \subseteq B \), then \( E(B) \leq E(B') \).

Proof. For \( B' \) and \( B \), the two subsets of \( A \), if \( B' \subseteq B \), we have \([x_i]_{B'} \subseteq [x_i]_B\) for all \( x_i \in U \).

As shown in Theorem 1, since \([x_i]_{B'} \subseteq [x_i]_B\), we have

\[
E(B) \leq E(B').
\]

Theorem 1 states that the rough entropy of the equivalence class decreases monotonously as the granularity of information
becomes smaller through finer partitions. Hence, the concept is more general while the value of rough entropy is larger.

3. THE ALGORITHM OF GENERATING CONCEPT HIERARCHIES

In this section, an algorithm of generating concept hierarchies for a database with categorical attributes is proposed. The symbols used in the algorithm are defined in Section 3.1. The proposed algorithm is described in Section 3.2. An example is given to explain the algorithm for generating concept hierarchy in detail in Section 3.3.

3.1 Notations

The symbols used in our proposed method are defined in the following.

- **U**: A database with categorical attributes.
- **n**: The number of objects in **U**.
- **A**: The finite set of categorical attributes in **U**.
- **m**: The number of attributes in **A**.
- **B**: A set of reduced attributes, **B** ⊆ **A**.
- **Bj**: The j-th set of reduced attributes, **Bj** ⊆ **A**.
- **|B|**: The number of attributes in **B**.
- **Aj**: The j-th attribute of **A**, **Aj** ∈ **A**, 1 ≤ j ≤ **m**.
- **|[x]|Bj**: Let **Bj** ⊆ **A**, the cardinality of equivalence classes of **[x]|Bj**, **x** ∈ **U**.
- **Ck**: The k-th concept.

Except the above symbols, we define an important property of inclusion for the partitions in different sets of reduced attributes as follows:

**Definition 4**: Let **S** = (**U**, **A**) and **x** ∈ **U**. Assume that **B**, **B′** ⊆ **A** and |**B**| = |**B′**|, we say that |**U**| is compatible included in |**U**| if the partitions of **U** on **B** and **B′** satisfy |**x||B| ≤ |**x||B′|, for all **x** ∈ **U**, denoted as |**U**| ≤ |**U**|.

3.2 The proposed hierarchy generating algorithm

After the definitions in Section 3.1. We give the algorithm for generating a meaningful concept hierarchy with respect to a given database. The algorithm is described in seven main steps:

**Algorithm**:

**Input**: A database **U** with the attributes **A**.

**Output**: A concept hierarchy **H** for **S** = (**U**, **A**).

Step 1: Initialize the values of **m′** and **k**.

Step 2: Generate the lowest level of concept hierarchy **Hm** with the set of attributes **A**. That is, **Hm** = |**U**|. Let the initial set of reduced attributes **B** = **A**.

Step 3: Let |**B**| be the number of attributes in the set of reduced attributes **B**, we set **m′** = |**B**| - 1. Find **Pm(B)**, which is the set of all possible subsets with |**m′** attributes in the set of reduced attributes **B**.

Step 4: Compute the rough entropy **E(B)** for all **Bj** ∈ **Pm(B)**.

Step 5: Generate a higher level of concept hierarchy **Hm+1** by the following three sub-steps:

5.1) Find **B′** = max( |**E(Bj)**| ), for **Bj** ∈ **Pm(B)**; then generate new concepts **Cj** in **Hm+1** with the partition of **[U]|B|**. The **Cj** is the concept that merge the concepts **Cj** in the lower level **Hm+1** belonging to the same partition of |**x||Bj||.

5.2) For the remainder concepts in **Hm+1**, if there exists **Bj** ∈ **Pm(B)** and **Bj** = **B′**, the property |**U||Bj|| is satisfied. We then generate another new concepts **Cj** in **Hm+1**. The **Cj** is also the concept that merge the concepts **Cj** in the lower level **Hm+1** belonging to the same partition of |**x||Bj||.

5.3) For the others **Cj** in the lower level **Hm+1**, if **Cj** = |**x||Bj|| and **Cj** = |**x||Bj||, the **Ci** is unchanged and added into **Hm+1**.

Step 6: Let the set of reduced attributes **B** = **B′**.

Step 7: If **B** ≠ ∅, then go to Step 3; otherwise, output all levels of the concept hierarchy **H0**, ..., **Hm**.

The time complexity of the proposed algorithm is dependent on the number of attributes **m** in each concept level. The number of attributes is decreased after a new concept level is generated. Hence, the overall time complexity of the algorithm is bounded by **O(m′n)**, where **n** is the size of the database.

3.3 An example

We give an example to explain the above algorithm more clearly. Assume that the example used in Table 1 shows a database containing eight objects **U** = {**x1**, **x2**, **x3**, **x4**, **x5**, **x6**, **x7**, **x8**} and five attributes **A** = {**gender**, **income**, **insurance**, **status**, **car**}. The symbolic categories in the five attributes, respectively, are:

- **gender**: {male (m), female (f), boy (b), girl (g)}
- **income**: {high (h), low (l)}
- **insurance**: {yes (y), no (n)}
- **status**: {single, married, divorce, yet}
- **car**: {yes, no}

We start the steps:

<table>
<thead>
<tr>
<th>Table 1. The example dataset.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>data</strong></td>
</tr>
<tr>
<td><strong>x1</strong></td>
</tr>
<tr>
<td><strong>x2</strong></td>
</tr>
<tr>
<td><strong>x3</strong></td>
</tr>
<tr>
<td><strong>x4</strong></td>
</tr>
<tr>
<td><strong>x5</strong></td>
</tr>
<tr>
<td><strong>x6</strong></td>
</tr>
<tr>
<td><strong>x7</strong></td>
</tr>
<tr>
<td><strong>x8</strong></td>
</tr>
</tbody>
</table>

**Input**: The dataset in Table 1.

**Output**: The concept hierarchy **H**.

Step 1: Initialize the values of **m′** = 5 and **k** = 1.

Step 2: Generate the lowest level of concept hierarchy **H0** with the set of attributes **A**.

Step 3: Let **B** = {**gender**, **income**, **insurance**, **status**}, **H0** = |**U**| = |**U**|.

**H1** = |**x1||x2||x3||x4||x5||x6||x7||x8||, where**

{**B1**} = {**income**, **insurance**, **status**, **car**},
{**B2**} = {**gender**, **insurance**, **status**, **car**},
{**B3**} = {**gender**, **income**, **status**, **car**},
{**B4**} = {**gender**, **income**, **insurance**, **car**},
{**B5**} = {**gender**, **income**, **insurance**, **status**}.

Step 4: Compute the rough entropy **E(B)** for all **Bj** ∈ **Pm(B)**.

**E(B1)** = **E(**income, **insurance, status, car**) = 4,
**E(B2)** = **E(**gender, **insurance, status, car**) = 0,
**E(B3)** = **E(**gender, **income, status, car**) = 0,
**E(B4)** = **E(**gender, **income, insurance, car**) = 8,
**E(B5)** = **E(**gender, **income, insurance, status**) = 0.
Step 5: Generate a higher level of concept hierarchy $H_4$ by the following three sub-steps:

5.1) $E(B_5) \geq \max E(B)$ for $B_i \in P_d(B)$ to the comprehensibility of the recovered structure to make generality of each rule. This procedure can provide some insight of desired examining levels but also specifies the degree of proposed algorithm not only allows users to specify the number of concept hierarchy. A more informative test is to make dataset. Furthermore, to explore how well the generated concept hierarchies reflect the structure of the given database. Therefore, how well the induced concept hierarchy for a large database. Thus $H_4 = C_7 \cup C_8 \cup C_9 \cup C_{10} \cup C_{11}$

5.3) For the others $C_i$ in the lower level $H_5$, because the $C_i$ is unchanged and added into $H_4$. Thus $H_5 = H_4 \cup C_6$ is generated in $H_5$.

Step 6: The new reduced set of attributes is $B_6$, $B = B' = B_4 = \{\text{gender, income, insurance, car}\}$.

Step 7: Because $B = \{\text{gender, income, insurance, car}\}$, $|B| = 4$, then go to Step 3.

The final result of the concept hierarchy is shown in Figure 3.

4. THE RULES GENERATION IN A CONCEPT HIERARCHY

The aim of this paper is to generate a meaningful concept hierarchy for a large database. Therefore, how well the induced concept hierarchies reflect the structure of the given database should be discussed. However, in unsupervised learning there are no target outputs associated with the inputs, a widely used method for evaluating unsupervised learning system is to compute predictive accuracy as is done for supervised classifiers. The resulting accuracy serves as a measure of how well the system has discovered the underlying structure in the dataset. Furthermore, to explore how well the generated concept hierarchies for datasets, examining only one level appears to be inadequate. It is likely that users would select only part of the hierarchy to describe the target domain so that only a limited number of levels would be labeled. We give a rules generation algorithm here which allows the induction of rules at different levels of abstraction by employing the knowledge stored in the concept hierarchy. A more informative test is to make predictions by traversing the hierarchy to a limited depth. The proposed algorithm not only allows users to specify the number of desired examining levels but also specifies the degree of generality of each rule. This procedure can provide some insight to the comprehensibility of the recovered structure to make good prediction.

We first define a measure of generality of a concept as follow:

**Definition 5:** For a given concept $X \in S$, a rough attribute membership function of $X$ on the set of attributes $B$ is defined as $\mu_B^X(x) = \frac{|\{y \in B | y \cap X\}|}{|B|}$. $|\{y \in B | y \cap X\}|$ denotes the cardinality of the set $|\{y \in B | y \cap X\}|$. The rough membership value $\mu_B^X(x)$ can be interpreted as the conditional probability that an object $x$ belongs to $X$, given that the object belongs to $[\{y \in B | y \cap X\}]$. We use the value of $\mu_B^X(x) \in [0, 1]$ to be the measure of concept generality.

Assuming that a concept hierarchy is built from $S = (U, A)$, which is a training data of supervised learning. $U$ is a database with the set of attributes $A$. $A$ contains a decision attribute $A_d \in A$. Let the possible values in $A_d$ be $\{v_1, \ldots, v_k\}$. That is, there are $K$ possible results for each object $x_i$ in $U$.

The rules generation algorithm for the hierarchy $H$ generated from $S$ is described in five main steps, as follows:

**Input** : The concept hierarchy $H$, the generality of rules $\alpha$ and the maximum examining depth of hierarchical level $\lambda$.

**Output** : The set of rules for the concept hierarchy of $S$ with the specified $\alpha$ and $\lambda$.

Step 1: Initially, $l = 1$, $H_d$ is the level that the decision attribute $A_d$ is removed, and $C = \{C_i | C_i \in H_{d,l}\}$.

Step 2: Let $B$ be the set of reduced attributes at $H_{d+1}$, $B' = B \setminus A_d$, and compute $\mu_B^x(C_i)$ for all $C_i \in C$, $v_k$ is the value of decision attribute $A_d$ of $C_i$.

Step 3: Evaluate the $\mu_B^x(C_i)$ value of each $C_i$ and generate the rule for $C_i$.

For all $C_i \in C$, we have

Case 1: If $\mu_B^x(C_i) \geq \alpha$, then generate the rule of $C_i$ with generality $\geq \alpha$ and $C = C \setminus C_i$.

Case 2: If $\mu_B^x(C_i) < \alpha$, then remove $C_i$. If $l \geq \lambda$, then $C_i$ will be removed, $C = C \setminus C_i$.

Step 4: If $C = \emptyset$, then go to Step 6.

Step 5: Set $l = l + 1$, let $C = \{C_i | C_i \in H_{d,l} \text{ and } C_j \subseteq C\}$. If $C = \emptyset$, then go to Step 6, else $C = C$, go to Step 2.

Step 6: Pruning Rules. If there exists at least two rules with the same values of condition attributes $B'$ and the $v_k$ values of decision attribute $A_d$ are different. We remove the rules with the same prepositions except having maximum $\mu_B^X(C_i)$ value.

Following the example in Section 3.3, the concept hierarchy $H$ in Figure 3 is evaluated by the rules we generated. Assume that the decision attribute in the data set of Table 1 is $\text{car}$. The example of the rules generating algorithm is shown as follow:

**Input** : The concept hierarchy $H$ in figure 3, $\alpha = 0.5$ and $\lambda = 2$.

**Output** : The set of the rules for the concept hierarchy with the specified $\alpha$ and $\lambda$.

Step 1: Initially, $l = 1$, $H_d = H_1$ and $C = \{C_i | C_i \in H_d\} = \{C_4, C_6, C_9\}$.

Step 2: Let $B = \{\text{income, car}\}$, $B' = \{\text{income}\}$. Compute
Experiment 1. The first experiment uses the weather data set with 14 objects as shown in Table 2. Let $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}\}$, $A = \{\text{outlook, temperature, humidity, windy, play}\}$. The decision attribute is \{\text{play}\}. The symbolic categories in the five attributes, respectively, are:
- \text{outlook}: \{\text{sunny, overcast, rainy}\},
- \text{temperature}: \{\text{hot, mild, cool}\},
- \text{humidity}: \{\text{high, normal}\},
- \text{windy}: \{\text{true, false}\},
- \text{play}: \{\text{yes, no}\}.

Table 3 shows the results of accuracy in the concept of different hierarchy levels and degrees generality.

Experiment 2. The second experiment uses the lenses data set selected from UCI Machine Learning Repository [2] with 24 objects and 5 attributes. Let $A = \{\text{age of the patient, spectacle prescription, astigmatic, tear production rate, class}\}$. The decision attribute is \{\text{class}\}. The symbolic categories in the five attributes, respectively, are:
- \text{age of the patient}: \{\text{young, pre-presbyopic, presbyopic}\},
- \text{spectacle prescription}: \{\text{myope, hypermetrope}\},
- \text{astigmatic}: \{\text{no, yes}\},
- \text{tear production rate}: \{\text{red, normal}\},
- \text{class}: \{\text{hard contact lenses, soft contact lenses, no contact lenses}\}.

The experiment results are shown in Table 4.

Both of the two experiments show that the concept level of the decision attribute has a high classification rate. Moreover, we can classify all of the data only expanding the nodes of concept for one or two lower concept levels. The final rules generated from the concept hierarchy are also concise and effective.

6. CONCLUSIONS

Knowledge discovery from a large database is an important research topic of late years. The concept hierarchy is an explicit representation of knowledge and can be widely used in many applications. In this paper, we present an algorithm for generating concept hierarchies based on rough set and rough entropy. The algorithm can construct a meaningful concept hierarchy by their partial ordering relationships among attributes in databases automatically. The experimental results demonstrate that the proposed method is feasible and can be applied to other issues of knowledge discovery, such as classification, clustering etc. In the future, we are trying to cope with the data having missing values.

7. REFERENCES

Table 3. Classification results for the weather dataset.

<table>
<thead>
<tr>
<th>α</th>
<th>λ = 1</th>
<th>λ = 2</th>
<th>λ = 3</th>
<th>λ = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RN Acc (%)</td>
<td>RN Acc (%)</td>
<td>RN Acc (%)</td>
<td>RN Acc (%)</td>
</tr>
<tr>
<td>0</td>
<td>2 71.43</td>
<td>2 71.43</td>
<td>2 71.43</td>
<td>2 71.43</td>
</tr>
<tr>
<td>0.1</td>
<td>2 71.43</td>
<td>2 71.43</td>
<td>2 71.43</td>
<td>2 71.43</td>
</tr>
<tr>
<td>0.2</td>
<td>2 71.43</td>
<td>3 64.29</td>
<td>3 64.29</td>
<td>3 64.29</td>
</tr>
<tr>
<td>0.3</td>
<td>2 71.43</td>
<td>3 64.29</td>
<td>3 64.29</td>
<td>3 64.29</td>
</tr>
<tr>
<td>0.4</td>
<td>2 71.43</td>
<td>2 71.43</td>
<td>3 78.57</td>
<td>3 78.57</td>
</tr>
<tr>
<td>0.5</td>
<td>2 71.43</td>
<td>3 71.43</td>
<td>5 85.71</td>
<td>5 85.71</td>
</tr>
<tr>
<td>0.6</td>
<td>1 42.86</td>
<td>2 57.14</td>
<td>7 100.0</td>
<td>7 100.0</td>
</tr>
<tr>
<td>0.7</td>
<td>1 42.86</td>
<td>1 42.86</td>
<td>8 100.0</td>
<td>8 100.0</td>
</tr>
<tr>
<td>0.8</td>
<td>1 42.86</td>
<td>1 42.86</td>
<td>8 100.0</td>
<td>8 100.0</td>
</tr>
<tr>
<td>0.9</td>
<td>0 00.00</td>
<td>1 28.57</td>
<td>10 100.0</td>
<td>10 100.0</td>
</tr>
<tr>
<td>1</td>
<td>0 00.00</td>
<td>1 28.57</td>
<td>10 100.0</td>
<td>10 100.0</td>
</tr>
</tbody>
</table>

RN: the number of rules  Acc: accuracy of the set of rules

Table 4. Classification results for the lenses dataset.

<table>
<thead>
<tr>
<th>α</th>
<th>λ = 1</th>
<th>λ = 2</th>
<th>λ = 3</th>
<th>λ = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RN Acc (%)</td>
<td>RN Acc (%)</td>
<td>RN Acc (%)</td>
<td>RN Acc (%)</td>
</tr>
<tr>
<td>0</td>
<td>2 70.83</td>
<td>2 70.83</td>
<td>2 70.83</td>
<td>2 70.83</td>
</tr>
<tr>
<td>0.1</td>
<td>2 70.83</td>
<td>2 70.83</td>
<td>2 70.83</td>
<td>2 70.83</td>
</tr>
<tr>
<td>0.2</td>
<td>2 70.83</td>
<td>2 70.83</td>
<td>2 70.83</td>
<td>2 70.83</td>
</tr>
<tr>
<td>0.3</td>
<td>2 70.83</td>
<td>3 79.17</td>
<td>4 75.00</td>
<td>4 75.00</td>
</tr>
<tr>
<td>0.4</td>
<td>2 70.83</td>
<td>3 87.50</td>
<td>4 91.67</td>
<td>5 95.83</td>
</tr>
<tr>
<td>0.5</td>
<td>1 50.00</td>
<td>3 87.50</td>
<td>4 91.67</td>
<td>5 95.83</td>
</tr>
<tr>
<td>0.6</td>
<td>1 50.00</td>
<td>3 87.50</td>
<td>4 91.67</td>
<td>5 95.83</td>
</tr>
<tr>
<td>0.7</td>
<td>1 50.00</td>
<td>2 70.83</td>
<td>3 83.33</td>
<td>7 100.0</td>
</tr>
<tr>
<td>0.8</td>
<td>1 50.00</td>
<td>2 70.83</td>
<td>3 83.33</td>
<td>7 100.0</td>
</tr>
<tr>
<td>0.9</td>
<td>1 50.00</td>
<td>1 50.00</td>
<td>3 75.00</td>
<td>9 100.0</td>
</tr>
<tr>
<td>1</td>
<td>1 50.00</td>
<td>1 50.00</td>
<td>3 75.00</td>
<td>9 100.0</td>
</tr>
</tbody>
</table>

RN: the number of rules  Acc: accuracy of the set of rules