ELSEVIER

Theoretical Computer Science 292 (2003) 387–416

Theoretical Computer Science

www.elsevier.com/locate/tcs

Data abstractions for decision tree induction

Yoshimitsu Kudoh*, Makoto Haraguchi, Yoshiaki Okubo

Division of Electronics and Information Engineering, Hokkaido University, N 13 W 8, Sapporo 060-8628, Japan

Abstract

When descriptions of data values in a database are too concrete or too detailed, the computational complexity needed to discover useful knowledge from the database will be generally increased. Furthermore, discovered knowledge tends to become complicated. A notion of *data abstraction* seems useful to resolve this kind of problems, as we obtain a smaller and more general database after the abstraction, from which we can quickly extract more abstract knowledge that is expected to be easier to understand. In general, however, since there exist several possible abstractions, we have to carefully select one according to which the original database is generalized. An inadequate selection would make the accuracy of extracted knowledge worse.

From this point of view, we propose in this paper a method of selecting an appropriate abstraction from possible ones, assuming that our task is to construct a *decision tree* from a relational database. Suppose that, for each attribute in a relational database, we have a class of possible abstractions for the attribute values. As an appropriate abstraction for each attribute, we prefer an abstraction such that, even after the abstraction, the *distribution of target classes* necessary to perform our classification task can be preserved within an acceptable error range given by user.

By the selected abstractions, the original database can be transformed into a small *generalized database* written in abstract values. Therefore, it would be expected that, from the generalized database, we can construct a decision tree whose size is much smaller than one constructed from the original database. Furthermore, such a size reduction can be justified under some theoretical assumptions. The appropriateness of abstraction is precisely defined in terms of the standard information theory. Therefore, we call our abstraction framework *Information Theoretical Abstraction*.

We show some experimental results obtained by a system ITA that is an implementation of our abstraction method. From those results, it is verified that our method is very effective in reducing the size of detected decision tree without making classification errors so worse. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Data mining; Machine learning; Abstraction; Classification

* Corresponding author.

E-mail addresses: kudo@db-ei.eng.hokudai.ac.jp (Y. Kudoh), makoto@db-ei.eng.hokudai.ac.jp (M. Haraguchi), yoshiaki@db-ei.eng.hokudai.ac.jp (Y. Okubo).

PII: S0304-3975(02)00178-0

1. Introduction

Since the late 1980s, studies on *Knowledge Discovery in Databases* (*KDD*) have been paid much attentions. Briefly speaking, a KDD process can be divided into four subprocesses [5,4]: (1) *Data selection*, (2) *Data cleaning and pre-processing*, (3) *Data mining* and (4) *Interpretation and evaluation*. The third process, data mining, is especially considered as a central one for extracting useful knowledge from large databases very efficiently. Therefore, many KDD studies have concentrated on developing various methods for the task. However, it is also well known that they often detect meaningless rules that do not meet a user's intention.

One reason seems to lie in a fact that some data irrelevant to the user's intention still remains in the database on which mining processes are carried out. In this case, the second process, data cleaning and pre-processing, must be useful to exclude the irrelevant data.

As another reason why meaningless rules are often detected, we can consider that data values in the database might be too concrete or too detailed. In order to detect useful knowledge from such a database, some computation with high complexity would be required in general. Furthermore, the detected knowledge would not be easy to understand because of its too detailed description.

To overcome the former problem of extracting meaningless rules by the influence of irrelevant data, the literatures [1,5] have proposed to use a database query language like SQL to specify a part of database with which the mining process is concerned. However, such an SQL approach is too strict because a user must specify (identify) which part of database is relevant to his/her mining problem beforehand.

On the other hand, for the latter problem of granularity, *data abstraction* is considered to be useful. As one of this kind of strategies, a *generalization of database* under a given concept hierarchy has been proposed in the method of *Attribute-Oriented Induction* [6] implemented in *DBMiner* [7], to generalize a database and to prevent KDD processes from extracting meaningless knowledge. In the generalization approach, however, a user or a system administrator is required to have good domain knowledge to provide just one appropriate concept hierarchy before mining processes. It would be a hard task for users who are not expert in the domain.

From these perspectives, we consider that the following functions are necessary to obtain meaningful knowledge from databases:

- To predict user's intention by fewer queries and to focus on the important relationships or structures among data.
- To automatically select a data abstraction that is adapted for the target which the user wants to discover and to generalize databases by the selected data abstraction.

In this paper, we are especially concerned with the second problem on *automatic selection of abstractions*. Here the generalization of a database means an act of replacing the data values in the original database with more abstract values. It should be noted here that some distinguishable data in the original database might be identified by the generalization. Such a generalization directly reflects the accuracy of extracted knowledge from the generalized database. If some significant differences among data values are missed by the generalization, then we would lose a chance to find significant rules from the generalized database. Therefore, it is very much important to perform an appropriate generalization according to the user's intention.

Although there may exist various ways to define the concise notion of the user's intention, we assume in this paper that a user intends to have a more understandable *decision tree* [20] whose accuracy is high. More precisely speaking, given target classes, the understandability and the accuracy are measured by the number of nodes in the decision tree and by its error rate with respect to the target class, respectively. In order to perform a generalization as to such a criterion about the user's intention, we have to carefully select an appropriate abstraction under which the error rate of the decision tree is not increased and the size of the tree is reduced. The decision tree detected from the generalized database according to such appropriate abstractions will become more compact and have the classification ability approximately equivalent to one before generalization.

Although the final goal is to present a method for automatically generating such a good generalization, it is generally understood as a hard task to synthesize a generalization hierarchy from scratch, as known in the fields of *natural language processing* and *information retrieval*.

Therefore, this paper tries to solve a problem of *selecting* an appropriate hierarchy among possible ones. More concretely, we consider the possible hierarchies as *layered hierarchies* each of which is defined as a grouping of attribute values in the original database, and propose a method for selecting an appropriate grouping (abstraction) from such ones. In this paper, therefore, the problem of generalizing database is regarded as the problem of selecting abstraction of attribute values.

Given a relational database, assume, for each attribute in the database, that we have a class of possible groupings (abstractions) of the values. For given target classes (attribute), if the class distribution can be preserved as much as possible even after generalizing a database according to the abstraction, the abstraction is preferred and selected as an appropriate one. In other words, if some attribute values share almost the same or a similar class distribution, they are considered not to have any significant difference about the class. This implies that we do not need to distinguish them for our classification task. Therefore, these values can be abstracted into a single abstract value. On the other hand, if they have distinguishable class distributions, the difference would be significant to perform the classification in terms of attribute values. Hence, the difference should be regarded after any abstraction.

The classification process under some attribute yields branches whose number is equal to the number of values of the attribute. By generalizing the original database according to an appropriate abstraction, many attribute values are identified as the same abstract one. This means that the number of branches for the generalized database is less than one for the original database when we classify each database under the same attribute. For example, in case of a database shown in Fig. 1, attributes values `US, Canada, Thailand, Japan,` ... of an attribute `native_country` is abstracted into abstract concepts `North_America, Asia,` ... . It is obvious that the number of branches corresponding to `North_America, Asia, ...` is less than an original one before generalization. Therefore, it is expected that the size of decision tree can be reduced by the abstraction.

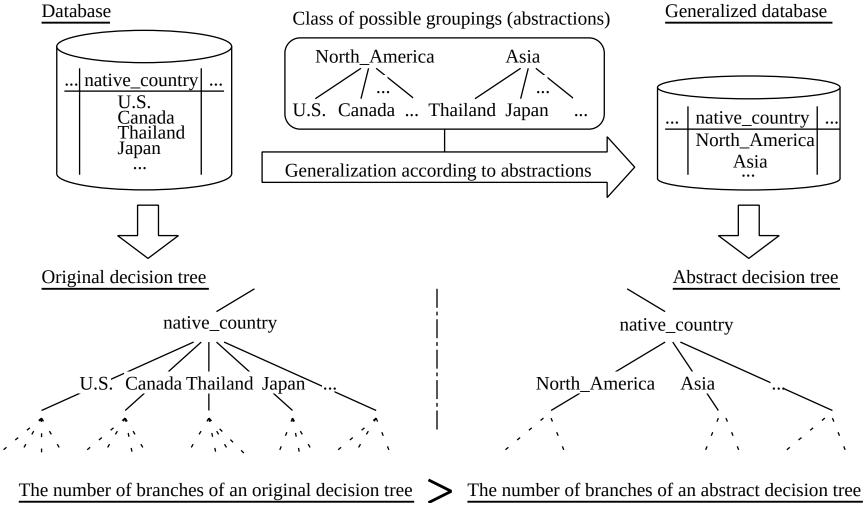

Database
... native_country ...
U.S.
Canada
Thailand
Japan
...
Class of possible groupings (abstractions)
North_America
Asia
U.S. Canada ... Thailand Japan ...
Generalization according to abstractions
Generalized database
... native_country ...
North_America
Asia
...
Original decision tree
native_country
U.S. Canada Thailand Japan ...
Abstract decision tree
native_country
North_America Asia ...
The number of branches of an original decision tree > The number of branches of an abstract decision tree

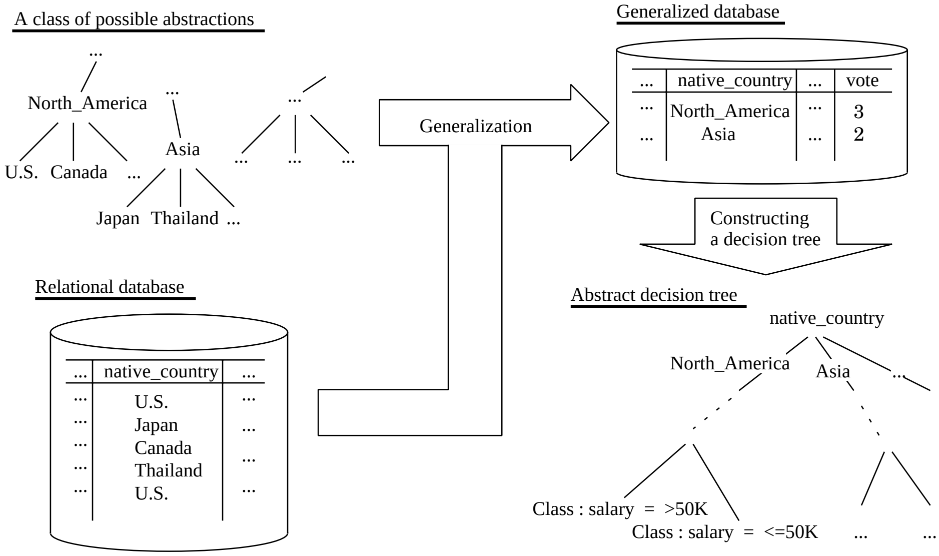
Fig. 1. A reduction of the number of branches.

Our appropriateness of abstraction is precisely defined in terms of *information theory* (e.g. [2]). Therefore, our abstraction framework is called *Information Theoretical Abstraction* (*ITA*). More concretely, in order to measure the similarity among class distributions, we adopt the notions of *mutual information* and *split information* proposed in C4.5 [20]. In a word, if the class distributions before abstraction are very close and are similar with respect to some metric, the distribution after abstraction, which is the mean of those original distributions, is also close to them. The amount of information about classes is therefore approximately preserved before and after the abstraction. In addition, even if we have some counter distribution not close to some cluster of similar distributions, such an exceptional one can be disregarded provided its probability is low. Consequently, we can say that our abstraction method tries to comprehend a global rule by integrating the distributions in the cluster so as to preserve the amount of information at least approximately, ignoring some exception.

We show some experimental results obtained by a system ITA that is an implementation of our abstraction method. Fig. 2 illustrates an overview of ITA system. Given a relational database, a target class (attribute) and a class of possible abstractions, ITA first selects an appropriate abstraction for each attribute from the possible ones. Then the system generalizes the original database according to the selected abstractions. The generalized database is given to C4.5 to construct an abstract decision tree.

The abstract tree is compared to the decision tree constructed from the original database in points of the size and error rate. The results show that the size of abstract tree is much smaller than the other and the error rate is still approximately equal to the original one.

This paper is organized as follows. In the next section, we gives preliminaries. Section 3 introduces some terminologies used throughout this paper, and analyzes some relationships between the appropriateness of abstractions and the mutual information.



...	native_country	...
...	U.S.	...
...	Japan	...
...	Canada	...
...	Thailand	...
...	U.S.	...

...	native_country	...	vote
...	North_America	...	3
...	Asia	...	2

Fig. 2. Overview of ITA.

Section 4 discuss the notion of clusters of similar distributions, and explain why our abstraction strategy contributes to the size reduction of decision trees without making their precision worse. In Section 5, we present an algorithm for generalizing a database under selected appropriate abstractions. Section 6 shows some experimental results on census database in US Census Bureau and discusses them. In Section 7, we conclude this paper with a summary and important future works.

2. Preliminaries

The type of data to which we try to apply our notion of data abstractions is specified by a relation schema $R(A_1,\ldots,A_m)$, where A_i is an attribute with its domain $dom(A_i)$, a non-empty set of attribute values, and R is a relation name. An instance of $R(A_1,\ldots,A_m)$ is a relation $R \subseteq dom(A_1) \times \cdots \times dom(A_m)$, where we identify a relation with its name for our notational convenience. R is called an instance relation of the schema. Each element $t \in R$ is a tupple $(v_1,\ldots,v_m)$ whose jth component is called an A_j-value of t, and is denoted by $t[A_j]$. Thus, $t=(v_1,\ldots,v_m)=(t[A_1],\ldots,t[A_m])$.

In addition to these attributes $A_1,\ldots,A_m$, we assume just one target attribute C whose value is called a class. So, the expression $t[C]=c$ means that the tupple t has a class c as its C-value. Our data mining task for this type of data is to find several conditions in terms of attributes $A_1,\ldots,A_m$ for discriminating or for characterizing the classes in C. Moreover, we suppose a family of data mining algorithms using posterior class distributions, e.g. ID3 and C4.5. The probability space used by these systems is $(R,\Pr)$, where the probability is given by a uniform distribution.

$$\Pr(\text{tupple } t) = 1/\text{the number of tupples in } R.$$

Then, each attribute A_j and the target one C as well are regarded as random variables.

$$\begin{aligned}\Pr(X = x) &= \Pr(\{t \in R \mid t[X] = x\})\\ &= \frac{\text{the number of tupples in } R \text{ whose } X\text{-value is } x}{\text{the number of tupples in } R}.\end{aligned}$$

When it is clear from the context, the expression $\Pr(X = x)$ is often denoted by $\Pr(x)$. Moreover, a distribution is represented by a probability vector $(p_1, \dots, p_n)$ with $0 \leqslant p_i \leqslant 1$ and $\sum_{j=1}^{n} p_j = 1$. The property about classes $c_1, \dots, c_n$ captured by an attribute value $a \in dom(A_j)$ is represented by a conditional distribution

$$(\Pr(C = c_1 | A_j = a), \dots, \Pr(C = c_n | A_j = a))$$

given A_j-value a, and is also called a posterior class distribution.

3. Data abstraction

In this section, we introduce our notion of data abstractions, discuss their information loss, and finally present our fundamental criterion to select the best abstraction called an appropriate abstraction. Let us start with defining the data abstractions.

3.1. Definition of data abstractions

Suppose that we have n classes $c_1, \dots, c_n$ and a relation R with its attributes $A_1, \dots, A_m$.

Definition 3.1 (*Data Abstraction*). (1) A data abstraction φ_A for an attribute A in $\{A_1, \dots, A_m\}$ is defined as a partition $\{g_1, \dots g_\ell\}$ of $dom(A)$, where $g_j \subseteq dom(A)$, $g_i \cap g_j = \phi$ whenever $i \neq j$, and $dom(A) = \bigcup_{j=1}^{\ell} g_j$. g_j is called a group of A-values.

(2) A data abstraction φ for attributes $A_1, \dots, A_m$ is defined as a tupple $(\varphi_{A_1}, \dots, \varphi_{A_m})$, where φ_{A_j} is a data abstraction for the attribute A_j.

(3) Given a data abstraction $\varphi = (\varphi_{A_1}, \dots, \varphi_{A_m})$ and a relation R with the same attributes $A_1, \dots, A_m$, we form a generalized relation $\varphi(R)$ which has its attributes $\overline{A_1}, \dots, \overline{A_m}$ with their domains $dom(\overline{A_j}) = \varphi_{A_j}$. Thus, each group $g \in \varphi_{A_j}$ is regarded as an abstract attribute value of $\overline{A_j}$. Moreover, a tupple $t = (a_1, \dots, a_m) \in R$ is called an instance (tupple) of an abstract tupple $\bar{t} = (g_1, \dots, g_m)$ if $a_j \in g_j$ holds for any j. Conversely, for a given $t = (a_1, \dots, a_m) \in R$, t is said to be abstracted to $\bar{t} = (g_1, \dots, g_m)$ with $a_j \in g_j \in \varphi_{A_j}$, where $\bar{t}$ is uniquely determined as there exists just one $g_j \in \varphi_{A_j}$ such that $a_j \in g_j$. Then, $\varphi(R)$ is defined as a set of abstract tupples $\bar{t}$ such that there exists at least one instance tupple in R. In other words, $\varphi(R)$ consists of abstract tupples to

X	Y	$\bar{X}$	$\bar{Y}$	*vote*
a	25	g_{ab}	*over*_10	2
c	15	g_{cde}	*over*_10	1
d	4	g_{cde}	*under*_10	1
b	35			

Fig. 3. Relations before and after data abstraction.

which some $t \in R$ is abstracted.

$$\varphi(R) = \{(g_1, \ldots, g_m) \mid \text{there exists } (a_1, \ldots, a_m) \in R \text{ such that } a_j \in g_j \in \varphi_j\}.$$

To calculate the probability of abstract tupples, we use the following function.

$$\mathit{vote}(\bar{t}) = \text{the number of instance tupples of } \bar{t}.$$

We here illustrate the generalization of a relation by a simple example. Suppose that we have a relation with two attributes X and Y, shown at the left of Fig. 3, and a data abstraction (φ_X, φ_Y), where $\varphi_X = \{g_{ab} = \{a, b\},\ g_{cde} = \{c, d, e\}\}$, and $\varphi_Y = \{\mathit{under}\_10 = \{x \mid x < 10\},\ \mathit{just}\_10 = \{10\},\ \mathit{over}\_10 = \{x \mid 10 < x\}\}$. For instance, the expression $g_{ab} = \{a, b\}$ means that $\{a, b\}$ is a group and g_{ab} is its name. In what follows, we do not distinguish any group from its name. Then we see the generalized relation at the right of Fig. 3. For example, $(a, 25)$ is abstracted to $(g_{ab}, \mathit{over}\_10)$, as $a \in g_{ab}$ and $25 \in \mathit{over}\_10$.

An abstract attribute $\overline{A_j}$ corresponding to A_j can be also considered as a random variable. In fact, for a group $g \in \varphi_{A_j}$,

$$\Pr(\overline{A_j} = g) = \Pr(\{t \in R \mid t \text{ is abstracted to } \bar{t} \text{ whose } \overline{A_j}\text{-value is } g\}).$$

Then, the following fact is often used, as it asserts a fundamental relationship among attributes before and after our data abstraction.

Proposition 3.1. *For a group $g \in \varphi_{A_j} = \mathit{dom}(\overline{A_j})$, the event specified by $\overline{A_j} = g$ is just a disjoint union of all events given by $A_j = a$ for $a \in g$. Hence*, $\Pr(\overline{A_j} = g) = \sum_{a \in g} \Pr(A_j = a)$.

This is simply because

$$\begin{aligned} &\{t \in R \mid t \text{ is abstracted to } \bar{t} \text{ whose } \overline{A_j}\text{-value is } g\} \\ &\quad = \{t \in R \mid t[A_j] = \text{some } a \in g\} = \bigcup_{a \in g} \{t \in R \mid t[A_j] = a\}, \quad \text{a disjoint union.} \end{aligned}$$

3.2. Abstract class distributions

Needless to say, there may exist a lot of possible data abstractions, so the major problem is to present a criterion for selecting the best one according to which the generalization of relations should be carried out. As several attributes are generally interrelated each other, the best abstraction for one attribute may be dependent on another attribute. However, in this paper, we choose the best one independent of another attribute, and actually present a criterion to select the best data abstraction φ_A for each attribute A. In what follows, we thus concentrate on an attribute A arbitrarily chosen from $\{A_1,\ldots,A_m\}$ and its abstraction φ_A. Even when the interrelation among attributes should be taken into account, we can provide a revised criterion which we briefly discuss in the last section.

Our criterion is concerned with how a class distribution given by

$$(\Pr(C = c_1|A = a),\ldots,\Pr(C = c_n|A = a))$$

is changed after data abstractions. We consider that the abstraction with the smaller changes is better.

The class distribution after a data abstraction φ_A, which is called an abstract class distribution, is obtained by extending the conditioning by a single A-value to the conditioning by a set of A-values. Here, the conditioning is specified by a group $g_j = \{a_{j_1},\ldots,a_{j_{n_j}}\} \in \varphi_A$, and means information that A-value is some a_{j_i} in that group g_j. From Proposition 3.1, this statement is equivalent to $\bar{A} = g_j$. So the conditioning is denoted as $\bar{A} = g_j$.

Definition 3.2 (*Abstract Class Distribution*). Given a group g in a data abstraction φ_A for an attribute A, the distribution

$$\begin{aligned} C_{\bar{A}=g} &= (\Pr(C = c_1|\bar{A} = g),\ldots,\Pr(C = c_n|\bar{A} = g)) \\ &= (\Pr(C = c_1|A \in g),\ldots,\Pr(C = c_n|A \in g)) \end{aligned}$$

is called an abstract (class) distribution given $\bar{A} = g$.

Based on Bayes' theorem, such an abstract distribution can be represented as a linear combination of original distributions, as shown by the following formula transformations:

$$\begin{aligned} \Pr(C = c_j \mid \bar{A} = g) &= \frac{\Pr(\{t \in R \mid t[C] = c_j\} \cap \bigcup_{a \in g} \{t \in R \mid t[A] = a\})}{\Pr(\bigcup_{a \in g} \{t \in R \mid t[A] = a\})} \\ &= \frac{\sum_{a \in g} \Pr(C = c_j, A = a)}{\sum_{a \in g} \Pr(A = a)} \\ &= \frac{\sum_{a \in g} \Pr(A = a)\Pr(C = c_j|A = a)}{\sum_{a \in g} \Pr(A = a)} \end{aligned}$$

$$= \sum_{a\in g} \lambda_a \Pr(C = c_j|A = a),$$

$$\text{where } \lambda_a = \frac{\Pr(A = a)}{\sum_{a\in g} \Pr(A = a)}. \qquad (3.1)$$

Since the equation holds for any c_j, we have the following fact.

Proposition 3.2. *For any attribute A, a data abstraction φ_A for A and $g \in \varphi_A$,*

$$C_{\bar{A}=g} = \sum_{a\in g} \lambda_a C_{A=a}, \quad \textit{where } 0 \leqslant \lambda_a = \Pr(A = a|\bar{A} = g) \leqslant 1 \textit{ and } \sum_{a\in g} \lambda_a = 1.$$

Thus, a data abstraction φ_A consisting of k groups $g_1,\dots,g_k$, forms k abstract distributions $C_{\bar{A}=g_1},\dots,C_{\bar{A}=g_k}$, each of which is a mean of class distributions in $\{C_{A=a} \mid a \in g_j\}$.

As a simple example, $\varphi_{\texttt{nc}} = \{\{$`Japan, Thailand|, ...`$\}$, $\{$`US, Canada,...`$\}$,...$\}$ is a possible data abstraction for `native_country` attribute abbreviated as `nc`. The first group $\{$`Japan, Thailand, ...`$\}$ means Asian countries, and the second group denotes North American countries. So, when we consider `salary` attribute as a class attribute, then $\texttt{salary}_{\overline{\texttt{nc}}=\texttt{Asia}}$ is an abstract distribution which is the expectation of salary distributions of Asian countries $\texttt{salary}_{\texttt{nc}=\texttt{Japan}}$, $\texttt{salary}_{\texttt{nc}=\texttt{Thailand}}$ and so on, where the weight is

$$\Pr(\texttt{nc} = \texttt{Japan}|\overline{\texttt{nc}} = \texttt{Asia}) = \frac{\text{the number of tupples whose } \texttt{nc}\text{-value is Japan}}{\text{the number of tupples whose } \texttt{nc}\text{-value is some Asian country}}$$

for example. We similarly have another salary distribution, $\texttt{salary}_{\overline{nc}} = \texttt{North America}$, of north American countries.

3.3. Information loss due to data abstractions

In this section, we analyze some relationships between the appropriateness of our data abstractions and their information loss measured by mutual information.

As we have just observed, an abstract distribution $C_{\bar{A}=g}$ is a mean of original class distributions $C_{A=a}$ such that $a \in g$. Hence, some information or properties possessed by the individual distributions $C_{A=a}$ may be lost in the abstract distribution. In the worst case, for example, two distributions of two classes $C_{A=a_1} = (1,0)$ and $C_{A=a_2} = (0,1)$ are abstracted to their average $C_{\bar{A}=\{a_1,a_2\}} = (0.5, 0.5)$, provided $\Pr(A = a_1) = \Pr(A = a_2)$. The first two show that if $A = a_j$ then $C = c_j$ with the probability 1, while the abstract distribution is uniform and has no characteristics about the classes. Thus, the actual information about classes has disappeared in the abstraction.

To measure the amount of information loss, we use Shannon's mutual information and discuss what information is actually preserved in our data abstraction when the amount of information loss is small. It should be noted here that, in the case of ID3 and C4.5, the mutual information measures a kind of *information gain for attribute selections*, while in this paper, it does *information loss for abstraction selections.*

More precisely, we compute the subtraction of mutual information before and after the data abstraction.

The mutual information $I(C;A)$ before abstraction: For a given relation R with attributes $\{A_1,\dots,A_m\}$, an attribute $A\in\{A_1,\dots,A_m\}$ and a target attribute C,

$$\begin{aligned}
&I(C;A)=H(C)-H(C|A),\\
&H(C)=\sum_{c\in dom(C)} L(\Pr(C=c)),\quad \text{where } L(x)=\begin{cases}-x\log_2 x & \text{if } 0<x\leqslant 1\\ 0 & \text{if } x=0\end{cases},\\
&H(C|A)=\sum_{a\in dom(A)} H(C|A=a),\\
&H(C|A=a)=\sum_{c\in dom(C)} L(\Pr(C=c|A=a)).
\end{aligned}$$

From the definition, we have an equation,

$$I(C;A)=\sum_{a\in dom(A)} \Pr(A=a)(H(C)-H(C|A=a)), \tag{3.2}$$

meaning that $I(C;A)$ is the expectation of reduction of entropy observing A-values.

The mutual information $I(C;\bar{A})$ after abstraction: The relation R with its attributes $A_1,\dots,A_m$ is generalized to $\varphi(R)$ with its attributes $\overline{A_1},\dots,\overline{A_m}$, where $dom(\overline{A_j})=\varphi_{A_j}$. Hence, the mutual information after the abstraction is

$$I(C;\bar{A})=\sum_{g\in\varphi_A} \Pr(\bar{A}=g)(H(C)-H(C|\bar{A}=g)). \tag{3.3}$$

The information loss due to φ_A: According to the information theory (see [2], for instance), the following inequalities hold.

$$H(C|A)\leqslant H(C|\bar{A}) \quad\text{and}\quad I(C;A)\geqslant I(C;\bar{A}).$$

Hence we can define the information loss due to φ_A given by

$$e(\varphi_A)=I(C;A)-I(C;\bar{A})=H(C|\bar{A})-H(C|A).$$

More precisely,

$$\begin{aligned}
e(\varphi_A)&=H(C|\bar{A})-H(C|A)\\
&=\sum_{g\in\varphi_A}\Pr(\bar{A}=g)H(C|\bar{A}=g)-\sum_{a\in dom(A)}\Pr(A=a)H(C|A=a)\\
&=\sum_{g\in\varphi_A}\Pr(\bar{A}=g)H(C|\bar{A}=g)-\sum_{g\in\varphi_A}\sum_{a\in g}\Pr(A=a)H(C|A=a)\\
&=\sum_{g\in\varphi_A}\left(\Pr(\bar{A}=g)H(C|\bar{A}=g)-\sum_{a\in g}\Pr(A=a)H(C|A=a)\right)
\end{aligned}$$

$$= \sum_{g \in \varphi_A} \Pr(\bar{A} = g) \left(H(C|\bar{A} = g) - \sum_{a \in g} \frac{\Pr(A = a)}{\Pr(\bar{A} = g)} H(C|A = a) \right)$$

(use λ_a in (3.1))

$$= \sum_{g \in \varphi_A} \Pr(\bar{A} = g) \left(H(C|\bar{A} = g) - \sum_{a \in g} \lambda_a H(C|A = a) \right)$$

$$\left(\text{let } e(g; \varphi_A) = H(C|\bar{A} = g) - \sum_{a \in g} \lambda_a H(C|A = a) \right)$$

$$= \sum_{g \in \varphi_A} \Pr(\bar{A} = g) e(g; \varphi_A).$$

$e(g; \varphi_A)$ is the loss of information due to the group g in φ_A. So the whole loss $e(\varphi_A)$ of φ_A is the summation of the loss $e(g; \varphi_A)$ for g.

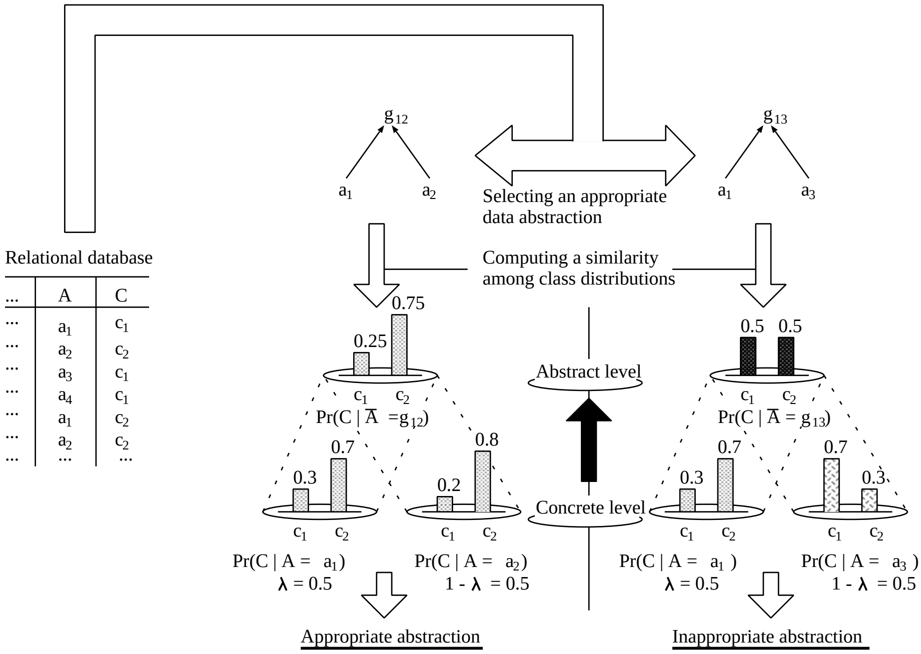
$$\begin{aligned}
e(g; \varphi_A) &= H(C|\bar{A} = g) - \sum_{a \in g} \lambda_a H(C|A = a) \\
&= \sum_{c \in dom(C)} L(\Pr(C = c|\bar{A} = g)) \\
&\quad - \sum_{a \in g} \lambda_a \sum_{c \in dom(C)} L(\Pr(C = c|A = a)) \\
&= \sum_{c \in dom(C)} \left(L(\Pr(C = c|\bar{A} = g)) - \sum_{a \in g} \lambda_a L(\Pr(C = c|A = a)) \right) \\
&= \sum_{c \in dom(C)} \left(L\left(\sum_{a \in g} \lambda_a \Pr(C = c|A = a) \right) - \sum_{a \in g} \lambda_a L(\Pr(C = c|A = a)) \right) \\
&\quad \text{(again from Eq. (3.1))} \\
&= \sum_{c \in dom(C)} e(c, g; \varphi_A) \quad \text{(we define } e(c, g; \varphi_A) \text{ as}
\end{aligned}$$

$$L\left(\sum_{a \in g} \lambda_a \Pr(C = c|A = a) \right) - \sum_{a \in g} \lambda_a L(\Pr(C = c|A = a)). \tag{3.4}$$

Thus, $e(g; \varphi_A)$ is the summation of $e(c, g; \varphi_A)$, the difference between the L-value of the mean of $\Pr(c|a)$ and the mean of L-value of $\Pr(c|a)$.

Here it should be noted that $H(C|\bar{A} = g) \leqslant \sum_{a \in g} \lambda_a H(C|A = a)$. Firstly, $H(C|Y = y)$ is the entropy of distribution $D = (\Pr(C = c_1|Y = y), \ldots, \Pr(C = c_n|Y = y))$, which is also denoted by $H(D)$. By this notation, we have $H(C|\bar{A} = g) = H(C_{\bar{A} = g})$ and $H(C|A = a) = H(C_{A = a})$. Finally, it is an well known property of the entropy function that

$$H\left(\sum_{j=1}^{k} \zeta_j D_j \right) \geqslant \sum_{j=1}^{k} \zeta_j H(D_j) \tag{3.5}$$



...	A	C
...	a1	c1
...	a2	c2
...	a3	c1
...	a4	c1
...	a1	c2
...	a2	c2
...	...	...

Fig. 4. An appropriate abstraction and an inappropriate one.

for any distribution D_j and a probability vector $(\zeta_1,\ldots,\zeta_k)$. We therefore have $H(C|\bar{A}=g)\geqslant\sum_{a\in g}\lambda_a H(C|A=a)$.

To evaluate the loss $e(g;\varphi_A)$ and to establish the basic relationship between the approximation of distributions and the information loss, we need the following definition and proposition.

Definition 3.3. (1) For two distributions $D_j=(p_1^j,\ldots,p_n^j)(j=1,2)$, the metric between D_1 and D_2 is given by

$$|D_1-D_2| = \max_{1\leqslant k\leqslant n}|p_k^1-p_k^2|. \tag{3.6}$$

Moreover, D_1 and D_2 are said δ-similar, denoted by $D_1\sim_\delta D_2$, if $|D_1-D_2|<\delta$.

(2) The diameter of class distributions $\mathscr{C}=\{C_1,\ldots,C_\ell\}$ is defined as $\max_{i,j}|C_i-C_j|$, and is denoted by $diam(\mathscr{C})$.

The following is a direct consequence of the definition, since $C_{\bar{A}=g}$ is a mean of $C_{A=a}$ in $\{C_{A=a}\,|\,a\in g\}$.

Proposition 3.3. (1) *For any $D_1,D_2\in\mathscr{C}$, D_1 and D_2 are $diam(\mathscr{C})$-similar.*

(2) *If $diam(\{C_{A=a}\,|\,a\in g\})<\delta$, then $C_{\bar{A}=g}$ is δ-similar to any $C_{A=a}$ in $\{C_{A=a}\,|\,a\in g\}$.*

Our first selection principle, illustrated by Fig. 4, is based on this simple proposition.

Preservingness of distributions: Select a data abstraction $\varphi=(\varphi_{A_1},\ldots,\varphi_{A_m})$ such that, for every $g\in\varphi_{A_j}$, any two distributions in $\{C_{A=a}\mid a\in g\}$ to be abstracted to an abstract distribution $C_{\bar{A}=g}$ are δ-similar with a small δ. Then distribution $C_{\bar{A}=g}$ after the abstraction is also similar to any distribution in $\{C_{A=a}\mid a\in g\}$ within the same error δ. As a result, the properties or characteristics about classes captured by the distributions are approximately preserved and are not lost in the abstraction.

To guarantee the first principle, we use the measure of information loss according to Proposition 3.4.

Proposition 3.4. *There exists a continuous function* $h(\delta)$ *defined on* $[0,1]$ *such that*
(1) $h(\delta_1)<h(\delta_2)$ *whenever* $\delta_1<\delta_2$,
(2) $h(\delta)\to 0$ *as* $\delta\to 0$, *and*
(3) $e(g;\varphi_A)\leqslant h(diam(\{C_a\mid a\in g\}))$.

Proof. Let $\delta=diam(\{C_a\mid a\in g\})$. Then, for an arbitrary chosen class $c\in dom(C)$, we first evaluate $e(c,g;\varphi_A)$. Let $d_1=\min_{a\in g}\Pr(c|a)$, $d_2=\max_{a\in g}\Pr(c|a)$ and represent each $\Pr(c|a)$ by $\zeta_a d_1+(1-\zeta_a)d_2$ where $0\leqslant\zeta_a\leqslant 1$.

Since $L(x)=-x\log_2 x$ is a concave function, we have

$$L\left(\sum_a \lambda_a \Pr(c|a)\right)\geqslant\sum_a\lambda_a L(\Pr(c|a)) \tag{3.7}$$

and

$$L(\Pr(c|a))\geqslant\zeta_a L(d_1)+(1-\zeta_a)L(d_2).$$

Hence

$$\begin{aligned}\sum_a\lambda_a L(\Pr(c|a))&\geqslant\sum_a\lambda_a(\zeta_a L(d_1)+(1-\zeta_a)L(d_2))\\&\geqslant\left(\sum_a\lambda_a\zeta_a\right)L(d_1)+\left(\sum_a\lambda_a(1-\zeta_a)\right)L(d_2)\\&=\left(\sum_a\lambda_a\zeta_a\right)L(d_1)+\left(1-\sum_a\lambda_a\zeta_a\right)L(d_2).\end{aligned}\tag{3.8}$$

Similarly, we have $\sum_a\lambda_a\Pr(c|a)=(\sum_a\lambda_a\zeta_a)d_1+(1-\sum_a\lambda_a\zeta_a)d_2$.

Hence, from (3.7) and (3.8), we have

$$\begin{aligned}&e(c,g;\varphi_A)\\&\quad\leqslant L\left(\left(\sum_a\lambda_a\zeta_a\right)d_1+\left(1-\sum_a\lambda_a\zeta_a\right)d_2\right)-\left(\sum_a\lambda_a\zeta_a\right)L(d_1)\\&\qquad-\left(1-\sum_a\lambda_a\zeta_a\right)L(d_2).\\&\quad\leqslant\max_{0\leqslant\zeta\leqslant 1}(L(\zeta d_1+(1-\zeta)d_2)-\zeta L(d_1)-(1-\zeta)L(d_2))\end{aligned}$$

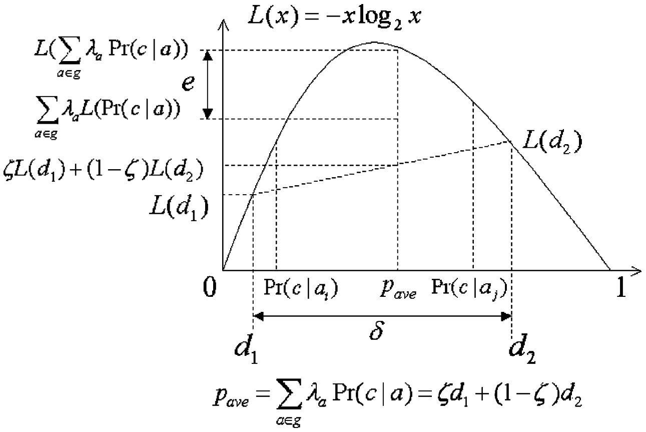

$L(x) = -x\log_2 x$
$L(\sum_{a\in g}\lambda_a \Pr(c\,|\,a))$
e
$\sum_{a\in g}\lambda_a L(\Pr(c\,|\,a))$
$\zeta L(d_1)+(1-\zeta)L(d_2)$
$L(d_1)$
$L(d_2)$
0
$\Pr(c\,|\,a_i)$
p_{ave}
$\Pr(c\,|\,a_j)$
1
δ
d_1
d_2
$p_{ave} = \sum_{a\in g}\lambda_a \Pr(c\,|\,a) = \zeta d_1 + (1-\zeta)d_2$

Fig. 5. Evaluation of the information loss $e(c, g; \varphi_A)$.

$$\leqslant \max_{\substack{|d_1-d_2|\leqslant\delta \\ 0\leqslant d_1\leqslant d_2\leqslant 1}} \max_{0\leqslant\zeta\leqslant 1} (L(\zeta d_1 + (1-\zeta)d_2)$$

$$-\zeta L(d_1) - (1-\zeta)L(d_2)). \tag{3.9}$$

Let $F(\delta)$ be the rightmost formula of the inequality (3.9). Now define the function h by

$$h(\delta) = nF(\delta),$$

where n is the cardinality of C. Then, it is clear from the definition that h satisfies the requirements. □

The proposition asserts that, as the distributions $\{C_{A=a}\,|\,a\in g\}$ to be abstracted to $C_{\bar{A}=g}$ are δ-similar with a smaller error δ, the information loss $e(g;\varphi_A)$ is bounded by a lower upperbound and is therefore smaller. Thus, our actual selection principle proposed in this paper can be simply stated as follows (see Fig. 5).

Minimum information loss: Choose a data abstraction whose information loss is the minimum among a class of possible abstractions.

As we see in the next section, the principle of minimizing the information loss covers the preservingness of distributions in the first criterion, allowing some exceptional distributions whose probability is low. A family of distributions with such an exception will be analyzed by a notion of clusters (Definition 4.2 in the next section).

A data abstraction actually chosen by this selection criterion depends on what space of possible abstractions we examine. The most general space is the lattice of all partitions except the trivial one, $\{\{a\}\,|\,a\in dom(A)\}$, whose information loss is always 0. As the lattice size is exponential, we will present in Section 6 its subspace and an algorithm running in it so that finding the best abstraction is computationally tractable.

Although the algorithm uses additional parameters and heuristics to improve its performance, the major factor to choose data abstractions is the minimization of information loss.

4. Clusters of distributions

As we have mentioned in Section 2, our data mining task is to characterize classes $c_1,\dots,c_n$ in terms attributes $\{A_1,\dots,A_m\}$. For this purpose, we suppose decision trees, as in ID3 and C4.5, and compares them at the two levels, concrete and abstract ones. As the decision trees consist of paths describing conditions for classifying those classes, we analyze in this section the quality of abstract paths in an abstract decision tree. Particularly, we evaluate the classification accuracy of paths by the entropy of distributions associated with them. An abstract path represents a family of instance paths at a concrete level, so its entropy is generally higher than the expectation of entropy of the concrete level distributions. However, we show that, if the abstract path has a highly weighted cluster of its instance distributions at the concrete level, the abstract path has its entropy which is closer to the entropy at the concrete level. So the accuracy is almost preserved in the abstraction.

Definition 4.1. Suppose a relation R with attributes $\{A_1,\dots,A_m\}$ and an abstraction $\varphi=(\varphi_{A_1},\dots,\varphi_{A_m})$.

A (concrete level) path p is a sequence $(A_{i_1}=a_{i_1},\dots,A_{i_k}=a_{i_k})$ of expressions of the form $A=a$ such that $a\in dom(A)$ and $A\in\{A_1,\dots,A_m\}$. k is called the length of p. Similarly, an abstract path is defined as a sequence $\bar{p}=(\overline{A_{i_1}}=g_{i_1},\dots,\overline{A_{i_k}}=g_{i_k})$, where $g_{i_j}\in\varphi_{A_{i_j}}=dom(\overline{A_{i_j}})$. When a concrete level path $p=(A_{i_1}=a_{i_1},\dots,A_{i_k}=a_{i_k})$ satisfies $a_{i_j}\in g_{i_j}$ for any j, then we say that p is an instance of $\bar{p}$ or that p is abstracted to $\bar{p}$ w.r.t. φ. $inst_\varphi(\bar{p})$ denotes the set of all instances of $\bar{p}$.

The concatination $p\cdot(X=x)$ of a path $p=(X_{i_1}=x_{i_1},\dots,X_{i_k}=x_{i_k})$ and an expression $(X=x)$ is also a path $(X_{i_1}=x_{i_1},\dots,X_{i_k}=x_{i_k},X=x)$. Moreover, we allow the empty path, ϕ, with its length 0.

A (concrete level) decision tree is inductively defined as a set of paths as follows:

(DT1) The set $\{\phi\}$ is a decision tree with root node only.

(DT2) Suppose that $\mathscr{D}T$ is a decision tree, and consider a path $p\in\mathscr{D}T$ and an attribute A not appearing in p. Then $\mathscr{D}T\cup\{p\cdot(A=a)\mid a\in dom(A)\}$ is a decision tree.

An abstract decision tree is similarly defined using abstract attributes and their values.

A path $p=(A_{i_1}=a_{i_1},\dots,A_{i_k}=a_{i_k})$ denotes an event $\bigcap_j\{t\in R\mid t[A_{i_j}]=a_{i_j}\}$. Similarly, an abstract path $\bar{p}=(\overline{A_{i_1}}=g_{i_1},\dots,\overline{A_{i_k}}=g_{i_k})$ means an event

$$\{t\in R\mid t[A_{i_j}]\in g_{i_j}\ \text{ for any } j\}=\bigcup_{\substack{a_{i_j}\in g_{i_j}\\ 1\leqslant j\leqslant k}}\{t\in R\mid t[A_{i_j}]=a_{i_j}\ \text{ for any } j\},$$

so we have

$$\Pr(\bar{p}) = \sum_{p \in inst_\varphi(\bar{p})} \Pr(p),$$

where $inst_\varphi(\bar{p})$ is defined as the set of all instance paths of $\bar{p}$. This is an extension of Proposition 3.1.

For a path $p=(A_{i_1}=a_{i_1},\dots,A_{i_k}=a_{i_k})$, a class distribution after observing A_{i_j}-value a_{i_j} is given by

$$C_p = (\Pr(C=c_1|p),\dots,\Pr(C=c_n|p)).$$

Similarly, the abstract class distribution $C_{\bar{p}}$, is defined as a distribution

$$C_{\bar{p}} = (\Pr(C=c_1|\bar{p}),\dots,\Pr(C=c_n|\bar{p})).$$

Then, $C_{\bar{p}}$ can be also represented as a linear combination of C_p, using Bayes' Theorem again as in Proposition 3.2.

$$C_{\bar{p}} = \sum_{p \in inst_\varphi(\bar{p})} \lambda_p C_p, \text{ where } 0 \leqslant \lambda_p = \frac{\Pr(p)}{\Pr(\bar{p})} \leqslant 1, \text{ and } \sum_{p \in inst_\varphi(\bar{p})} \lambda_p = 1.$$

The following inequality is a direct consequence of (3.5) and the above representation.

$$H(C_{\bar{p}}) \geqslant \sum_{p \in inst_\varphi(\bar{p})} \lambda_p H(C_p). \tag{4.1}$$

The entropy of paths thus increases after data abstractions in general. The inequality (4.1) holds, even when we choose a good abstraction $\varphi=(\varphi_{A_1},\dots,\varphi_{A_m})$ with a small information loss for each data abstraction φ_{A_j}. For example, suppose that we have two data abstractions $\varphi_A=\{\dots,g_A=\{a_1,a_2,\dots,a_{20}\},\dots\}$ and $\varphi_B=\{\dots,g_B=\{b_1,b_2,b_3,\dots,b_{100}\},\dots\}$ such that each group consists of attribute values whose distributions are δ-similar with a small δ. So, for any i,j, $C_{A=a_i}$ is δ-similar to $C_{A=a_j}$, and $C_{B=b_i}$ is also δ-similar to $C_{B=b_j}$. In this case, the abstract path, $\bar{p}=(\bar{A}=g_A,\bar{B}=g_B)$, corresponds to 2000 instance paths $p_{ij}=(A=a_i,B=b_j)$ with their distributions $C_{p_{ij}}$ for $1\leqslant i\leqslant 20$ and $1\leqslant j\leqslant 100$. However, the class distributions $C_{p_{ij}}$ given the paths $A=a_i$, $B=b_j$ may not be similar in general. As a result, their mean, $C_{\bar{p}}$, is flattened, and $H(C_{\bar{p}})$ becomes higher. On the other hand, suppose that the distributions $C_{p_{ij}}$ are δ'-similar for 1900 combinations of i,j and a small error δ' as well. Then $C_{\bar{p}}$ is the linear combination of such similar distributions $C_{p_{ij}}$ with the weight $\frac{1900}{2000}$ and the remaining distributions with the weight $\frac{100}{2000}$, provided each instance path has the equal probability. $C_{\bar{p}}$ is closer to the mean of 1900 similar distributions and the remaining ones can be ignored as exceptional ones. Thus, the similarity of distributions is almost preserved in the abstraction, and $H(C_{\bar{p}})$ is closer to the mean of those at concrete level. This observation leads us to the following definition of clusters, highly weighted families of similar distributions. The notion is illustrated in Fig. 6.

Definition 4.2. ((τ,δ) *cluster of abstract path* $\bar{p}$) Given an abstract path $\bar{p}=(\overline{A_{i_1}}=g_{i_1},\dots,\overline{A_{i_k}}=g_{i_k})$ and $\tau,\delta>0$, a (τ,δ) cluster of $\bar{p}$ is a family of instance paths $\mathscr{M}\subseteq inst_\varphi$

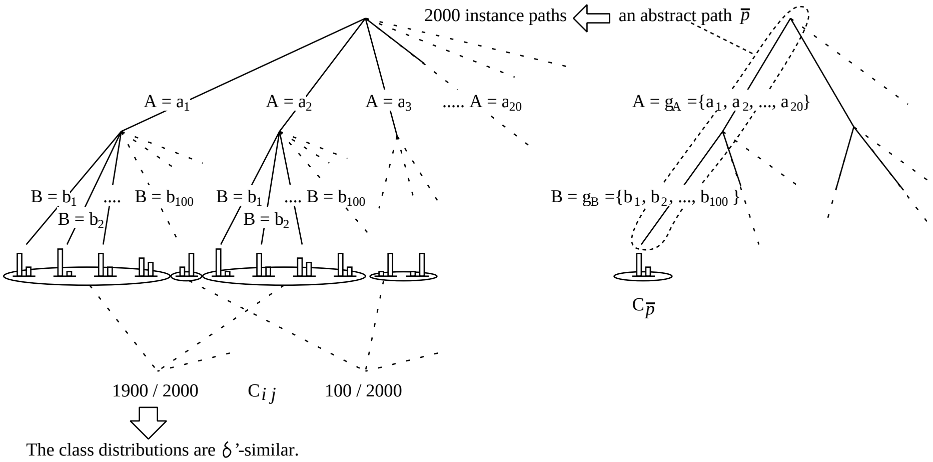


Fig. 6. A cluster of similar distributions.

$(\bar{p})$ such that

$$diam(\{C_p \mid p \in \mathcal{M}\}) < \delta \quad \text{and} \quad \lambda_{\mathcal{M}} = \frac{\sum_{p\in\mathcal{M}} \Pr(p)}{\Pr(\bar{p})} > 1 - \tau.$$

We call $\lambda_{\mathcal{M}}$ the cluster probability. In addition, a cluster of distributions $\mathscr{C}D(M)$ is defined as $\{C_p \mid p \in \mathcal{M}\}$.

The representation of $C_{\bar{p}}$ in terms of its instances C_p is furthermore transformed to a combination of distributions in $\mathscr{C}D(M)$ and the remaining ones in $\{C_p \mid p \in \mathcal{M}^c\}$, where $\mathcal{M}^c = inst_\varphi(\bar{p}) - \mathcal{M}$.

$$\begin{aligned} C_{\bar{p}} &= \sum_{p\in inst_\varphi(\bar{p})} \lambda_p C_p = \lambda_{\mathcal{M}} \sum_{p\in\mathcal{M}} \frac{\lambda_p}{\lambda_{\mathcal{M}}} C_p + (1-\lambda_{\mathcal{M}}) \sum_{p\in\mathcal{M}^c} \frac{\lambda_p}{1-\lambda_{\mathcal{M}}} C_p \\ &= \lambda_{\mathcal{M}} \sum_{p\in\mathcal{M}} \frac{\Pr(p)}{\sum_{p\in\mathcal{M}} \Pr(p)} C_p + (1-\lambda_{\mathcal{M}}) \sum_{p\in\mathcal{M}^c} \frac{\Pr(p)}{\sum_{p\in\mathcal{M}^c} \Pr(p)} C_p \\ &= \lambda_{\mathcal{M}} \sum_{p\in\mathcal{M}} w_{\mathcal{M}}(p) C_p + (1-\lambda_{\mathcal{M}}) \sum_{p\in\mathcal{M}^c} w_{\mathcal{M}^c}(p) C_p, \end{aligned}$$

where $w_{\mathscr{X}}(p) = \Pr(p)/\sum_{p\in\mathscr{X}} \Pr(p)$, the conditional probability of path p given a set of paths $\mathscr{X}$.

$\sum_{p\in\mathcal{M}} w_{\mathcal{M}}(p) C_p$ is just the mean of C_p in the cluster $\mathscr{C}D(\mathcal{M})$. Then, there exists a continuous function g such that

$$0 \leqslant H\left(\sum_{p\in\mathcal{M}} w_{\mathcal{M}}(p) C_p\right) - \sum_{p\in\mathcal{M}} w_{\mathcal{M}}(p) H(C_p)$$

$$= \sum_{c\in dom(C)} L\left(\sum_{p\in\mathscr{M}} w_{\mathscr{M}}(p)\Pr(c|p)\right) - \sum_{p\in\mathscr{M}} w_{\mathscr{M}}(p) \sum_{c\in dom(C)} L(c|p). \tag{4.2}$$

$$\leqslant g(\delta), \quad \text{where } g(\delta)\to 0 \text{ as } \delta\to 0. \tag{4.3}$$

In fact, the formula (4.2) and (3.4) are just the same by corresponding a, λ_a and g in (3.4) to $p, w_{\mathscr{M}}(p)$ and $\mathscr{M}$ in (4.2), respectively. Hence, we can similarly construct the function g, as in Proposition 3.4.

Moreover, for a variable x with its range $[0,1]$,

$$\lim_{x\to 1-0} H\left(x\sum_{p\in\mathscr{M}} w_{\mathscr{M}}(p)C_p + (1-x)\sum_{p\in\mathscr{M}^c} w_{\mathscr{M}^c}(p)C_p\right)$$
$$= H\left(\sum_{p\in\mathscr{M}} w_{\mathscr{M}}(p)C_p\right).$$

So, for any $\varepsilon_1 > 0$, there exists $\tau(\varepsilon_1) > 0$ such that

$$\left|H\left(x\sum_{p\in\mathscr{M}} w_{\mathscr{M}}(p)C_p + (1-x)\sum_{p\in\mathscr{M}^c} w_{\mathscr{M}^c}(p)C_p\right) - H\left(\sum_{p\in\mathscr{M}} w_{\mathscr{M}}(p)C_p\right)\right|$$
$$< \varepsilon_1$$

whenever $x > 1-\tau(\varepsilon_1)$. Thus,

$$\text{if } \lambda_{\mathscr{M}} > 1-\tau(\varepsilon_1) \text{ then } \left|H(C_{\bar{p}}) - H\left(\sum_{p\in\mathscr{M}} w_{\mathscr{M}}(p)C_p\right)\right| < \varepsilon_1. \tag{4.4}$$

Thus, from the inequalities (4.3) and (4.4), an evaluation of the entropy of abstract distribution in terms of entropy of its instance distributions is obtained.

Proposition 4.1. *Suppose that we have a $(\tau(\varepsilon_1),\delta)$ cluster $\mathscr{M}$ of an abstract path $\bar{p}$ satisfying the condition of* (4.4). *Then $H(C_{\bar{p}}) < \varepsilon_1 + g(\delta) + \sum_{p\in\mathscr{M}} w_{\mathscr{M}}(p)H(C_p)$ holds.*

If the cluster probability $\lambda_{\mathscr{M}}$ is higher, then we can take more tight ε_1. Moreover, if the diameter δ of instance distributions $\mathscr{C}D(M)$ is smaller, $g(\delta)$ is also a small number because of (4.3). Therefore, in such a case, $H(C_{\bar{p}})$ is closer to the mean $\sum_{p\in\mathscr{M}} w_{\mathscr{M}}(p)H(C_p)$, independent of the entropy of remaining distributions in $\{C_p \mid p\in\mathscr{M}^c\}$.

In addition, if every distribution in $\mathscr{C}D(M)$ has a low entropy, say $H(C_p) < \varepsilon$, then $\sum_{p\in\mathscr{M}} w_{\mathscr{M}}(p)H(C_p) < \varepsilon$. Therefore, $H(C_{\bar{p}}) < \varepsilon_1 + g(\delta) + \varepsilon$. So, $H(C_{\bar{p}})$ is expected to be closer to ε, as both δ and $1-\lambda_{\mathscr{M}}$ are smaller.

From these observations, we can claim a sufficient condition for our data abstraction to work well.

There exist many abstract paths for which there exist their clusters such that
(1) the cluster probability $\lambda_{\mathscr{M}}$ is sufficiently high,
(2) the diameter $diam(\mathscr{C}D(M))$ of instance distributions is sufficiently small, and
(3) the entropy of each instance distribution $C_p \in \mathscr{C}D(M)$ is low.
As the precision is expected to be achieved at such an abstract path, we need not expand it to obtain a longer path with its higher precision. On the other hand, there may exists an instance path in $inst_{\varphi}(\bar{p}) - \mathscr{M}$ whose probability is less than $1 - \lambda_{\mathscr{M}}$ and whose entropy is not sufficiently low. The abstract path ignores such a minor and exceptional instance, while we must expand such an instance path at concrete level to have a longer path within the required precision. Consequently, just one abstract path $\bar{p}$ corresponds to the paths in $\mathscr{M}$ plus such extended paths whose prefixes are in $inst_{\varphi}(\bar{p}) - \mathscr{M}$. This is a general reason why our data abstraction contributes to the reduction of the number of paths (nodes) in a decision tree without loosing its precision. In the next section, we see how many numbers of clusters are actually formed according to our selection criterion.

5. Information theoretical abstraction

We present here an algorithm based on our ITA method.

5.1. Two parameters: a split information and a change ratio

In Section 3, we proposed to adopt the mutual information as a measure to decide whether the class distributions before abstraction are almost the same (that is, similar) or not. More concretely speaking, we consider that a data abstraction φ_A with higher $I(C;\bar{A})$ is more preferable. However, if two or more preferable data abstractions for an attribute A (e.g. $\varphi_A, \psi_A, \ldots$) exist, we cannot decide an appropriate data abstraction from them according to the measure which is the mutual information. So, in the algorithm, we use the entropy $H(\bar{A})$ of the attribute $\bar{A}$, called a *split information*, to compare two or more preferable data abstractions and adopt an *information gain ratio* [20] $I(C;\bar{A})/H(\bar{A})$ as a selection measure of an appropriate data abstraction. The split information is given by Eq. (5.14).

$$H(A) = \sum_{a \in dom(A)} L(\Pr(A = a)), \quad \text{where } L(x) = \begin{cases} -x \log_2 x & \text{if } 0 < x \leqslant 1 \\ 0 & \text{if } x = 0 \end{cases}$$

For example, let us consider two preferable data abstractions φ_A and ψ_A for an attribute A. In this example, an attribute A after applying φ_A is denoted by $\bar{A}_{\varphi_A}$ and an attribute A after applying ψ_A is denoted by $\bar{A}_{\psi_A}$. According to the entropy theory, $H(A) \geqslant H(\bar{A}_{\varphi_A}) \geqslant H(\bar{A}_{\psi_A})$ holds, provided that φ_A is a refinement of ψ_A (that is, $\varphi_A \prec \psi_A$.). Hence, dividing $I(C;\bar{A}_{\varphi_A})$ by $H(\bar{A}_{\varphi_A})$, an information gain ratio $I(C;\bar{A}_{\varphi_A})/H(\bar{A}_{\varphi_A})$ tends to favor a data abstraction φ_A that identifies more numbers of attribute values with an abstract value. Under such a data abstraction φ_A, we will obtain a simpler generalized database than others. This will help the decision tree to perform their classification tasks. This is the reason why we adopt the information gain ratio, just as in the case of C4.5 [20].

The algorithm further introduces more a new value, *change ratio*, that is defined as the ratio of the information gain ratio after abstraction to one before abstraction. Intuitively speaking, a lower value of the change ratio implies that the class distributions before abstraction cannot be preserved well by the data abstraction. Therefore, if the value of change ratio for the data abstraction is lower than a threshold given by a user, our system rejects the data abstraction.

5.2. Algorithm

Let $R=(A_1,\ldots,A_m)$ be a relation. For each attribute A_i, assume that we have a class of possible data abstractions for A_i, $PosAbs(A_i)$. The task of our ITA algorithm is to select an appropriate data abstraction φ_{A_i} from $PosAbs(A_i)$ and to generalize R according to the selected data abstractions $\varphi=\{\varphi_{A_i} \mid 1\leqslant i\leqslant m\}$. Our algorithm is summarized in Fig. 7.

6. Experiments on a census database

We have made some experiments using our ITA system implemented in Visual C++ on PC/AT. This section shows the experimental results and discusses its usefulness.

In our experiments, we try discovering meaningful knowledge from a *Census Database in US Census Bureau* found in UCI repository [18]. The database consists of 32561 tupples each of which has values for 15 attributes including *age*, *marital_status*, *hours_per_week*, *salary*, etc. Apart from this database (it is referred to as *training data*), a small database consisting of 15 060 tupples is prepared in order to check usefulness of discovered knowledge (it is referred to as *test data*). A class of possible abstractions for each attribute is constructed based on a machine-readable dictionary *WordNet* [16] and is given to our system. This reason is that if we directly extract possible groups from large numbers of attribute values of an attribute, the extraction causes a combinatorial explosion of attribute values. WordNet has numerous concept hierarchies in various contexts. The concept hierarchies are classified into several semantic categories. It should be noted that there exist many multiple inheritances in the hierarchy.

The way to construct hierarchies (i.e. data abstractions) used in ITA system is illustrated in Fig. 8. ITA extracts many primitive views (i.e. groups) firstly from WordNet. A primitive view is defined as a mapping $pv: A\rightarrow\{a'\}$ which abstracts many attribute values $a_1,\ldots,a_\ell$ of an attribute A into one abstract concept a'. That is, the primitive view has one abstract concept and many concrete values (i.e. attribute values), and is extracted from WordNet on each semantic category. Secondly, ITA composes primitive views to generate possible data abstractions. A data abstraction is defined as composite views that are composed by primitive views. ITA finally receives these data abstractions.

```
INPUT($R=(A_1,\ldots,A_m)$): a relational database
        $C$: a target attribute (i.e. attribute $A_t$) in $R$,
        $PosAbs(A_i)$: a class of possible data abstractions for $A_i$ $(i \neq t)$,
        $\mathcal{V}_{lower}$: a lower bound of change ratio)
for $A_i$ in $R$ $(i \neq t)$ do
   begin
     compute $gain\_ratio(C, A_i)$.
     $\varphi \leftarrow \emptyset$
     $max\_gain\_ratio \leftarrow 0$
     for $\varphi^j_{A_i} \in PosAbs(A_i)$ do
         begin
           compute $gain\_ratio(C, \varphi^j_{A_i}(A_i))$,
               where $\varphi^j_{A_i}(A_i)$ is an attribute after applying $\varphi^j_{A_i}$.
           if $max\_gain\_ratio \leqslant gain\_ratio(C, \varphi^j_{A_i}(A_i))$ then
             begin
               $max\_gain\_ratio \leftarrow gain\_ratio(C, \varphi^j_{A_i}(A_i))$
               $\varphi_{A_i} \leftarrow \varphi^j_{A_i}$
               $\overline{A_i} \leftarrow \varphi^j_{A_i}(A_i)$
             end
         end
     compute $change\_ratio(C, \overline{A_i}) = gain\_ratio(C, \overline{A_i})/gain\_ratio(C, A_i)$
     if $\mathcal{V}_{lower} \leqslant change\_ratio_{\overline{A_i}}(C, A_i)$ then
       begin
         $\varphi \leftarrow \varphi \cup \{\varphi_{A_i}\}$.
       end
   end
transform $R$ into $\varphi(R)$ under $\varphi$.
add a special attribute $vote$ to a relational schema $(A_1,\ldots,A_m)$
and build a schema $R^* = (A_1,\ldots,A_m, vote)$, where $dom(vote) = \mathbf{N}$.
$R^* \leftarrow \{(a_1,\ldots,a_m,1) \mid (a_1,\ldots,a_m) \in \varphi(R)\}$.
for $t_i, t_j \in R^*$ such that they are identical except their vote values do
   begin
     $R^* \leftarrow R^* - \{t_i\}$
     $vote(t_j) \leftarrow vote(t_i) + vote(t_j)$
   end
Output($R^*$: a generalized database of $R$
        according to appropriate data abstractions)
```

Fig. 7. ITA algorithm.

6.1. Construction of a decision tree from the generalized database

Our ITA system can appropriately generalize the given original database according to a given class of possible data abstractions for each attribute. Since the descriptions

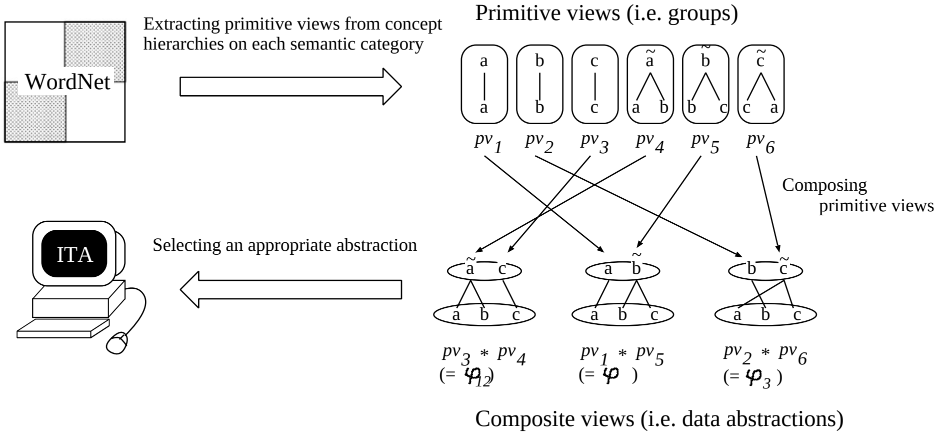

WordNet
Extracting primitive views from concept hierarchies on each semantic category
Primitive views (i.e. groups)
a
a
b
b
c
c
ã
a b
b̃
b c
c̃
c a
pv1 pv2 pv3 pv4 pv5 pv6
Composing primitive views
ITA
Selecting an appropriate abstraction
ã c
a b c
a b̃
a b c
b c̃
a b c
pv3 * pv4
(= φ12)
pv1 * pv5
(= φ)
pv2 * pv6
(= φ3)
Composite views (i.e. data abstractions)

Fig. 8. A composition of primitive views.

of such a generalized database are not too complicated, we can expect to obtain an abstract decision tree from the generalized database that is more compact than an original decision tree obtained from the original database. For the census database, we compare the abstract decision tree with the original decision tree in terms of the size and the error rate.

Basically according to the algorithm shown in Fig. 7, the generalized database is obtained by substituting attribute values with the corresponding abstract concept in the class of possible abstractions. The generalized database is given to C4.5 in order to obtain a decision tree that is referred to as the abstract decision tree by "ITA+C4.5". We construct various abstract decision trees by varying the threshold of the change ratio, and compare these abstract decision trees with the original one obtained from the original census database by C4.5. The test data is used to evaluate the accuracy of them. The experimental results are shown in Figs. 9–11.

Fig. 12 furthermore shows a part of the constructed abstract decision tree. Various data abstractions are applied to each value of node. For example, Bachelor and Master are abstracted into university. An abstract path

$$\bar{p} = \{\texttt{Marrital\_status} = \texttt{married\_civ\_spouse},\ \texttt{Education} = \texttt{university},$$

$$\texttt{Age} = \texttt{young},\ \texttt{Occupation} = \texttt{skilled\_worker}\}$$

corresponds to 120 $(= 2 \times 30 \times 2)$ concrete paths (i.e. instance paths) at concrete level. From this, we expect an application of data abstractions to reduce the size of a decision tree effectively when many instance paths of the abstract path correspond to paths in an original decision tree which is constructed from a database before generalization.

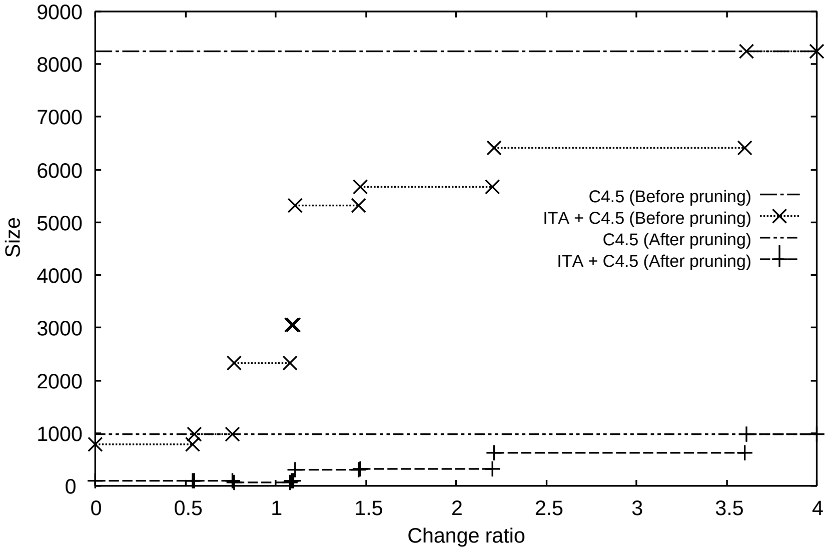


Fig. 9. The size of the decision tree.

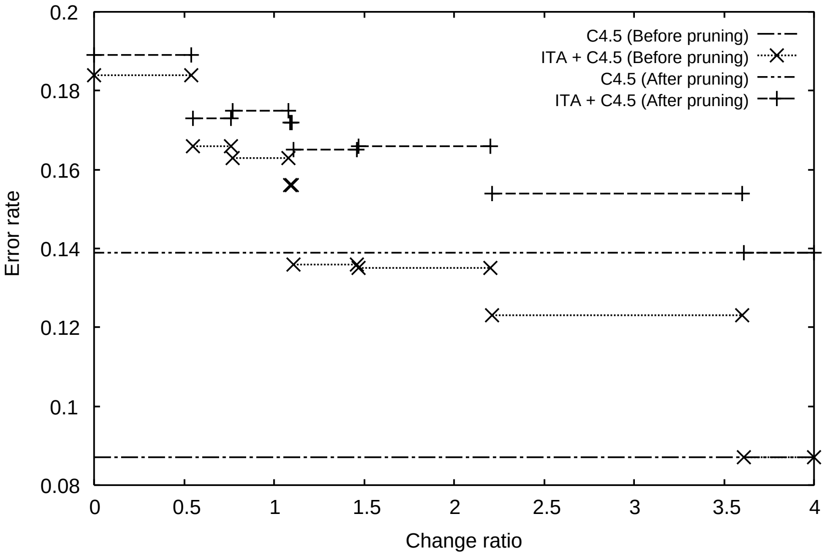


Fig. 10. The error rate for training data.

6.1.1. Decision tree before pruning

We examine here the decision trees before pruning. Fig. 9 shows that the size of the abstract decision tree constructed by ITA+C4.5 is smaller than the original one. Furthermore, the size of the original decision tree within the threshold 0.00 to 1.10 is

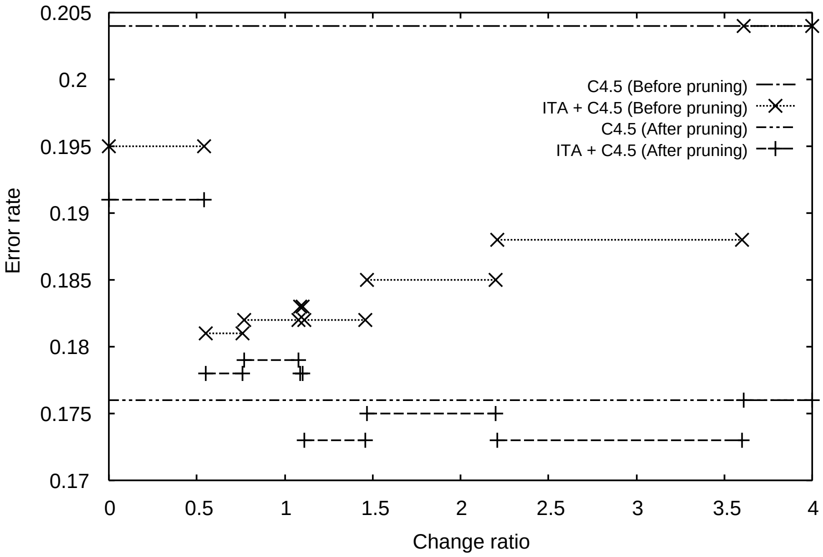


Fig. 11. The error rate for test data.

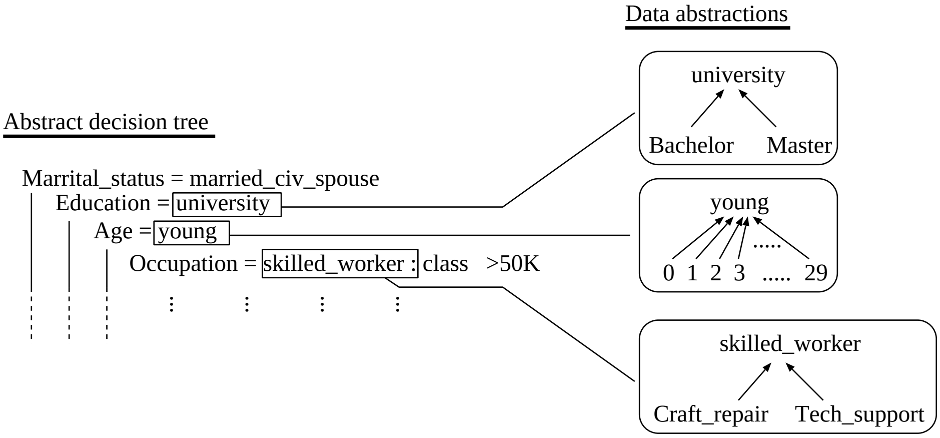


Fig. 12. A part of an abstract decision tree.

reduced dramatically as compared with reduction of the size of the original one after pruning. Fig. 10 shows that the original decision tree is better than the abstract one as to the error rate for training data. Conversely, concerning the error rate for test data, the abstract decision tree is better than the original one, as shown in Fig. 11. One reason for this is that the original decision tree may include many redundant nodes which do not contribute to classifying test data, because the original decision tree can classify training data in detail. Hence, we consider that many redundant nodes in the original decision tree are not important for classification task and are removed by the generalization.

6.1.2. Decision tree after pruning

In this section, we investigate the decision trees after pruning. In all range of the threshold, the size of each abstract decision tree constructed by ITA+C4.5 is smaller than the original one constructed by C4.5, as shown in Fig. 9. Especially, the size of the abstract decision tree within the threshold 0.00 to 1.10 is quite small. From this, redundant nodes which still remain in the abstract decision tree before pruning are eliminated by pruning. Fig. 10 shows that the decision tree constructed by C4.5 is better than the abstract one constructed by ITA+C4.5 in terms of the error rate for the original database (i.e. training data). However, we cannot find any remarkable difference between two error rates for the test data, as shown in Fig. 11. Furthermore, it is considered that the original decision tree would be too specific (over-fit) to the original database (i.e. training data). On the other hand, the error rate of the abstract decision tree is almost equal for both the training data and the test data. Hence, it is considered that ITA is very useful to decrease the size of decision tree and still preserve the classification accuracy.

6.2. Observations of a cluster of similar class distributions

Moreover, we observe a cluster of similar class distributions on each abstract path of the abstract decision tree before pruning. The number of the abstract paths that satisfy preservingness of similarity among the class distributions is shown in Fig. 13. The number of all abstract paths is 2158. Meanings of the terms used in Fig. 13 are as follows. If the number of instances (i.e. tupples) which are classified into one abstract path is not less than the threshold given by a user, we call such an abstract path a "*valid path*". The threshold is called "*The threshold of valid path*". An abstract path that forms a cluster of similar class distributions on instance paths of it is called a "*cluster path*". In our experiment, we observe cluster paths in a set of valid paths. More concretely, ITA system firstly receives two thresholds. One threshold δ is an acceptable error of the similarity among class distributions, and another threshold λ is a cluster probability. Secondly ITA system counts the number of cluster paths which meets the following condition using two thresholds. In cluster paths, an acceptable error of the similarity preservingness is δ or less and the cluster probability is λ or more. Finally, we vary two thresholds δ and λ and repeat the above processes. When the number of instances which are classified into one abstract path is not less than 2, 30 and 150 (i.e. a threshold of the valid path is not less than 2, 30 and 150), the experimental results are shown in Fig. 13(a), (b) and (c), respectively.

Furthermore, we calculates the average of the number of instance paths which form the cluster, when the threshold of the valid path is 2, 30 and 150 instances. Figs. 14–16 show the results of the calculations (i.e. the number of instance paths per one abstract path).

(a) The threshold of the valid path : 2 (instances)
The number of paths : 2158
The number of valid paths : 1245

The number of cluster paths (depth: 1 / depth: 2 / depth: 3)

δ λ	0.60	0.65	0.70	0.75	0.80	0.85
0.20	707 (586 / 120 / 1)	707 (586 / 120 / 1)	706 (585 / 120 / 1)	706 (585 / 120 / 1)	704 (584 / 119 / 1)	701 (581 / 119 / 1)
0.25	740 (612 / 127 / 1)	740 (612 / 127 / 1)	738 (610 / 127 / 1)	738 (610 / 127 / 1)	736 (609 / 126 / 1)	727 (632 / 123 / 1)
0.30	764 (632 / 131 / 1)	764 (632 / 131 / 1)	762 (630 / 131 / 1)	758 (628 / 129 / 1)	758 (628 / 129 / 1)	746 (620 / 125 / 1)
0.35	792 (653 / 138 / 1)	792 (653 / 138 / 1)	788 (649 / 138 / 1)	788 (649 / 138 / 1)	778 (641 / 136 / 1)	750 (624 / 125 / 1)
0.40	841 (684 / 156 / 1)	841 (684 / 156 / 1)	837 (680 / 156 / 1)	835 (679 / 155 / 1)	813 (663 / 149 / 1)	758 (629 / 128 / 1)

(b) The threshold of the valid path : 30 (instances)
The number of paths : 2158
The number of valid paths : 157

The number of cluster paths (depth: 1 / depth: 2)

δ λ	0.60	0.65	0.70	0.75	0.80	0.85
0.20	66 (54 / 12)	66 (54 / 12)	65 (54 / 11)	65 (54 / 11)	64 (54 / 10)	62 (52 / 10)
0.25	73 (59 / 14)	73 (59 / 14)	72 (59 / 13)	72 (59 / 13)	72 (59 / 13)	67 (55 / 12)
0.30	79 (61 / 18)	79 (61 / 18)	78 (61 / 17)	78 (61 / 17)	77 (61 / 16)	74 (59 / 15)
0.35	82 (63 / 19)	82 (63 / 19)	81 (63 / 18)	81 (63 / 18)	79 (62 / 17)	75 (60 / 15)
0.40	95 (65 / 30)	95 (65 / 30)	94 (65 / 29)	92 (65 / 27)	87 (63 / 24)	77 (60 / 17)

(c) The threshold of the valid path : 150 (instances)
The number of paths : 2158
The number of valid paths : 36

The number of cluster paths (depth: 1 / depth: 2)

δ λ	0.60	0.65	0.70	0.75	0.80	0.85
0.20	15 (14 / 1)	15 (14 / 1)	15 (14 / 1)	15 (14 / 1)	15 (14 / 1)	13 (12 / 1)
0.25	16 (14 / 2)	16 (14 / 2)	16 (14 / 2)	16 (14 / 2)	16 (14 / 2)	14 (13 / 1)
0.30	17 (14 / 3)	17 (14 / 3)	17 (14 / 3)	17 (14 / 3)	16 (14 / 2)	16 (14 / 2)
0.35	17 (14 / 3)	17 (14 / 3)	17 (14 / 3)	17 (14 / 3)	16 (14 / 2)	16 (14 / 2)
0.40	19 (14 / 5)	19 (14 / 5)	19 (14 / 5)	19 (14 / 5)	18 (14 / 4)	16 (14 / 2)

Fig. 13. The cluster of similar class distributions.

6.2.1. The number of cluster paths

When 2 or more instances are classified into one abstract path, the number of the valid path is 1245 and the number of the cluster paths in a set of the valid paths is shown in Fig. 13a. About 60 percent of the valid paths are cluster paths ($\delta = 0.2$–0.3 and $\lambda = 0.6-0.85$). As a matter of course, the number of the cluster paths tends to decrease when δ changes from 0.40 to 0.20 and λ changes from 0.60 to 0.85 (i.e. a condition which satisfies the cluster of similar class distributions is severe). Three values in parentheses are the number of cluster paths at each depth (i.e. 1, 2 and 3) of an abstract path from a root node to a leaf node. If the depth is 4 or more, cluster paths do not exist. From this observation, the cluster of similarity class distributions is formed on the short abstract path, because ITA performs only one application of data abstraction for the whole database. That is, ITA guarantees the similarity preservingness of the class distributions at the root node but does not generally guarantee the preservingness at other nodes.

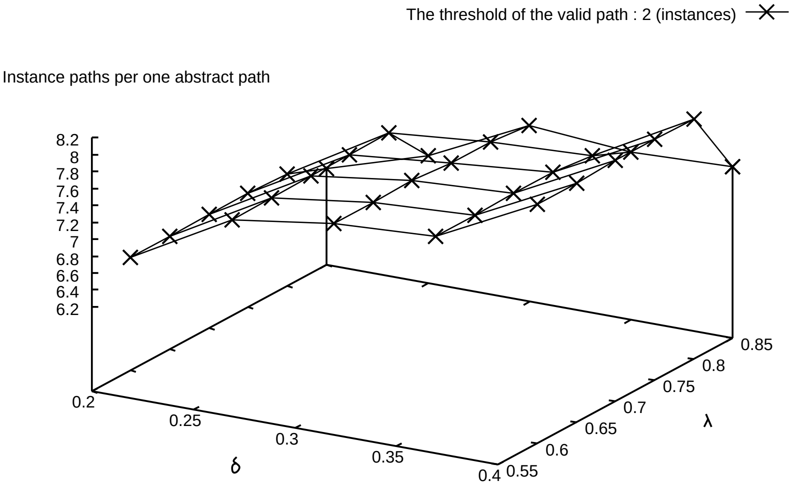


Fig. 14. The threshold of the valid path: 2 (instances).

In Fig. 13b, the number of the valid paths is 157 and it is 7.28 percent (=157/2158) of whole paths. The clusters are formed at a depth 1 and 2. In case of $\delta = 0.2$, the cluster paths are nearly 40 percent of the valid paths.

Fig. 13c shows that the number of valid paths is 36 when 150 or more instances are classified into one abstract path. It is 1.67 percent of all abstract paths. On condition that $\delta = 0.20$ and $\lambda = 0.85$ (i.e. most severe condition), the number of cluster paths is 13 and it accounts for about 36 percent of the valid paths, and class distributions in the cluster on each cluster path are extremely similar.

From above observations, the cluster paths account for 40–50 percent of the abstract paths in the abstract decision tree.

6.2.2. *The number of instance paths per one abstract path*

Fig. 14 shows that the average of instance paths per one abstract path is about 7. The average is less than the other results shown in Fig. 15 and Fig. 16. One reason is that there are many abstract paths corresponding only to 2 instances paths since the abstract path which classifies 2 or more instances is regard as the valid path.

When 30 or more instances are classified into one abstract path, the average of the number of the corresponding instance paths is about 40 (Fig. 15).

Furthermore, when more than 150 instances are classified into one abstract path, in Fig. 16, the average is 83.31 ($\delta = 0.2$ and $\lambda = 0.85$). From this result, many instance paths of the abstract path can form the cluster. That is, in these cases, the cluster path corresponds to many instance paths.

Consequently, we confirm that the abstract path consists of many corresponding paths when the abstract path which classifies many instances forms the cluster of similar class distribution. We also observe that the size of the decision tree decreases since such paths often exist in the original decision tree constructed by C4.5.

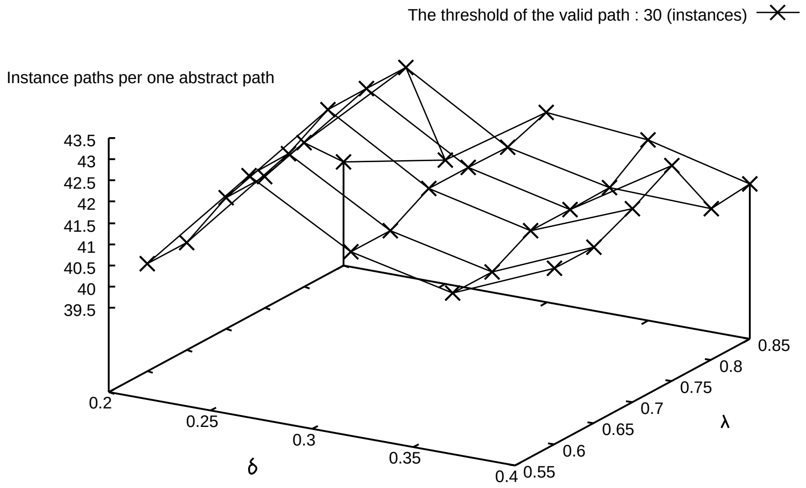


Fig. 15. The threshold of the valid path: 30 (instances).

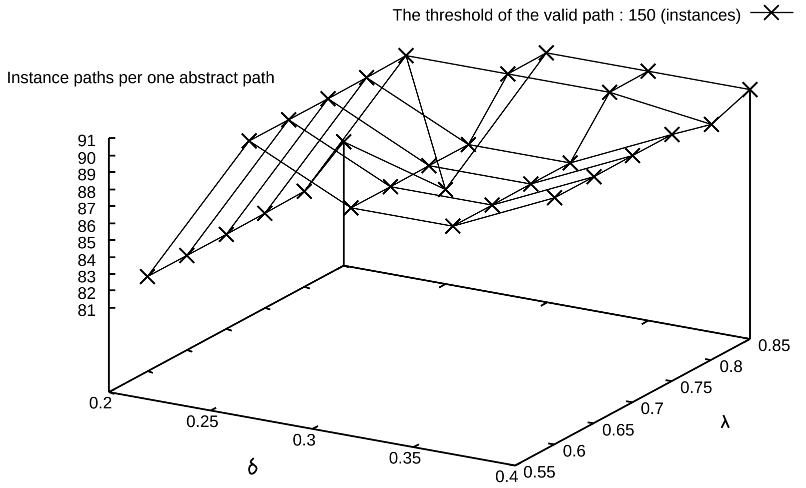


Fig. 16. The threshold of the valid path: 150 (instances).

7. Concluding remarks

In this paper, we proposed a generalization method, called Information Theoretical Abstraction, in which an appropriate abstraction among possible ones is selected according to the information theoretical measure.

Our analysis presents some effective cases in which our ITA can work well from a theoretical point of view. Furthermore, such effective cases have actually been observed through our experimentation. More concretely, we have observed an effective case in which the similarity preservingness holds for a cluster of distributions of a very low entropy with a very high probability under a *short* conditioning. However, it would be expected that satisfying similarity preservingness would increasingly be harder when our decision tree must have longer paths in order to accurately classify data. The current ITA, unfortunately, would not be useful in such a case because ITA in this paper has to *simultaneously* select an appropriate abstraction for each attribute before constructing decision tree. This means that the current ITA selects abstractions without taking account of any conditioning by selected attributes during constructing a decision tree. In order to cope with the problem, we can propose to select appropriate abstractions at each expansion step of nodes. That is, ITA *iteratively* finds appropriate abstractions satisfying the similarity preservingness at each expansion step, taking conditioning so far into account. We have roughly formulated such a method and made a preliminary experimentation [12]. Although the current formulation has not been an exact realization of the method yet, our preliminary experimental results show that iteratively finding abstractions seems to be useful even in case for decision tree with longer paths. We are currently further studying the new abstraction method from theoretical and experimental points of view.

References

[1] P. Adriaans, D. Zantinge, Data Mining, Addison-Wesley, Reading, MA, Longman, New York, 1996.

[2] S. Arimoto, Probability, Information, Entropy, Morikita Shuppan, 1980 (in Japanese).

[3] K. Cherkauer, J. Shavlik, Growing simpler decision trees to facilitate knowledge discovery, Proc. 2nd Internat. Conf. on Knowledge Discovery and Data Mining, 1996, pp. 315–318.

[4] U.N. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery: an overview, in: U.N. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, Cambridge, MA, 1996, pp. 1–33.

[5] U.N. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, Cambridge, MA, 1996.

[6] J. Han, Y. Cai, N. Cercone, Knowledge discovery in databases: an attribute-oriented approach, Proc. of VLDB'92, Canada, 1992, pp. 547–559.

[7] J. Han, Y. Fu, Attribute-oriented induction in data mining, in: U.N. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, Cambridge, MA, 1996, pp. 399–421.

[8] M. Holsheimer, M. Kersten, Architectural support for data mining, CWI Technical Report CS-R9429, Amsterdam, The Netherlands, 1994.

[9] Y. Kudoh, M. Haraguchi, An appropriate abstraction for an attribute-oriented induction, Proc. 2nd Internat. Conf. on Discovery Science, Lecture Notes in Artificial Intelligence, Vol. 1721, 1999, pp. 43–55.

[10] Y. Kudoh, M. Haraguchi, Data mining by generalizing database based on an appropriate abstraction, J. Japanese Soc. Artif. Intell. 15 (4) (2000) 638–648 (in Japanese).

[11] Y. Kudoh, M. Haraguchi, An appropriate abstraction for constructing a compact decision tree, Proc. 3rd Internat. Conf. on Discovery Science, Lecture Notes in Artificial Intelligence, 2000, pp. 295–298.

[12] Y. Kudoh, M. Haraguchi, Detecting a compact decision tree based on an appropriate abstraction, Proc. 2nd Internat. Conf. on Intelligent Data Engineering and Automated Learning, Lecture Notes in Computer Science, Springer, Berlin, 2000, pp. 60–70.

[13] K. Matsumoto, C. Morita, H. Tsukimoto, Generalized rule discovery in databases by finding similarities, SIG-J-9401-15, Japanese Society for Artificial Intelligence, 1994, pp. 111–118 (in Japanese).

[14] R.S. Michalski, I. Bratko, M. Kubat (Eds.), Machine Learning and Data Mining: Methods and Applications, Wiley, London, 1997.

[15] R.S. Michalski, K.A. Kaufman, Data mining and knowledge discovery: a review of issues and a multistrategy approach, in: R.S. Michalski, I. Bratko, M. Kubat (Eds.), Machine Learning and Data Mining: Methods and Applications, Wiley, London, 1997, pp. 71–112.

[16] G.A. Miller, Nouns in WordNet: a lexical inheritance system, Internat. J. Lexicography 3(4) (1990) 245–264. ftp://ftp.cogsci.princeton.edu/pub/wordnet/5papers.ps.

[17] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K. Miller, Introduction to WordNet: an on-line lexical database, Internat. J. Lexicography 3 (4) (1990) 235–244.

[18] P.M. Murphy, D.W. Aha, UCI Repository of machine learning databases, http://www.ics.uci.edu/mlearn/MLRepository.html.

[19] D.A. Plaisted, Theorem proving with abstraction, Artif. Intell. 16 (1981) 47–108.

[20] J.R. Quinlan, C4.5—Programs for Machine Learning, Morgan Kaufmann, Los Altos, CA, 1993.

[21] J.D. Tenenberg, Abstracting first-order theories, in: D.P. Benjamin (Ed.), Change of Representation and Inductive Bias, Kluwer Academic Publisher, Dordrecht, 1989, pp. 67–79.

[22] J.D. Tenenberg, Abstraction in planning, in: J.F. Allen, et al., (Eds.), Reasoning about Plan, Morgan Kaufmann, San Mateo, CA, 1991, pp. 213–283.