WILEY InterScience® DISCOVER SOMETHING GREAT

The Generalized Dependency Degree Between Attributes

Haixuan Yang, Irwin King and Michael R. Lyu
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong.
Email: {hxyang, king, lyu}@cse.cuhk.edu.hk

Inspired by the dependency degree γ, a traditional measure in Rough Set Theory, we propose a generalized dependency degree, Γ, between two given sets of attributes, which counts both deterministic and indeterministic rules while γ counts only deterministic rules. We first give its definition in terms of equivalence relations and then interpret it in terms of minimal rules, and further describe the algorithm for its computation. To understand Γ better, we investigate its various properties. We further extend Γ to incomplete information systems. To show its advantage, we make a comparative study with the conditional entropy and γ in a number of experiments. Experimental results show that the speed of the new C4.5 using Γ is greatly improved when compared with the original C4.5R8 using conditional entropy, while the prediction accuracy and tree size of the new C4.5 are comparable with the original one. Moreover, Γ achieves better results on attribute selection than γ. The study shows that the generalized dependency degree is an informative measure in decision trees and in attribute selection.

Introduction

As one of several models that are used to extract previously unknown and potentially useful information from the databases, Rough Set Theory provides an effective tool for mining deterministic rules from a database. In recent years, it has received considerable attention. For example, Yao, Li, Lin, and Liu (1994) makes it possible to obtain the upper and lower bounds by eliminating transitivity, reflexivity, and symmetry axioms. In other works (Kryszkiewicz, 1998, 1999; Leung & Li, 2003; Lingras & Yao, 1998), various extensions of the Rough Set Theory to incomplete information systems are considered.

The objective of this article is to generalize the dependency degree γ that is widely used in the rough set theory. Besides, we aim to develop a deeper understanding of the generalized dependency degree and to justify it both theoretically and empirically. On the theoretical side, we give its various forms and describe its properties; on the empirical side, we show its effectiveness in decision trees and in attribute selection. For this sake, we first need to introduce the following basic concepts concerning the dependency degree.

An information system is represented by an attribute-value table in which rows are labeled by objects of the universe and columns by their attributes. Denote the universe of objects by U, the set of attributes or features by A, and the set of all possible values of attribute a by V_a. Let P be a subset of A, that is, P is a subset of attributes. The P-indiscernibility relation, denoted by $IND(P)$, is defined as

$$IND(P) = \{(x, y) \in U \times U | (\forall a \in P) a(x) = a(y)\},$$

is an equivalence relation. The set of equivalence classes is denoted by $U/IND(P)$ or by U/P and the equivalence class in U/P is called the P-class. For $x \in X$, let P(x) denote the P-class containing x.

Let C and D be two subsets of A. The dependency degree $\gamma(C, D)$ is defined (Pawlak, 1999) as

$$\gamma(C, D) = 1/|U| \sum_{X \in U/D} |\underline{C}(X)|,$$

where, $\underline{C}(X) = \cup \{Y \in U/IND(C) | Y \subseteq X\}$, $|U|$ and $|\underline{C}(X)|$ denote the cardinality of the set U and the cardinality of the set $\underline{C}(X)$, respectively. $|.|$ denotes the cardinality of a set without further notice throughout the article. $\gamma(C, D)$ expresses the percentage of objects that can be correctly classified into the D-class by employing attribute C. It is also the relative number of elements of U that can be described by deterministic rules because each C-class contained in a D-class corresponds to a deterministic rule, and vice versa. γ is a traditional measure in Rough Set Theory (Gediga & Düntsch, 2001) and is employed to find the reduction of attributes.

Despite its effectiveness in discovering attribute reduction, there are two problems concerning. One problem is that the rough set methods developed so far are not always sufficient for extracting rules from decision tables, and so the set of all decision rules generated from all conditional attributes can contain many chaotic rules inappropriate for unseen object classification (Gunther & Ivo, 2000; Ivo & Gunther, 1997).

Received February 12, 2007; revised March 5, 2007; accepted March 5, 2007

© 2007 Wiley Periodicals, Inc. • Published online 27 September 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.20697

To ensure that the prediction is not based on a few observations, Hassanien (2004) introduces a significance testing to evaluate the statistical significance of the rules based on the permutation distribution of γ, and discard those rules that can not pass the significance test. In a case study, this way achieves higher accuracy rates and less number of rules than the traditional method.

The other problem is that $\gamma(C, D)$ loses some kind of dependency, and so it does not accurately express the dependency. In this article, we focus on this problem. Because $\gamma(C, D)$ only counts deterministic rules (explained in the Connections Between $\Gamma(C, D)$ and Minimal Rules section, where all the needed concepts will have been introduced), in the extreme cases when there is no deterministic rule, the dependency degree $\gamma(C, D)$ will be equal to zero; however, there may actually be some kind of indeterministic dependency between C and D. For example, in Table 1 where a, b, c, and d represent *headache, muscle pain, body temperature,* and *influenza,* respectively. Let $C = \{a\}$, $D = \{d\}$. There are two D-classes: $\{e1, e4, e5\}$ and $\{e2, e3, e6, e7\}$, and two C-classes: $\{e1, e2, e3, e7\}$ and $\{e4, e5, e6\}$. Because no C-class is completely contained in a D-class, by the definition of $\gamma(C, D)$, we have $\gamma(C, D) = 0$. This contradicts the intuition that D depends on C in some way. We can analyze this contradiction by an equivalent definition of $\gamma(C, D)$ shown in the Connections Between $\Gamma(C, D)$ and Minimal Rules section. The phenomenon that $\gamma(C, D) = 0$ can be explained by the fact that none of these rules

$$a = Y \rightarrow d = Y,$$

$$a = Y \rightarrow d = N,$$

$$a = N \rightarrow d = N,$$

$$a = N \rightarrow d = N$$

is deterministic, and therefore all these rules are counted as zero by $\gamma(C, D)$. Although the rule $a = Y \rightarrow d = Y$, is indeterministic, it actually contains the dependency relation (indeterministic) between the attribute a and the attribute b, and so it should be counted. To avoid the loss of the indeterministic dependency, we propose the generalized dependency degree $\Gamma(C, D)$ by counting both deterministic rules and indeterministic rules.

In many cases, the databases used for data mining contain missing values of attributes. The problem of rule generation from incomplete information systems is considered in the literature. The simplest method is to remove examples with unknown values. Replacing every missing value with the set of all possible values is another method (Lingras & Yao, 1998). Introducing the similarity relation and completion of an incomplete information system is a more accurate way to handle missing values (Kryszkiewicz, 1998, 1999; Leung & Li, 2003). To make the generalized dependency degree applicable to incomplete information systems, we extend its definition to the case of incomplete information systems by replacing missing values with their probabilistic distributions at first and then extending the definition of the confidence and the strength of a rule to incomplete information systems.

TABLE 1. Influenza data.

	Headache (a)	Muscle pain (b)	Body temperature (c)	Influenza (d)
e1	*Y*	*Y*	*0*	*N*
e2	*Y*	*Y*	*1*	*Y*
e3	*Y*	*Y*	*2*	*Y*
e4	*N*	*Y*	*0*	*N*
e5	*N*	*N*	*1*	*N*
e6	*N*	*Y*	*2*	*Y*
e7	*Y*	*N*	*1*	*Y*

The generalized dependency degree $\Gamma(C, D)$ is different from the γ-like statistics introduced by Gediga and Düntsch (2001), the idea of which is to count the number of errors, whereas $\Gamma(C, D)$ counts every object by a corresponding fraction (as we will explain later, this is equivalent to counting all the minimal rules whose confidences are not equal to zero).

The rest of this article is organized as follows. In the Definition of the Generalized Dependency Degree section, we give the first two forms of the generalized dependency degree. In the Properties of the Generalized Dependency Degree section, we discuss the properties of this measure and give its third form. In the Extension of Γ to Incomplete Information Systems section, we extend it to incomplete information systems. In the Discussion: Comparison with the Conditional Entrophy section, we compare it with conditional entropy. In the Experiments section, we conduct experiments to support the generalized dependency degree concept. In the Conclusion and Fututre Work section, we draw a conclusion about the generalized dependency degree.

Definition of the Generalized Dependency Degree

In this section, we first cite the formal language in a complete information system, which is used to describe the decision rules. Then we give the definition of the generalized dependency degree. Finally we connect the minimal decision rules and the generalized dependency degree.

A Formal Language to Describe the Decision Rule

The decision language is defined in (Pawlak, 2002a, 2002b). Let $S = (U, A, V, f)$ be an information system. With every $B \subseteq A$, we associate a formal language, i.e., a set of formulae *For* (B). Formulae of *For* (B) are built up from attribute-value pairs $a = v$ where $a \in B$ and $v \in V_a$ by means of logical connectives $\wedge$ (and), $\vee$ (or), $\sim$ (not) in the standard way. For any $\Phi \in$ *For* (B), we denote the set of all objects satisfying Φ by *supp* (Φ); this is called the support of Φ.

A decision rule in S is an expression $\Phi \rightarrow \Psi$, where $\Phi \in$ *For* (C), $\Psi \in$ *For* (D), C, D are condition and decision attributes, respectively, and Φ and Ψ are referred to as the condition and decision of the rule, respectively. A decision rule $\Phi \rightarrow \Psi$ is called a *deterministic rule* in S if *supp*(Φ) $\subseteq$

supp(Ψ), and an *indeterministic rule* otherwise. With every decision rule $\Phi \rightarrow \Psi$, we associate a conditional probability called the *confidence* (the *certainty factor* of the rule $\Phi \rightarrow \Psi$), and denote it by $Con(\Phi \rightarrow \Psi)$, which can be written as

$$Con(\Phi \rightarrow \Psi) = \frac{|supp\ (\Phi \wedge \Psi)|}{|supp(\Phi)|}.$$

We denote the *strength* of decision rule $\Phi \rightarrow \Psi$ by $Str(\Phi \rightarrow \Psi)$, which is defined as:

$$Str(\Phi \rightarrow \Psi) = \frac{|supp\ (\Phi \wedge \Psi)|}{|U|}.$$

The denominator $|supp(\Phi)|$ in $Con(\Phi \rightarrow \Psi)$ counts only the objects that satisfy the formula Φ, while the denominator $|U|$ in $Str(\Phi \rightarrow \Psi)$ counts all the objects. We show the definitions of the confidence and strength of a rule by the following example.

Example 1: In Table 1, $A = \{a, b, c, d\}$, $U = \{e1, e2, e3, e4, e5, e6, e7,\}$. Let $C = \{a, b\}$, $D = \{d\}$. We consider the rule $a = Y \wedge b = Y \rightarrow d = Y$. The condition part Φ is equal to $a = Y \wedge b = Y$, while decision part Ψ is equal to $d = Y$, and so $\Phi \wedge \Psi$ is equal to $a = Y \wedge b = Y \wedge d = Y$. Because $supp(\Phi) = \{e1, e2, e3\}$ and $supp(\Phi \wedge \Psi) = \{e2, e3\}$, we have $|supp(\Phi)| = 3$ and $|supp(\Phi \wedge \Psi)| = 2$, and so $Con(\Phi \rightarrow \Psi) = 2/3$, and $Str(\Phi \rightarrow \Psi) = 2/7$.

The Generalized Dependency Degree

We give our first form of the generalized dependency degree in terms of equivalence relations as follows.

Definition 1: The generalized dependency degree $\Gamma(C, D)$ is defined as

$$\Gamma(C, D) = \frac{1}{|U|}\sum_{x \in U}\frac{|D(x) \cap C(x)|}{|C(x)|}, \qquad (1)$$

where $D(x)$ and $C(x)$ denote the *D*-class containing *x* and *C*-class containing *x* respectively (recall that, in the Introduction section, we defined *P*-class for any attribute set *P*).

Note that the dependency degree $\gamma(C, D)$ can be rewritten as

$$\gamma(C, D) = \frac{1}{|U|}\sum_{x \in U \wedge C(x) \subseteq D(x)}\frac{|D(x) \cap C(x)|}{|C(x)|}, \qquad (2)$$

and that $|D(x) \cap C(x)|/|C(x)|$ is the confidence of the rule $C(x) \rightarrow D(x)$ (the meaning of the rule $C(x) \rightarrow D(x)$ will be explained later). From this, one can discern the difference between $\Gamma(C, D)$ and $\gamma(C, D)$ easily. In $\gamma(C, D)$, if $|D(x) \cap C(x)|/|C(x)| < 1$, then *x* is not counted, while in $\Gamma(C, D)$ every object is counted by a fraction $|D(x) \cap C(x)|/|C(x)|$ that may not be equal to 1.

We show the definition of $\Gamma(C, D)$ by the following two examples.

Example 2: In Table 1, $A = \{a, b, c, d\}$, $U = \{e1, e2, e3, e4, e5, e6, e7\}$. We use Equation 1 to calculate $\Gamma(C, D)$ when $C = \{a, b, c\}$, $D = \{d\}$. Because $C(e1) = \{e1\}$, $C(e2) = \{e2\}$, $C(e3) = \{e3\}$, $C(e4) = \{e4\}$, $C(e5) = \{e5\}$, $C(e6) = \{e6\}$, $C(e7) = \{e7\}$, $D(e1) = D\{e4\} = D\{e5\} = \{e1, e4, e5\}$, $D\{e2\} = D\{e3\} = D\{e6\} = D\{e7\} = \{e2, e3, e6, e7\}$, we have

$$\Gamma(C, D) = \left(\frac{|D(e1) \cap C(e1)|}{|C(e1)|} + \frac{|D(e2) \cap C(e2)|}{|C(e2)|} + \frac{|D(e3) \cap C(e3)|}{|C(e3)|} + \frac{|D(e4) \cap C(e4)|}{|C(e4)|} + \frac{|D(e5) \cap C(e5)|}{|C(e5)|} + \frac{|D(e6) \cap C(e6)|}{|C(e6)|} + \frac{|D(e7) \cap C(e7)|}{|C(e7)|}\right)\Big/ 7$$
$$= (1+1+1+1+1+1+1)/7 = 1.$$

Example 3: Also in Table 1, we calculate $\Gamma(C, D)$ and $\gamma(C, D)$ when $C = \{\mathrm{a}\}$, $D = \{d\}$. Because $C(e1) = C(e2) = C(e3) = C(e7) = \{e1, e2, e3, e7\}$, $C(e4) = C(e5) = C(e6) = \{e4, e5, e6\}$, $D(e1) = D(e4) = D(e5) = \{e1, e4, e5\}$, $D(e2) = D(e3) = D(e6) = D(e7) = \{e2, e3, e6, e7\}$, we have

$$\Gamma(C, D) = (1/4 + 3/4 + 3/4 + 2/3 + 2/3 + 1/3 + 3/4)/7 = 25/42,$$

while we have $\gamma(C, D) = 0$ according to Equation 2.

Next we will interpret $\Gamma(C, D)$ from another point of view, i.e., we will change our viewpoint from the equivalence classes to minimal decision rules.

Connections Between Γ(C, D) and Minimal Rules

We first give the definition of a minimal formula.

Definition 2 (Minimal Formula): A formula $\Phi \in For\ (B)$ is called a minimal formula in $For\ (B)$ if $supp(\Phi) \neq \emptyset$, and for any $\Psi \in For\ (B)$, $supp(\Psi) \subset supp(\Phi)$ implies $supp(\Psi) = \emptyset$.

The minimal formula has the meaning that the support of a formula $\Psi \in For\ (B)$ cannot be smaller than the support of the minimal formula unless Ψ has an empty support. For example, in Table 1, let $B = \{a, b\}$, then $a = Y \wedge b = Y$ is a minimal formula in *For* (*B*). The support of $a = Y \wedge b = Y$ is $\{e1, e2, e3\}$. We make the following notes:

1. Every formula whose support is not empty in *For* (*B*) can be expressed as a conjunction of some minimal formulae. For example, let $B = \{a, b\}$, the formula $a = N$ can be expressed as

$$a = N \wedge (b = Y \vee b = N) \Leftrightarrow (a = N \wedge b = Y),$$
$$\vee\ (a = N \wedge b = N),$$

where $a = N \wedge b = Y$ and $a = N \wedge b = N$ are minimal formulae in *For* (*B*); the formulae $\sim (a = Y \wedge b = N)$ can be expressed as

$$a = N \vee b = Y \Leftrightarrow (a = N \wedge b = Y)$$
$$\vee\ (a = N \wedge b = N) \vee (a = Y \wedge b = Y),$$

where $a = N \wedge b = Y$, $a = N \wedge b = N$, and $a = Y \wedge b = Y$ are minimal formulae in *F or* (*B*).

2. If $B = \{b_1, b_2, b_3, \ldots, b_l\}$, then the minimal formula in *For* (*B*) has a expression

$$b_1 = w_1 \wedge b_2 = w_2 \wedge \cdots \wedge b_l = w_l,$$

where $w_1 \in V_{b_1}, w_2 \in V_{b_2}, w_3 \in V_{b_3}, \ldots, w_l \in V_{b_l}$.

We call a rule a minimal rule if both its condition part and decision part are minimal formulae, and we define the minimal rule formally, as follows.

Definition 3 (Minimal Rule): Let $C = \{c_1, c_2, c_3, \ldots, c_n\}$, $D = \{d_1, d_2, d_3, \ldots, d_m\}$.
Then we call the rule

$$\begin{aligned} c_1 = u_1 \wedge c_2 = u_2 \wedge \cdots \wedge c_n = u_n \rightarrow d_1 \\ = v_1 \wedge d_2 = v_2 \wedge \cdots \wedge d_m = v_m \end{aligned}$$

a minimal rule, where $u_1 \in V_{c_1}, u_2 \in V_{c_2}, u_3 \in V_{c_3}, \ldots, u_n \in V_{c_n}$, $v_1 \in V_{d_1}, v_2 \in V_{d_2}, \ldots, v_m \in V_{d_m}$.

If $x \in U$, by $C(x) \rightarrow D(x)$ we denote the rule

$$\begin{aligned} c_1 = c_1(x) \wedge c_2 = c_2(x) \wedge c_3 = c_3(x) \wedge \cdots \wedge c_n \\ = c_n(x) \rightarrow d_1 = d_1(x) \wedge d_2 = d_2(x) \wedge \cdots \wedge d_m \\ = d_m(x), \end{aligned}$$

where $C(x)$ is the C-Class containing x, $D(x)$ is the D-class containing x, $c_i(x)$ is the value of x at the attribute c_i, and $d_j(x)$ is the value of x at the attribute d_j.

Note that the rule $C(x) \rightarrow D(x)$ is a minimal rule, and that any minimal rule, whose confidence and strength are not equal to zero, can be written as $C(x) \rightarrow D(x)$.

Let $MinR(C, D)$ be the set of all the minimal rules, r be any rule in $MinR(C, D)$, $Con(r)$ be the confidence of the rule r, and $Str(r)$ be the strength of the rule r. Then

$$\sum_{r \in MinR(C,D)} Str(r) \cdot Con(r), \tag{3}$$

the weighted average of the confidence $Con(r)$ of minimal rule r weighted by the strength $Str(r)$, is exactly the generalized dependency degree $\Gamma(C, D)$. This is our second form of the generalized dependency degree $\Gamma(C, D)$; it is defined in terms of minimal rules. We explain this by the following.

Let X be a $(C \cup D)$-class. Then for any $y, x \in X$, y has the same values as x at the attributes in $(C \cup D)$, and so y has the same values as x at the attributes in both C and D, i.e., y and x are both in the same C-class and in the same D-class, i.e., $C(y) = C(x)$, $D(y) = D(x)$, and therefore for any $x \in X$ we can denote $C(x)$ by $C(X)$, $D(x)$ by $D(X)$, and $|D(x) \cap C(x)|/|C(x)|$ by $|D(X) \cap C(X)|/|C(X)|$. Because $|X| = |D(X) \cap C(X)|$, we have

$$\begin{aligned} \Gamma(C, D) &= \frac{1}{|U|} \sum_{x \in U} \frac{|D(x) \cap C(x)|}{|C(x)|} \\ &= \frac{1}{|U|} \sum_{X \in U/(C \cup D)} \sum_{x \in X} \frac{|D(x) \cap C(x)|}{|C(x)|} \\ &= \frac{1}{|U|} \sum_{X \in U/(C \cup D)} \sum_{x \in X} \frac{|D(X) \cap C(X)|}{|C(X)|} \\ &= \frac{1}{|U|} \sum_{X \in U/(C \cup D)} |X| \frac{|D(X) \cap C(X)|}{|C(X)|} \\ &= \frac{1}{|U|} \sum_{X \in U/(C \cup D)} \frac{|D(X) \cap C(X)|^2}{|C(X)|} \\ &= \sum_{X \in U/(C \cup D)} \frac{1}{|U|} \frac{|D(X) \cap C(X)|^2}{|C(X)|} \\ &= \sum_{X \in U/(C \cup D)} \frac{|D(X) \cap C(X)|}{|U|} \cdot \frac{|D(X) \cap C(X)|}{|C(X)|} \\ &= \sum_{X \in U/(C \cup D)} Str(C(X) \rightarrow D(X)) \\ &\qquad \cdot Con(C(X) \rightarrow D(X)) \\ &= \sum_{r \in M\ in\ R(C,D)} Str(r) \cdot Con(r) \end{aligned}$$

The dependency degree $\gamma(C, D)$ can be rewritten correspondingly as

$$\gamma(C, D) = \sum_{r \in MinR(C,D) \wedge Con(r) = 1} Str(r) \cdot Con(r),$$

which means that in $\gamma(C, D)$, only those minimal rules whose confidences are equal to 1 are counted while in $\Gamma(C, D)$, every minimal rule whose confidence is not equal to zero is counted. In other words, $\gamma(C, D)$ only counts deterministic minimal rules while $\Gamma(C, D)$ counts both deterministic minimal rules and indeterministic minimal rules.

In fact, we can include $\gamma(C, D)$ and $\Gamma(C, D)$ in a general form $\gamma^{\varepsilon}(C, D)$, which is defined as

$$\gamma^{\epsilon}(C, D) = \sum_{r \in MinR(C,D) \wedge Con(r) \geq \epsilon} Str(r) \cdot Con(r).$$

When $\varepsilon = 0$, $\gamma^{\varepsilon}(C, D) = \Gamma(C, D)$, while when $\varepsilon = 1$, $\gamma^{\varepsilon}(C, D) = \gamma(C, D)$. In this article, we only focus on $\Gamma(C, D)$.

Properties of the Generalized Dependency Degree

Recall that in the introduction section, we define the P-indiscernibility relation for a subset P of attributes, denoted by $IND(P)$, which is an equivalence relation on U, the universe of objects. $\Gamma(C, D)$ is actually defined on two equivalence relations induced by subsets C and D of attributes. The definition of $\Gamma(C, D)$ can be easily generalized to the definition of $\Gamma(R_1, R_2)$ for any two equivalence relations R_1 and R_2 on the universe U as follows:

$$\Gamma(R_1, R_2) = \frac{1}{|U|} \sum_{x \in U} \frac{|R_2(x) \cap R_1(x)|}{|R_1(x)|}, \tag{4}$$

$$\Gamma(R_1, R_2) = \sum_{r \in MinR(R_1, R_2)} Str(r) \cdot Con(r). \tag{5}$$

Here the set $MinR(R_1, R_2)$ is the set of all the minimal rules, r is any rule in $MinR(R_1, R_2)$, and by $Con(r)$ and $Str(r)$ we denote the confidence and strength of the rule r, respectively. The minimal rule in $MinR(R_1, R_2)$ is defined as

$$x \in G \rightarrow x \in H,$$

where G and H are any R_1-class and R_2-class, respectively.

Note that $\Gamma(C, D) = \Gamma(IND(C), IND(D))$. In fact, Lingras and Yao (1998) extends the rough set model to any binary relation. Equation 4 is a general form, in which the equivalence relations can be understood as any binary relations. This explains why we can say that the first form of the generalized dependency degree is a flexible form. In this article, we only focus on the case of equivalence relations.

The definition of $\gamma(C, D)$ can also be generalized to $\gamma(R_1, R_2)$ for any equivalence relations R_1, R_2 on the universe U. We rewrite $\gamma(R_1, R_2)$ as follows:

$$\gamma(R_1, R_2) = \frac{1}{|U|} \sum_{x \in U \wedge R_1(x) \subseteq R_2(x)} \frac{|R_2(x) \cap R_1(x)|}{|R_1(x)|}, \quad (6)$$

$$\gamma(R_1, R_2) = \frac{1}{|U|} \sum_{x \in U \wedge R_1(x) \subseteq R_2(x)} Str(r) \cdot Con(r). \quad (7)$$

Throughout the rest of this article, all the relations we use are all on the finite universe U, and the set of all equivalence relations on U is denoted by $\mathcal{ER}(U)$. By the definition of $\gamma(R_1, R_2)$ and $\Gamma(R_1, R_2)$ we have

Theorem 1: For any equivalence relations R_1 and R_2, the inequality $0 \leq \gamma(R_1, R_2) \leq \Gamma(R_1, R_2) \leq 1$ holds.

This theorem shows that $\Gamma(R_1, R_2)$ can serve as an index because it is between zero and one. Moreover, it reveals the relation between $\Gamma(R_1, R_2)$ and $\gamma(R_1, R_2)$. By the next theorem, we will show their relation further in the extreme condition that one of them is equal to one.

Theorem 2: For any equivalence relations R_1 and R_2, $\gamma(R_1, R_2) = 1 \Leftrightarrow \Gamma(R_1, R_2) = 1 \Leftrightarrow \gamma(R_1, R_2) = \Gamma(R_1, R_2)$.

Proof: According to Equations 4 and 6, the conclusion follows immediately.

By the following theorem, we will reveal how $\Gamma(R_1, R_2)$ changes when the second equivalence relation R_2 is changed to be larger.

Theorem 3 (Partial Order Preserving Property): For any equivalence relations R_1, R_2 and R. If $R_2 \subseteq R$, then $\Gamma(R_1, R_2) \leq \Gamma(R_1, R)$.

Proof: According to Equation 4, the conclusion follows immediately.

This means that the finer the equivalence relation R_2 is, the less the equivalence relation R_2 depends on the equivalence relation R_1. From the viewpoint of classification, the more the decision attribute values group together, i.e., the larger the equivalence class induced by the decision attribute is, the easier we can classify the objects into the new D-class by employing attribute C.

TABLE 2. Influenza data.

	a	b	c	d
e1	*Y*	*Y*	*0*	*Z*
e2	*Y*	*Y*	*1*	*Z*
e3	*Y*	*Y*	*2*	*Z*
e4	*N*	*Y*	*0*	*Z*
e5	*N*	*N*	*1*	*Z*
e6	*N*	*Y*	*2*	*Z*
e7	*Y*	*N*	*1*	*Z*

Example 4: In Table 1, Let $C = \{a\}$, $D = \{d\}$, $V_d = \{Y, N\}$; if we group Y and N together such that both Y and N become a new value Z, then $D' = \{d\}$, $V_d = \{Z\}$, and Table 1 becomes Table 2. D induces the equivalence relation $IND(D)$, and the set of the equivalence classes is calculated as $U/D = \{\{e1, e4, e5\}, \{e2, e3, e6, e7\}\}$; on the other hand, D' induces the equivalence relation $IND(D')$, and

$$U/D' = \{\{e1, e2, e3, e4, e5, e6, e7\}\}.$$

Let $R_1 = IND(C)$, $R_2 = IND(D)$, $R = IND(D') = U \times U$, then $\Gamma(R_1, R_2) = 25/42$ as shown in Example 3. For each $x \in U$, $R(x) = U$, so we have $R(x) \cap R_1(x) = R_1(x)$, and therefore

$$\begin{aligned} \Gamma(R_1, R) &= 1/|U| \textstyle\sum_{x \in U} |R(x) \cap R_1(x)|/|R_1(x)| \\ &= 1/|U| \textstyle\sum_{x \in U} |R_1(x)|/|R_1(x)| = 1. \end{aligned}$$

The inequality $\Gamma(R_1, R_2) \leq \Gamma(R_1, R)$ means that we can classify objects into U/D' more easily than into U/D.

Theorem 3 leads to the following theorem, which shows the properties of $\Gamma(R_1, R_2)$ when R_2 becomes the smallest equivalence relation (the identity relation) or the largest equivalence relation (the universal relation).

Theorem 4: For any given equivalence relation R_1,

$$\min_{R_2 \in R(U)} \Gamma(R_1, R_2) = \Gamma(R_1, I_U), \max_{R_2 \in R(U)} \Gamma(R_1, R_2), \\ = \Gamma(R_1, U \times U) = 1$$

where I_U is the identity relation on U, and $U \times U$ is the universal relation on U.

Proof: Since $I_U \subseteq R_2 \subseteq U \times U$, the conclusion is immediate by Theorem 3.

In order to obtain more information about properties of the generalized dependency degree $\Gamma(R_1, R_2)$, we need the following lemma.

Lemma 1: The inequality

$$\frac{a_1^2}{b_1} + \frac{a_2^2}{b_2} + \cdots + \frac{a_n^2}{b_n} \geq \frac{(a_1 + a_2 + \cdots a_n)^2}{b_1 + b_2 + \cdots + b_n}$$

holds for any real number a_i, and real number $b_i > 0$, $i = 1, 2, \ldots, n$.

Proof: It is well known that for any function $f(x)$ in which $f''(x) > 0$, the inequality

$$f(\mu_1 x_1 + \mu_2 x_2 + \cdots + \mu_n x_n) \\ \leq \mu_1 f(x_1) + \mu_2 f(x_2) + \cdots + \mu_n f(x_n)$$

holds if $\mu_1, \mu_2, \ldots, \mu_n \geq 0, \mu_1 + \mu_2 + \cdots + \mu_n = 1$. In this inequality, let

$$f(x) = x^2,\ x_i = (b_1 + b_2 + \cdots + b_n)a_i/b_i, \\ \mu_i = b_i/(b_1 + b_2 + \cdots + b_n),$$

$i = 1, 2, \ldots, n$. Then the desired inequality follows.

In Theorem 3, we have shown the partial order preserving property of $\Gamma(R_1, R_2)$ on the second item. By the next theorem, we continue to show the antipartial order preserving property of $\Gamma(R_1, R_2)$ on the first item. We first show the third form of the generalized dependency degree.

Suppose that there are m R_2-classes, denoted by $X_1, X_2, X_3, \ldots, X_m$. Then we analyze $\Gamma(R_1, R_2)$ for any given R_1. In order to achieve this goal, we need to examine the set X_i. We assume there are k_i different nonempty subsets of X_i of the form $R_1(x) \cap X_i$, for $i = 1, 2, \ldots, m$. Note that for any $x, y \in U$, either $R_1(x) = R_1(y)$ or $R_1(x) \cap R_1(y) = \phi$, and $\cup_{x \in U} R_1(x) = U$. So we can assume that these k_i different nonempty subsets of X_i take the forms

$$R_1(x_{i1}) \cap X_i, R_1(x_{i2}) \cap X_i, \ldots, R_1(x_{ik_i}) \cap X_i,$$

and they satisfy

$$(R_1(x_{ip}) \cap X_i) \cap (R_1(x_{iq}) \cap X_i) = \phi,$$

for $p \neq q, p, q = 1, 2, \ldots, k_i$; and

$$\bigcup_{p=1}^{k_i} (R_1(x_{ip}) \cap X_i) = X_i.$$

Note that for $y \in R_1(x_{ij}) \cap X_i$,

$$R_1(y) \cap X_i = R_1(x_{ij}) \cap X_i.$$

By Equation 4, we have

$$\begin{aligned} \Gamma(R_1, R_2) &= \frac{1}{|U|} \sum_{i=1}^{m} \sum_{x \in X_i} \frac{|X_i \cap R_1(x)|}{|R_1(x)|} \\ &= \frac{1}{|U|} \sum_{i=1}^{m} \sum_{j=1}^{k_i} \sum_{x \in R_1(x_{ij}) \cap X_i} \frac{|X_i \cap R_1(x_{ij})|}{|R_1(x_{ij})|} \\ &= \frac{1}{|U|} \sum_{i=1}^{m} \sum_{j=1}^{k_i} \frac{|X_i \cap R_1(x_{ij})|^2}{|R_1(x_{ij})|} \\ &= \frac{1}{|U|^2} \sum_{i=1}^{m} \sum_{j=1}^{k_i} \frac{|U|}{|R_1(x_{ij})|} |X_i \cap R_1(x_{ij})|^2. \quad (8) \end{aligned}$$

Equation 8 is our third form of the generalized dependency degree. In Algorithm 1, we give the complete description of the computation of $\Gamma(C, D)$ according to Equation 8.

Algorithm 1: Input: $S = (U, A, V, f)$: an information table; C, D: two attribute sets. Output: $\Gamma(C, D)$.

```
PROCEDURE $\Gamma(C, D)$

Find all the $D$-classes $X(1), X(2), \ldots, X(m)$ and all the
  $C$-classes $Y(1), Y(2), \ldots, Y(n)$.
$Total \leftarrow$ the number of cases
for $j = 1$ TO $n$
    $b(j) \leftarrow$ the number of cases in $Y(j)$
end for
$\Gamma \leftarrow 0$
for $i = 1$ TO $m$
    for $j = 1$ TO $n$
        $a(i, j) \leftarrow$ the number of cases in $Y(j) \cap X(i)$
        $\Gamma \leftarrow \Gamma + a(i, j) * a(i, j)/b(j)$
    end for
end for
$\Gamma \leftarrow \Gamma/Total$
RETURN
```

By Theorem 3, we show the partial order preserving property of the generalized dependency degree on the second item, in the following, we continue to show the anti-partial order preserving property of the generalized dependency degree on the first item.

Theorem 5 (Anti-Partial Order Preserving Property): For any equivalence relations R_1, R_2, and R. If $R_1 \subseteq R$, then $\Gamma(R_1, R_2) \geq \Gamma(R, R_2)$.

Proof: Because $R_1 \subseteq R$, each R-class is the union of some R_1-classes, and each set $R(y_j) \cap X_i$ is the union of some sets of the form $R(x_j) \cap X_i$. We assume that, in X_i, there are l_i different nonempty subsets of the form $R(y_{ij}) \cap X_i$. We assume without loss of generality that

$$\begin{aligned} R(y_{i1}) \cap X_i = {} & (R_1(x_{i1}) \cap X_i) \cup (R_1(x_{i2}) \cap X_i) \cup \cdots \\ & \cup (R_1(x_{ip_1}) \cap X_i), \end{aligned}$$

$$R(y_{i1}) \supseteq R_1(x_{i1}) \cup R_1(x_{i2}) \cup \cdots \cup R_1(x_{ip_1}),$$

$$\begin{aligned} R(y_{i2}) \cap X_i = {} & (R_1(x_{ip_1+1}) \cap X_i) \cup (R_1(x_{ip_1+2}) \cap X_i) \\ & \cup \cdots \cup (R_1(x_{ip_2}) \cap X_i), \end{aligned}$$

$$R(y_{i2}) \supseteq R_1(x_{ip_1+1}) \cup R_1(x_{ip_1+2}) \cup \cdots \cup R_1(x_{ip_2}),$$

$$\ldots,$$

$$\begin{aligned} R(y_{il_i}) \cap X_i = {} & (R_1(x_{ip_{l_i-1}+1}) \cup X_i) \cup (R_1(x_{ip_{l_i-1}+2}) \cap X_i) \\ & \cup \ldots \cup (R_1(x_{ip_{l_i}}) \cap X_i), \end{aligned}$$

$$R(y_{il_i}) \supseteq R_1(x_{ip_{l_i-1}+1}) \cup R_1(x_{ip_{l_i-1}+2}) \cup \cdots \cup R_1(x_{ip_{l_i}}).$$

Using Equation 8, we have

$$\begin{aligned} \Gamma(R_1, R_2) &= \frac{1}{|U|^2} \sum_{i=1}^{m} \sum_{j=1}^{k_i} \frac{|U|}{|R_1(x_{ij})|} |X_i \cap R_1(x_{ij})|^2 \\ &= \frac{1}{|U|^2} \sum_{i=1}^{m} \sum_{j=1}^{k_i} \frac{a_{ij}^2}{b_{ij}}, \end{aligned}$$

where $a_{ij} = |X_i \cap R_1(x_{ij})|$, $b_{ij} = |R_1(x_{ij})|/|U|$, $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, k_i$;

$$\Gamma(R, R_2) = \frac{1}{|U|^2}\sum_{i=1}^{m}\sum_{j=1}^{l_i}\frac{|U|}{|R(y_{ij})|}|X_i \cap R(y_{ij})|^2$$

$$= \frac{1}{|U|^2}\sum_{i=1}^{m}\sum_{j=1}^{l_i}\frac{a'^2_{ij}}{b'_{ij}},$$

where

$$a'_{i1} = |X_i \cap R(y_{i1})| = a_{i1} + a_{i2} + \cdots + a_{ip_1},$$

$$a'_{i2} = |X_i \cap R(y_{i2})| = a_{ip_1+1} + a_{ip_1+2} + \cdots + a_{ip_2},$$

$$\vdots$$

$$a'_{il_i} = |X_i \cap R(y_{il_i})| = a_{ip_{l_i-1}+1} + a_{ip_{l_i-1}+2} + \cdots + a_{ip_{l_i}},$$

$$b'_{i1} = |R(y_{i1})|/|U| \geq b_{i1} + b_{i2} + \cdots + b_{ip_1},$$

$$b'_{i2} = |R(y_{i2})|/|U| \geq b_{ip_1+1} + b_{ip_1+2} + \cdots + b_{ip_2},$$

$$\vdots$$

$$b'_{il_i} = |R(y_{il_i})|/|U| \geq b_{ip_{l_i-1}+1} + b_{ip_{l_i-1}+2} + \cdots + b_{ip_{l_i}},$$

$i = 1, 2, \ldots, m$. By Lemma 1, we have

$$\begin{aligned}\sum_{j=1}^{k_i}\frac{a^2_{ij}}{b_{ij}} = {} & \frac{a^2_{i1}}{b_{i1}} + \frac{a^2_{i2}}{b_{i2}} + \cdots + \frac{a^2_{ip_1}}{b_{ip_1}} \\ & + \frac{a^2_{ip_1+1}}{b_{ip_1+1}} + \frac{a^2_{ip_1+2}}{b_{ip_1+2}} + \cdots + \frac{a^2_{ip_2}}{b_{ip_2}} \\ & + \cdots \\ & + \frac{a^2_{ip_{l_i-1}+1}}{b_{ip_{l_i-1}+1}} + \frac{a^2_{ip_{l_i-1}+2}}{b_{ip_{l_i-1}+2}} + \cdots + \frac{a^2_{ip_{l_i}}}{b_{ip_{l_i}}} \\ & \geq \frac{\left(\sum_{j=1}^{p_1} a_{ij}\right)^2}{\sum_{j=1}^{p_1} b_{ij}} + \frac{\left(\sum_{j=p_1+1}^{p_2} a_{ij}\right)^2}{\sum_{j=p_1+1}^{p_1} b_{ij}} \\ & + \cdots + \frac{\left(\sum_{j=p_{l_i-1}+1}^{p_{l_i}} a_{ij}\right)^2}{\sum_{j=p_{l_i-1}+1}^{p_{l_i}} b_{ij}} \geq \frac{a'^2_{i1}}{b'_{i1}} \\ & + \frac{a'^2_{i2}}{b'_{i2}} + \cdots + \frac{a'^2_{il_i}}{b'_{il_i}}.\end{aligned}$$

Therefore,

$$\Gamma(R_1, R_2) = \frac{1}{|U|^2}\sum_{i=1}^{m}\sum_{j=1}^{k_i}\frac{a^2_{ij}}{b_{ij}} \geq \frac{1}{|U|^2}\sum_{i=1}^{m}\sum_{j=1}^{l_i}\frac{a'^2_{ij}}{b'_{ij}} = \Gamma(R, R_2).$$

This means that the finer the equivalence relation R_1 is, the more R_2 depends on R_1. From the viewpoint of classification, the more the condition attribute values group together, i.e., the larger the equivalence class induced by the decision attribute is, the more difficult it is to classify the objects into the new D-class by employing attribute C.

TABLE 3. Influenza data.

	a	b	c	d
e1	*Y*	*Y*	*3*	*N*
e2	*Y*	*Y*	*3*	*Y*
e3	*Y*	*Y*	*3*	*Y*
e4	*N*	*Y*	*3*	*N*
e5	*N*	*N*	*3*	*N*
e6	*N*	*Y*	*3*	*Y*
e7	*Y*	*N*	*3*	*Y*

Example 5: In Table 1, let $C = \{c\}$, $V_c = \{0, 1, 2\}$, $D = \{d\}$ if we group 0, 1 and 2 together such that 0, 1, 2 become a new value 3, then $C' = \{c\}$, $V_c = \{3\}$, and Table 1 becomes Table 3.

In both Table 1 and Table 3, D induces the equivalence relation $IND(D)$, and the set of the equivalence classes is calculated as $U/D = \{\{e1, e4, e5\}, \{e2, e3, e6, e7\}\}$; In Table 1, C induces the equivalence relation $IND(C)$, and the set of the corresponding equivalence classes is

$$U/C = \{\{e1, e4\}, \{e2, e5, e7\}, \{e3, e6\}\}.$$

In Table 3, C' induces the equivalence relation $IND(C')$, and the set of the corresponding equivalence classes is

$$U/C' = \{\{e1, e2, e3, e4, e5, e5, e6, e7\}\}.$$

Let $R_1 = IND(C)$, $R_2 = IND(D)$, $R = IND(C') = U \times U$.

$$\begin{aligned}\Gamma(R_1, R_2) &= 1/|U|\textstyle\sum_{x\in U}|R_2(x) \bigcap R_1(x)|/|R_1(x)| \\ &= 1/7(2/2 + 2/3 + 2/2 + 2/2 + 1/3 \\ &\quad + 2/2 + 2/3) \\ &= 17/21.\end{aligned}$$

For each $x \in U$, $R(x) = U$, we have $R(x) \bigcap R_2(x) = R_2(x)$, and therefore

$$\begin{aligned}\Gamma(R, R_2) &= 1/|U| \textstyle\sum_{x\in U}|R_2(x) \cap R(x)|/|R(x)| \\ &= 1/|U| \textstyle\sum_{x\in U}|R_2(x)|/|R(x)| \\ &= 1/7(3/7 + 4/7 + 4/7 + 3/7 + 3/7 \\ &\quad + 4/7 + 4/7) \\ &= 25/49.\end{aligned}$$

The inequality $\Gamma(R_1, R_2) > \Gamma(R, R_2)$ means that it is harder for us to classify objects into D-class by employing the attribute C' than employing the attribute C.

Because $IND(C) = \cap_{c\in C} IND(\{c\})$, when we drop some attributes from C such that a new attribute set C' is formed, we have $IND(C') \supseteq IND(C)$. So by Theorem 5, we have $\Gamma(C', D) \leq \Gamma(C, D)$. This means that generally, the less the condition attribute set contains attributes, the harder we can classify the objects into D-class by employing the condition attribute set.

The next theorem shows the extreme cases when R_1 becomes the smallest equivalence relation (the identity relation) or the largest equivalence relation (the universal relation).

DOI: 10.1002/asi

Theorem 6: For any given equivalence relation R_2, we have

$$\max_{R_1 \in R(U)} \Gamma(R_1, R_2) = \Gamma(I_U, R_2) = 1,$$

$$\min_{R_1 \in R(U)} \Gamma(R_1, R_2) = \Gamma(U \times U, R_2).$$

Proof: This follows immediately from Theorem 5.

In the following theorem, we show the extreme cases when both R_1 and R_2 vary.

Theorem 7:

$$\min_{R_1, R_2 \in \mathcal{ER}(U)} \Gamma(R_1, R_2) = \frac{1}{|U|}, \max_{R_1\ R_2 \in \mathcal{ER}(U)} \Gamma(R_1, R_2) = 1$$

Proof: By Theorem 3 and Theorem 5, we only need to verify that $\Gamma(U \times U, I_U) = 1/$ u$|U|$. Let $R_1 = U \times U$, $R_2 = I_U$. According to Equation 4, we have

$$\Gamma(U \times U, I_U) = \frac{1}{|U|} \sum_{x \in U} \frac{|R_2(x) \cap R_1(x)|}{|R_1(x)|}$$

$$= \frac{1}{|U|} \sum_{x \in U} \frac{|\{x\} \cap U|}{|U|}$$

$$= \frac{1}{|U|} \sum_{x \in U} \frac{1}{|U|} = \frac{1}{|U|}.$$

Then the desired conclusion follows.

This means that for any two equivalence relations R_1 and R_2, R_2 depends on R_1 to a degree of at least $1/|U|$ and that we can infer some information about R_2 even when R_1 contains no useful information about R_2. This arises from the fact that R_2 contains useful information about itself. However, in the extreme case when R_1 is the universal relation, R_2 is the identity relation (the identity relation contains little information about itself), and the number of objects tends to infinity, the degree that R_2 depends on R_1 tends to zero.

Extension of Γ to Incomplete Information Systems

In this section, we expand the definition of the generalized dependency degree to incomplete information systems by reinterpreting the meaning of the support of a formula and the cardinality of the support in incomplete information systems.

If an information system has some missing values, we call this information system an incomplete information system. For example, there are three missing values in Table 4, indicated by "*".

How to Handle Missing Values in Incomplete Information Systems

Here, we introduce an approximate approach by replacing each missing value by its possible distributions as shown in Table 5.

TABLE 4. Influenza data.

	a	b	c	d
e1	*Y*	*Y*	*Normal (0)*	*N*
e2	*Y*	*	*High (1)*	*Y*
e3	*Y*	*Y*	*	*Y*
e4	*N*	*	*Normal (0)*	*N*
e5	*N*	*N*	*High (1)*	*N*
e6	*N*	*Y*	*Very high (2)*	*Y*
e7	*Y*	*N*	*High (1)*	*Y*

TABLE 5. Influenza data.

	a	b	c	d
e1	*Y*	*Y*	*Normal (0)*	*N*
e2	*Y*	$\{P_1/Y, P_2/N\}$	*High (1)*	*Y*
e3	*Y*	*Y*	$\{S_1/0, S_2/1, S_3/2\}$	*Y*
e4	*N*	$\{Q_1/Y, Q_2/Y\}$	*Normal (0)*	*N*
e5	*N*	*N*	*High (1)*	*N*
e6	*N*	*Y*	*Very high (2)*	*Y*
e7	*Y*	*N*	*High (1)*	*Y*

In the e_2-row, by $\{P_1/Y, P_2/N\}$ we mean that e_2 takes the value Y with a probability of P_1, and N with a probability of P_2. In the e_4-row, the expression $\{Q_1/Y, Q_2/N\}$ has a similar meaning. In e_3-row, $\{S_1/0, S_2/1, S_3/2\}$ means that e_3 takes the value 0, 1, and 2 with probability S_1, S_2, and S_3, respectively.

In order to reduce the complexity of computing, we introduce an approximate method for determining the values of all the unknown parameters $P_1, P_2, Q_1, Q_2, S_1, S_2, S_3$. We let P_1, P_2, Q_1, Q_2 take the values of the distribution of Y and N in column b, i.e., $P_1 = Q_1 = 3/5$, $P_2 = Q_2 = 2/5$; and we let S_1, S_2, S_3 take the values of the distribution of 0, 1 and 2 in column c, i.e., $S_1 = 2/6$, $S_2 = 3/6$, $S_3 = 1/6$.

Definition of Γ in Incomplete Information Systems

Although we can also define some kinds of equivalence relations induced by the attributes in an incomplete information table, here we introduce a direct way to calculate the generalized dependency degree Γ in an incomplete information table. That is, we choose Equation 3

$$\Gamma(C, D) = \sum_{r \in MinR(C,D)} Str(r) \cdot Con(r),$$

as our definition of the generalized dependency degree in an incomplete information table. To carry out this idea, we have to define the confidence and the strength of a rule in an incomplete information table. We show our definition using the example of the Influenza Data in Table 5.

Before going forward, we need to re-interpret the meaning of $supp(\Phi)$ and the meaning of $|supp(\Phi)|$ where the set $supp(\Phi)$ may be a "fractional" set in an incomplete information table. Here, we interpret $supp(\Phi)$ as a fuzzy set.

If $x \in U$ satisfies Φ with a probability of p, then we consider that the object x belongs to the set $supp(\Phi)$ with a membership of p, and we write the element x in $supp(\Phi)$ as p/x. For example, in Table 5, let Φ be the formula $b = Y$. e_1 satisfies the formula $b = Y$ with a probability of 1, the probability of Y in e_1-row, b-column, while e_2 satisfies the formula $b = Y$ with a probability of $P_1 = 3/5$, the probability of Y in e_2-row, b-column. We have

$$supp(\Phi) = \{1/e_1, 0.6/e_2, 1/e_3, 0.6/e_4, 0/e_5, 1/e_6, 0/e_7\}.$$

We can delete all the elements whose probabilities are equal to zero, i.e., we can write $supp(\Phi)$ as $supp(\Phi) = \{1/e_1, 0.6/e_2, 1/e_3, 0.6/e_4, 1/e_6\}$.

Then we define the fuzzy set $supp(\Phi)$ inductively as follows: If x belongs to $supp(\Phi)$ with a membership of $\mu_{supp(\Phi)}(x) = p$, and x belongs to $supp(\Psi)$ with a membership of $\mu_{supp(\Psi)}(x) = q$, then x belongs to $supp(\Phi \wedge \Psi)$ with a membership of $\mu_{supp(\Phi \wedge \Psi)}(x) = pq$, x belongs to $supp(\sim\Phi)$ with a membership of $\mu_{supp(\sim\Phi)}(x) = 1 - p$, and x belongs to $supp(\Phi \vee \Psi)$ with a membership of $\mu_{supp(\Phi\vee\Phi)}(x) = 1 - (1 - p)(1 - q)$. Formally $supp(\Phi)$ is defined inductively in terms of algebraic operations of a fuzzy set as follows:

$F1 : supp(a = v) = \{\mu(x)/x | x \in U, P(a(x) = v) = \mu(x)\}$ for $a \in B$ and $v \in V_a$
$F2 : supp(\Phi \vee \Psi) = supp(\Phi) + supp(\Psi)$
$F3 : supp(\Phi \wedge \Psi) = supp(\Phi) \cdot supp(\Psi)$
$F4 : supp(\sim\Phi) = \sim supp(\Phi)$

where $supp(\Phi) + supp(\Psi)$ is the algebraic sum of the fuzzy sets $supp(\Phi)$ and $supp(\Psi)$, $supp(\Phi) \cdot supp(\Psi)$ is the algebraic product of the fuzzy sets $supp(\Phi)$, and $supp(\Psi)$, and $\sim supp(\Phi)$ is the complement of the fuzzy sets $supp(\Phi)$ (Zimmerman, 2001).

The cardinality $|supp(\Phi)|$ can be defined in term of the fuzzy set, i.e.,

$$|supp(\Phi)| = \sum_{x \in U} \mu_{supp(\Phi)}(x). \tag{9}$$

Next, as an example, we will calculate the generalized dependency degree between $C = \{a, b, c\}$ and $D = \{d\}$ in Table 5 by Equation 3. First, we need to calculate the confidence and strength of each minimal decision rule using the following definitions:

$$Con(\Phi \rightarrow \Psi) = |supp(\Phi \wedge \Psi)| / |supp(\Phi)|, \tag{10}$$

$$Str(\Phi \rightarrow \Psi) = |supp(\Phi \wedge \Psi)| / |U|. \tag{11}$$

Example 5: We show in the following calculation process the confidence and strength of one minimal rule; the results of all the other minimal rules are listed in Table 6. Since

$supp(a = Y \wedge b = Y \wedge c = 0 \wedge d = Y) = \{S_1/e_3\}$,
$|\{S_1/e_3\}| = S_1 = 2/6$, $supp(\text{a} = Y \wedge b = Y \wedge c = 0) = \{1/e_1, S_1/e_3\}$
$|\{1/e_1, S_1/e_3\}| = 1 + S_1 = 1 + 2/6 = 4/3$,

we have the minimal rule $a = Y \wedge b = Y \wedge c = 0 \rightarrow d = Y$ with confidence = 1/4, strength = 1/21. So, we have

TABLE 6. Results of all minimal rules.

a	b	c	d	Con	Str	a	b	c	d	Con	Str
Y	*Y*	*0*	*Y*	*1/4*	*1/21*	*N*	*Y*	*0*	*Y*	*0*	*0*
Y	*Y*	*0*	*N*	*3/4*	*1/7*	*N*	*Y*	*0*	*N*	*1*	*3/35*
Y	*Y*	*1*	*Y*	*1*	*11/70*	*N*	*Y*	*1*	*Y*	*0*	*0*
Y	*Y*	*1*	*N*	*0*	*0*	*N*	*Y*	*1*	*N*	*0*	*0*
Y	*Y*	*2*	*Y*	*1*	*1/42*	*N*	*Y*	*2*	*Y*	*1*	*1/7*
Y	*Y*	*2*	*N*	*0*	*0*	*N*	*Y*	*2*	*N*	*0*	*0*
Y	*N*	*0*	*Y*	*0*	*0*	*N*	*N*	*0*	*Y*	*0*	*0*
Y	*N*	*0*	*N*	*0*	*0*	*N*	*N*	*0*	*N*	*1*	*2/35*
Y	*N*	*1*	*Y*	*1*	*1/5*	*N*	*N*	*1*	*Y*	*0*	*0*
Y	*N*	*1*	*N*	*0*	*0*	*N*	*N*	*1*	*N*	*1*	*1/7*
Y	*N*	*2*	*Y*	*0*	*0*	*N*	*N*	*2*	*Y*	*0*	*0*
Y	*N*	*2*	*N*	*0*	*0*	*N*	*N*	*2*	*N*	*0*	*0*

$$\begin{aligned}\Gamma(C, D) &= \sum_{r \in M\,in\,R(C,D)} Str(r) \cdot Con(r) \\ &= 1/21 \cdot 1/4 + 1/7 \cdot 3/4 + 11/70 \cdot 1 \\ &\quad + 1/42 \cdot 1 + 1/5 \cdot 1 + 3/35 \cdot 1 + 1/7 \cdot 1 \\ &\quad + 2/35 \cdot 1 + 1/7 \cdot 1 = 13/14.\end{aligned}$$

By the next theorem, we show one more property of the generalized dependency.

Theorem 6: In an incomplete information system, we have $0 \leq \Gamma(C, D) \leq 1$.

Proof: Because every object contributes $1/|U|$ to the sum of $\Sigma_{r \in MinR(C,D)} Str(r)$ and there are $|U|$ objects in total, we have $\Sigma_{r \in MinR(C,D)} Str(r) = 1$. It is obvious that $Con(r) \leq 1$ for any rule r, so we have

$$\sum\nolimits_{r \in MinR(C,D)} Str(r) \cdot Con(r) \leq \sum\nolimits_{r \in MinR(C,D)} Str(r) = 1.$$

Note that our method enables us to handle an information table whose values are probabilistic distributions, and that an information table without missing values can be understood as a special case of an incomplete information table.

We also note that the method of handling missing values introduced in this section is one of many possible ways. If the method of handling missing values is changed to a new one, we can still use Equation 3 as our definition of the generalized dependency degree Γ in an incomplete information table by re-interpreting correspondingly the meaning of $supp(\Phi)$ and the meaning of $|supp(\Phi)|$.

Discussion: Comparison with the Conditional Entropy

Yao (2003a, 2003b) classifies rules into two types: one-way rule and two-way rule. The generalized dependency degree is in fact a measure for one-way rule, which is different from the eight information measures for one-way rule summarized in (Yao, 2003b). Among these eight information measures, the conditional entropy is a well-known measure, and we will make a comparison between the generalized dependency degree and the conditional entropy in this section.

Malvestuto (1986), Lee (1987), and Nambiar (1980) introduce the idea of applying the Shannon entropy function to measure the "information content" of the data in the columns of an attribute set. They extend the idea to develop a measure that, given a finite table T, quantifies the amount of information the columns of C contain about D. This measure is the conditional entropy (Giannella & Robertson, 2004). The formulation for the conditional entropy is as follows:

$$\begin{aligned} H(D|C) &= -\sum_c \sum_d \Pr(c) \cdot \Pr(d|c) \cdot \log_2(\Pr(d|c)) \\ &= -\sum_c \Pr(c) \cdot \sum_d \Pr(d|c) \cdot \log_2(\Pr(d|c)), \end{aligned}$$

where c and d denote the vectors consisting of the values of attributes in C and in D, respectively.

Dalkilic and Robertson (2000) refer to the conditional entropy as an information dependency measure, denoted by $H_{C\rightarrow D}$. They develop a variety of arithmetic inequalities for this measure. The formulation of the entropy is

$$H(D) = -\sum_d \Pr(d) \cdot \log_2(\Pr(d)).$$

The conditional entropy is well discussed in the literature of Information Theory (Cover, 1991; Yeung, 2002), and it is used in the C4.5 decision tree algorithm (Quinlan, 1993) and the latest open-source version C4.5R8 (Quinlan, 1996).

The generalized dependency degree and the conditional entropy are similar in two different aspects:

- Both the generalized dependency degree and the conditional entropy measure the degree to which D depends on C.
- The generalized dependency degree is computed as a type of weighted average of the confidence of decision rules, and conditional entropy averages over the logarithm of the confidence of decision rules. The same weights are employed in both the generalized dependency degree and the conditional entropy.

However, the generalized dependency degree and the conditional entropy are different in three aspects:

- The value of the conditional entropy is between zero and infinity, while the value of the generalized dependency degree is between zero and one, and from this point of view, the generalized dependency degree can serve directly as an index.
- The first form of the conditional entropy is defined in terms of equivalence relations, and so it can be extended to binary relations.
- To compute the generalized dependency degree by the third form, we need only to carry out simple arithmetic operations, while to compute the conditional entropy, we have to compute the logarithm of the frequency, a time-consuming operation.

The idea for Γ is based on the idea of rough set, we have compared Γ with γ in Equations 1, 2, 4, 5, 6, and 7. In the next section, on one hand, we will compare Γ with the conditional entropy on their applications in the decision tree classifier in which the attribute is selected one by one according to its conditional entropy or Γ value on the current node. On the other hand, we will conduct some experiments to make an empirical comparison between Γ and γ on their applications in attribute selection in which attributes are selected as a subset according to its Γ value or its γ value.

Experiments

In the above sections, we have given a detailed explanation of the generalized dependency degree by presenting its various forms and developing its various properties. In this section, we will show its significance in decision trees and attribute selection.

There are several reasons to choose C4.5R8 decision tree classifier for our comparison. First and the most important, C4.5R8 uses the conditional entropy that we want to compare with Γ, while neural networks do not use the conditional entropy. Second, C4.5R8 can handle continuous attributes and missing values, which makes it easy to compare Γ with the conditional entropy in various cases-handling discrete attribute, handling continuous attributes, and handling missing values. Third, compared to other classifiers, a decision tree can be understood easily. Fourth, it often takes large amounts of time to train a neural network, while C4.5R8 decision tree classifier is efficient in training time (Lim, Loh, & Shih, 2000) and thus suitable for large training sets. Lastly, as comparable with neural networks, decision trees already display good classification accuracy (Hassanien, 2004).

Comparison With the Conditional Entropy in Decision Trees

We will replace the conditional entropy used in the C4.5 algorithm with the generalized dependency degree such that a new C4.5 algorithm is formed.

C4.5 has its origins in Hunt's Learning Systems by way of ID3. The latest open-source version of C4.5 is C4.5R8 (Quinlan, 1996). The C4.5R8 algorithm uses a divide-and-conquer approach to grow decision trees. A brief explanation of the C4.5R8 algorithm is given below. For further details, see Quinlan, 1993 and 1996.

The basic idea of the C4.5R8 decision tree algorithm is similar to ID3. It divides the whole training set into smaller subsets until subsets with all or the majority of data corresponding to the same class are created. It generates a decision tree from the whole training set. The whole training set corresponds to the root node. Each of the interior nodes including the root node of the tree is labeled by an attribute, while branches that lead from the node are labeled by the value of the attribute. The leaves of the tree correspond to the classes. The tree construction process is guided by choosing the most informative attribute at each step. Let T be the current set of training cases and $\{O_j, j = 1, 2, \ldots, n\}$ be the set of current attribute values. Then, T is partitioned into subsets $T_1, T_2, \ldots, T_n$, where T_i contains all the cases in T whose current attribute value is O_j. A decision tree is constructed by recursively applying the algorithm to each subset of training cases, so that each branch leads to the decision tree constructed from the subset T_i of training cases. Tree construction stops when all examples in a node are of the same class. This node, called a leaf, is labeled by a value of the class variable. Each leaf is labeled by exactly one class name. However, leaves can also be empty, if there are no training examples having attribute values that would lead to a leaf, and such empty leaf is labeled as the most frequent class in C4.5.

C4.5R8 employs gain criterion and gain ration criterion to select the most informative attribute at each subset of training cases.

If the algorithm is run with option -g, then for every condition attribute a, its information gain is computed by the formula

$$G(D, \{a\}) = \begin{cases} H(D) - H(D|\{a\}), & \\ \qquad\qquad \text{if } a \text{ is a discrete attribute,} & \\ H(D) - H(D|\{a\}) - \log_2(N-1)/|U|, & \\ \qquad\qquad \text{if } a \text{ is a continuous attribute,} & \end{cases}$$

where N is the number of distinct values of the attribute a, and $\log_2(N-1)/|U|$ is used to reduce the bias towards the continuous attribute according to MDL principle. Note that if a is a continuous attribute, $H(D|\{a\})$ is the maximum value of $H(D|\{a^t\})$ among all possible tests, such as $a \leq t$ for a potential threshold t, and the new attribute a^t is defined as a^t = true if $a \leq t$ and a^t = false otherwise. The attribute that has the maximum gain among all the condition attributes is chosen; then the training cases T are partitioned into subsets $T_1, T_2, \ldots, T_n$ according to the value of the chosen attribute. The same procedure is applied recursively to each subset of the training cases. If, instead, the algorithm is run with the default option, then for every condition attribute a, its information gain ratio is computed by the formula

$$\frac{G(D,\{a\})}{H(\{a\})}.$$

If the algorithm is run with option -s, then the values of discrete attributes will be grouped for test, and again the gain ratio criterion will be used. If the algorithm is run with option -g -s, then the values of discrete attributes will be grouped for test, and the gain criterion will be used.

In case of missing values, the information gain for attribute a is computed by the formula

$$G(D, \{a\}) = \begin{cases} P(a) * (H(D) - H(D|\{a\})), & \\ \qquad\qquad \text{if } a \text{ is discrete,} & \\ P(a) * (H(D) - H(D|\{a\}) - \log_2(N-1)/|U|), & \\ \qquad\qquad \text{if } a \text{ is continuous.} & \end{cases}$$

where $p(a)$ is the probability that a is known.

We replace the information gain in the original C4.5R8 algorithm with

$$G(D,\{a\}) = \Gamma(\{a\}, D) - \Gamma(D)$$

in our new C4.5 algorithm, where $\Gamma(D) = \Gamma(U \times U, IND(D))$. Note that by Theorem 6, we have $G(D,\{a\}) \geq 0$. Similar to the conditional entropy, if a is a continuous attribute, $\Gamma(\{a\}, D)$ is the maximum value of $\Gamma(\{a^t\}, D)$ among all possible tests such as $a \leq t$ for a potential threshold t, and the new attribute a^t is defined as a^t = true if $a \leq t$, and a^t = false otherwise. In case of missing values, we use our definition of $\Gamma(C, D)$ in incomplete information systems introduced in the Extension of Γ to Incomplete Information Systems section.

Moreover, in the new C4.5 algorithm, we do not employ the MDL principle by which the original C4.5R8 can correct the split selection bias towards the continuous attribute. Because the conditional entropy has the meaning of average code length, it is compatible with the MDL principle in the original C4.5R8; in contrast, the generalized dependency degree means the degree to which the decision attribute depends on the condition attribute, so the new C4.5 using the generalized dependency degree is not compatible with the MDL principle and so we does not include it in the new C4.5.

One further change we make from the original C4.5R8 is that we stop the procedure of building the tree earlier by applying a new criterion: in the current node, if for every attribute, the number of the gain cases is less than a given value 0.75 and then the splitting procedure stops. The number of the gain cases is calculated by multiplying the number of cases in the current node by the dependency gain.

Both the original C4.5R8 and the new C4.5 are applied to all of the same 20 datasets from the UCI machine learning repository as Quinlan (1996) uses. Note that the datasets we use may have slight differences from those Quinlan (1996) uses. For example, the glass dataset we use has a different order of the cases from Quinlan's. Table 7 is a description of the datasets we use. The first column shows the names of the datasets, the second column gives the numbers of cases in each dataset, the third column gives the number of classes, the fourth column gives the number of continuous attributes, the fifth column gives the number of discrete attributes, and the final column describes whether there are missing values in each dataset.

The experiments are conducted on a workstation whose hardware model is Nix Dual Intel Xeon 2.2GHz, with 1GB of RAM, and whose OS is Linux Kernel 2.4.18-27smp (RedHat7.3). Both algorithms use 10-fold cross-validations with each task. The figures shown in Table 8 are the mean error rate of the 10-fold cross-validations of both the original C4.5R8 and the new C4.5 with the same option -g -s.

The second and fourth columns in Table 8 are the mean error rates (error rate = 100% – classification rate) before and after pruning, respectively, obtained by running with the

TABLE 7. Description of the datasets.

Dataset	Cases	Classes	Cont	Discr	Missing
Anneal	898	6	6	32	Y
Auto	205	6	15	10	Y
Breast-w	699	2	9	0	Y
Colic	368	2	7	15	Y
Credit-a	690	2	6	9	Y
Credit-g	1000	2	7	13	N
Diabetes	768	2	8	0	N
Glass	214	6	9	0	N
Heart-c	303	2	6	7	Y
Heart-h	294	2	8	5	Y
Hepatitis	155	2	6	13	Y
Allhyper	3772	5	7	22	Y
Iris	150	3	4	0	N
Labor	57	2	8	8	Y
Letter	20000	26	16	0	N
Segment	2310	7	19	0	N
Sick	3772	2	7	22	Y
Sonar	208	2	60	0	N
Vehicle	846	4	18	0	N
Wave	300	3	21	0	N

TABLE 8. Mean error rates of the original C4.5 and the new C4.5.

Dataset	O Unpruned (%)	N Unpruned (%)	O Pruned (%)	N Pruned (%)
Anneal	**3.9**	6.1	**4.6**	7.9
Auto	**20.5**	22.0	**22.0**	22.5
Breast-w	5.7	**4.2**	**4.3**	4.5
Colic	19.8	**16.3**	16.0	**15.4**
Credit-a	19.7	**15.2**	17.1	**15.6**
Credit-g	30.5	**27.2**	28.0	**27.0**
Diabetes	**24.7**	26.0	**24.5**	25.6
Glass	**31.2**	31.7	**30.3**	**30.3**
Heart-c	**22.4**	23.4	**21.4**	23.1
Heart-h	24.2	**20.7**	22.8	**21.1**
Hepatitis	20.0	**19.3**	19.9	**19.3**
Allhyper	1.4	**1.1**	1.4	**1.2**
Iris	6.0	**4.0**	6.0	**4.0**
Labor	24.7	**15.7**	26.3	**19.3**
Letter	**11.9**	12.5	**11.9**	12.4
Segment	**3.2**	3.5	**3.2**	3.7
Sick	1.2	**1.0**	1.1	**1.0**
Sonar	**20.7**	27.9	**20.7**	27.9
Vehicle	**27.8**	30.1	**28.0**	30.4
Wave	28.4	**26.0**	28.4	**26.3**
Sum	347.9	333.9	337.9	338.5

option that uses the gain criteria (not the gain ratio) and the grouping method in the original C4.5R8 system. The third and fifth columns are the results before and after pruning, respectively, obtained by running with the same option in the new C4.5 system. This means that when the new gain criteria based on the generalized dependency degree is used, the grouping method is also used, but the MDL principle is not used. In each row of the second and third columns, the smaller result is shown in bold, and so do the fourth and fifth columns. The final row shows the sum of results of the experiments on the 20 datasets.

The values shown in Table 9 describe the average run time of the 10-fold cross-validations. The time unit in Table 9 is 0.01 second. The second column and the fifth column in Table 9 show the average run time of the original procedure C4.5R8 before pruning and after pruning in the 10-fold cross-validations. The third and the fourth columns show the average run time of the new C4.5 without the pruning procedure and the reduced time rate relative to the second column, while the sixth and seventh columns give the corresponding results of the new C4.5 with the pruning procedure. Note that the run time does not include the run time for data preparation for the cross-validation, or the run time for final result reporting, in both C4.5 systems.

The values shown in Table10 describe the average number of leaves of the decision trees of the 10-fold cross-validations.

TABLE 9. Average run time of the original C4.5R8 and the new C4.5.

Dataset	O Unpruned	N Unpruned	Reduced (%)	O Pruned	N Pruned	Reduced (%)
Anneal	6.2	4.600	25.8	6.800	5.100	25.0
Auto	9.6	2.600	72.9	9.700	2.600	73.2
Breast-w	1.5	1.000	33.3	1.600	1.000	37.5
Colic	3.9	1.500	61.5	4.100	1.500	63.4
Credit-a	7	2.400	65.7	8.300	2.500	69.9
Credit-g	9.5	4.700	50.5	11.500	5.200	54.8
Diabetes	4.2	2.400	42.9	4.600	2.600	43.5
Glass	1.4	0.900	35.7	2.300	1.500	34.8
Heart-c	1.7	0.700	58.8	2.000	0.900	55.0
Heart-h	1.6	0.700	56.3	1.900	0.700	63.2
Hepatitis	0.8	0.600	25	0.900	0.700	22.2
Allhyper	40	18.500	53.8	45.000	18.500	58.9
Iris	0.25	0.200	20	0.500	0.400	20.0
Labor	0.2	0.100	50	0.400	0.400	0.0
Letter	7.4	5.540	25.1	8.150	5.940	27.1
Segment	41.1	24.800	39.7	46.700	25.500	45.4
Sick	35.6	17.100	52	38.100	20.800	45.4
Sonar	11.2	5.000	55.4	12.900	5.100	60.5
Vehicle	8.4	5.800	31	10.800	6.300	41.7
Wave	5.7	2.00	64.9	6.8	2.10	69.1

The second and the fourth columns in Table10 show the results of the original procedure C4.5R8 before pruning and after pruning. The third and the fifth columns show the results of the new C4.5 before pruning and after pruning. In each row of the second and third columns, the smaller result is shown in bold, and so do the fourth and fifth columns.

The experiments show that the generalized dependency degree $\Gamma(C, D)$ is a useful measure. We compare three aspects of the new C4.5 algorithm using the generalized dependency degree with the original C4.5R8 algorithm using the conditional entropy:

Speed: To compute $\Gamma(C, D)$, we only need to carry out arithmetic operations, while the computation of the commonly used conditional entropy needs to compute the logarithm of the frequency, a time-consuming operation. Furthermore, the building tree procedure in the new C4.5 algorithm stops earlier in most cases. This explains why the new C4.5 procedure with (or without) the pruning procedure runs much faster than the original C4.5R8 procedure with (or without) the pruning procedure. In fact, the new C4.5 procedure runs in about half of the time required by the original C4.5R8 procedure, and the new C4.5 procedure without pruning procedure can run a little faster still. Note that the original C4.5R8 algorithm is the fastest algorithm in terms of training time among a group of 33 tree-based, rule-based, and statistics-based classification algorithms (Lim et al., 2000).

TABLE 10. Average number of leaves of the original C4.5R8 and the new C4.5.

Dataset	O Unpruned	N Unpruned	O Pruned	N Pruned
Anneal	**139.8**	144	93.4	**83**
Auto	**55.1**	58.1	**45.6**	47.9
Breast-w	41.2	**17.4**	22.2	**15.8**
Colic	80.5	**30.5**	**15.8**	19.1
Credit-a	137.4	**56.2**	59.7	**51.5**
Credit-g	333.6	**151.1**	190.4	**139.4**
Diabetes	**49.4**	90.2	**43.4**	80.8
Glass	**49**	55.8	**46.2**	48
Heart-c	69.6	**33**	36	**26.5**
Heart-h	78.2	**25.8**	**15.7**	19
Hepatitis	29.4	**16.8**	**13.8**	15.6
Allhyper	63.7	**46.8**	34	**28.2**
Iris	**8.6**	8.8	**8**	8.4
Labor	14.1	**7.8**	7.8	**5.3**
Letter	**2581.8**	2694	**2412.4**	2458.4
Segment	**86.4**	97.2	**81.8**	94.8
Sick	66.1	**37**	48.8	**37**
Sonar	27.2	**25**	27.2	**25**
Vehicle	**151**	171	**134.8**	163.2
Wave	49.2	**46.2**	48.4	**45**

Prediction accuracy: Before pruning, the new C4.5 outperforms the original C4.5R8 in prediction accuracy (the new C4.5 wins 11 cases, while C4.5R8 wins 9 cases). After pruning, the new C4.5 is comparable with the original C4.5R8 (the new C4.5 wins 10 cases, while C4.5R8 wins 9 cases). The new C4.5 algorithm seems more successful in the dataset labor in which the algorithm achieves a 15.7% prediction error rate, while the original algorithm has a 24.7% error rate.

The original C4.5R8 algorithm performs best using the pruning procedure, while the new C4.5 algorithm performs best without using the pruning procedure. The difference between the sum 333.9% of the results of experiments on 20 datasets in the third column and the sum 337.9% in the fourth column is 4%. This means that when we compare their best, the new C4.5 algorithm is comparable with the original C4.5R8 algorithm in prediction accuracy. Note that the prediction accuracy of the original C4.5R8 algorithm is not statistically significantly different from POL, whose prediction accuracy is the best among a group of 33 classification algorithms (Lim et al., 2000).

Size of tree: Before pruning, in 12 datasets, there are less leaves in trees created by the new C4.5 than those created by the original C4.5R8. After pruning, in 10 datasets, these are less leaves in trees created by the new C4.5 than those created by the original C4.5R8. This means the new C4.5 is better than the original C4.5R8 in size of tree when we do not use the pruning procedure, while it is comparable with the original C4.5R8 algorithm in size of tree when we use the pruning procedure.

Comparison With γ In Attribute Selection

We compare Γ with γ in attribute section on the zoo dataset, which is obtained from the UCI machine learning repository. The zoo dataset has 101 cases, 16 conditional attributes, and one decision attributes. All the attributes are discrete.

Let D be the set of the decision attributes. For a given number k, we select a subset C of conditional attributes such that $\Gamma(C, D)$ $(\gamma(C, D))$ is maximal among all possible subsets with k conditional attributes. Then we apply the selected subset of conditional attributes to C4.5R8 algorithm. The results are shown in the Table 11. The first column is the number

TABLE 11. Attribute selection by γ and Γ on the dataset "zoo."

k	C_1 by γ	$\gamma(C_1, D)$	O-default	O-g-s	C_2 by Γ	$\Gamma(C_2, D)$	O-default	O-g-s
1	{4}	0.41	39.5	39.5	{13}	0.60	**26.4**	**26.4**
2	{1,13}	0.65	15.7	14.7	{4,13}	0.83	**12.7**	**12.7**
3	{3,12,13}	0.76	13.7	11.9	{4,6,13}	0.92	**12.7**	**10.8**
4	{3,10,13,14}	0.96	17.7	10.9	{4,6,8,13}	0.98	**9.8**	**7.9**
5	{4,6,8,12,13}	1.0	5.9	5.9	{4,6,8,12,13}	1.0	5.9	5.9
16	T	1.0	6.6	6.9	T	1.0	6.6	6.9

of selected attributes. When the number of selected attributes is greater than five, there will no difference for attribute selection between Γ and γ, and we omit the cases when $5 < k < 16$. This can be explained by Theorem 2 and Theorem 5. By Theorem 5, when $C \subseteq C'$, $\Gamma(C, D) \leq \Gamma(C', D)$ because of $IND(C) \supseteq IND(C')$, so the maximum $\Gamma(C, D)$ is equal to one when C ranges over all the subsets with $k(k > 5)$ conditional attributes because the maximum $\Gamma(C, D)$ is equal to one when C ranges over all the subsets with $k = 5$ conditional attributes; by Theorem 2, both γ and Γ is equal to one if one of them is equal to one, and therefore, the maximal value of Γ and the maximal value of γ is equal when $k > 5$. In the seventh row, T denotes the whole conditional attributes, i.e., $T = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}$. The second column and the sixth column are the set of conditional attribute selected by γ and Γ, respectively. The third column and the seventh column are the γ value and the Γ value, respectively, corresponding to the selected attributes.

The fourth column and the eighth column are the mean error rates of the 10-fold cross-validations on the selected attributes by γ and Γ, respectively, and the results are obtained by the C4.5R8 with default option; the fifth column and the ninth column are obtained by the C4.5R8 with the option -g -s.

The experimental results show that both Γ and γ are efficient in attribute selection on dataset zoo. By either of Γ and γ, we can find the same best attribute set $\{4, 6, 8, 12, 13\}$ on which the C4.5R8 performs better in accuracy than employing all the conditional attributes. However, when the number of selected attributes is less than five, the C4.5R8 performs better in accuracy on the attributes selected by Γ than on those selected by γ.

Conclusion and Future Work

We give three different forms of the generalized dependency degree in terms of equivalence relations, minimal rule, and arithmetic operation, respectively.

Among our three different forms of the generalized dependency degree, the first form of the measure (in terms of equivalence relations) is the most important. Besides its simplicity, the first form is flexible, and it can therefore be extended not only to an equivalence relation but also to an arbitrary relation. The first form (in terms of equivalence relations) and the second form (in terms of minimal rules) share the advantage of being easily understood, while the third form of the measure (in terms of arithmetic operations) is computationally efficient. So these three forms of the measure are suited to different situations. When we want to extend the measure to a more complicated data structure (such as partial order relation, totally order relation or others) than an equivalence relation, or when we want to find some properties of this measure, we can employ the first two forms of the measure. When we use it in a computing situation, the third form of the measure may be the best choice. In fact, in this article, we determine its properties using the first two forms, and then in the experiments, we use the third form.

The generalized dependency degree Γ has some properties, such as the Partial Order Preserving Property and the Anti-Partial Order Preserving Property. Besides, its value is between zero and one. Therefore, it can serve as an index to measure how much decision attributes depend on conditional attributes. The experimental study shows that the generalized dependency degree is an informative measure in the decision tree and the attribute selection.

Our experiments only show some possible applications of the generalized dependency degree in the field of decision trees and in the field of attribute selection, and there is much future work left. In the future, we will investigate Γ in other fields where the conditional entropy is applicable. For the application in decision trees, the original pruning method is designed for the original C4.5R8, and we will also design a new pruning method that is suitable for the new C4.5.

Acknowledgment

This work is fully supported by two grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4150/07E and Project No. CUHK4235/04E). We thank J. R. Quinlan for his useful comments and his source code of the C4.5R8 algorithm, without which the experiments in this work would have taken much longer. The UCI Data Repository owes its existence to David Aha and Patrick Murphy. The breast cancer data (breast-w) was provided to the Repository by Dr. William H. Wolberg. We also want to express our thanks to the anonymous reviewers for their valuable comments.

References

Cover, T. (1991). Elements of information theory. New York: Wiley.

Dalkilic, M., & Robertson, E. (2000). Information dependencies. In Proceedings of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principals of Database Systems (pp. 245–253). Dallas, ACM.

Gediga, G., & Düntsch, I. (2001). Rough approximation quality revisited. Artificial Intelligence, 132, 219–234.

Giannella, C., & Robertson, E. (2004). On approximation measures for functional dependencies. Information Systems, 29(6), 483–507.

Gunther, G., & Ivo, D. (2000). Statistical techniques for rough set data analysis in rough sets: New developments (pp. 545–565). Heidelberg/Berlin, Physica Verlag/Springer-Verlag.

Hassanien, A. (2004). Rough set approach for attribute reduction and rule generation: A case of patients with suspected breast cancer. Journal of the American Society for Information Science and Technology, 55(11), 954–962.

Ivo, D., & Gunther, G. (1997). Statistical evaluation of rough set dependency analysis. International Journal of Human-Computer Studies, 46, 589–604.

Kryszkiewicz, M. (1998). Rough set approach to incomplete information systems. Information Sciences, 112, 39–49.

Kryszkiewicz, M. (1999). Rules in incomplete information systems. Information Sciences, 113, 271–292.

Lee, T. (1987). An information-theoretic analysis of relational databases part i: data dependencies and information metric. IEEE Transactions on Software Engineering, 13(10), 1049–1061.

Leung, Y., & Li, D. (2003). Maximal consistent block technique for rule acquisition in incomplete information systems. Information Sciences, 153, 85–106.

DOI: 10.1002/asi

Lim, T.-S., Loh, W.-Y., & Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning, 40, 203–228.

Lingras, P., & Yao, Y. (1998). Data mining using extensions of the rough set model. Journal of the American Society for Information Science, 49(5), 415–422.

Malvestuto, F. (1986). Statistical treatment of the information content of a database. Information Systems, 11(3), 211–223.

Nambiar, K. (1980). Some analytic tools for the design of relational database systems. In Proceedings of the Sixth International Conference on Very Large Databases (pp. 417–428). Montreal: IEEE Computer Society.

Pawlak, Z. (1999). Rough classification. International Journal of Human-Computer Studies, 51, 369–383.

Pawlak, Z. (2002a). Rough sets and intelligent data analysis. Information Sciences, 147, 1–12.

Pawlak, Z. (2002b). Rough sets, decision algorithms and Bayes' theorem. European Journal of Operational Research, 136, 181–189.

Quinlan, J.R. (1993). C4.5: Programs for machine learning. San Mateo: Morgan Kaufmann.

Quinlan, J.R. (1996). Improved use of continuous attributes in c4.5. Journal of Artificial Intelligence Research, 4, 77–90.

Yao, Y. Y., Li, X., Lin, T. Y., & Liu, Q. (1994). Representation and classification of rough set models. Soft Computing: Proceedings of the Third International Workshop on Rough Sets and Soft Computing (RSSC'94), San Jose, CA, November 10–12, T. Y. Lin and A. M.Wildberger (Eds.), San Diego, CA, the Society for Computer Simulation (pp. 44–47).

Yao, Y. (2003a). Entropy measures, maximum entropy and emerging applications, chapter information-theoretic measures for knowledge discovery and data mining (pp. 115–136). Springer: Berlin.

Yao, Y. (2003b). Probabilistic approaches to rough sets. Expert Systems, 20(5), 287–297.

Yeung, R.W. (2002). A first course in information theory. New York: Kluwer Academic Publishers.

Zimmerman, H. (2001). Fuzzy set theory and its application. Boston: Kluwer Academic Publishers.

DOI: 10.1002/asi