The Chinese University of Hong Kong
Department of Computer Science and Engineering

Ph.D. – Term Paper

Title:	Data Processing with Missing Information
Name:	Yang Haixuan
Student I.D.:	03499020
Contact Tel. No.:	3163-4257
Email A/C:	hxyang@cse.cuhk.edu.hk
Supervisor:	Prof. Irwin King & Prof. Michael R. Lyu
Markers:	Prof. Christopher C. Yang (SEEM) & Prof. Evangeline F.Y. Young
Mode of Study:	Full-time
Submission Date:	November 30, 2004
Term:	2
Fields:	
Presentation Date:	December 13, 2004(Monday)
Time:	10:30–12:00 am
Venue:	Rm. 101A, Ho Sin-Hang Engineering Building

Data Processing with Missing Information

Abstract

We handle incomplete information in two aspects: One is about the web structure; the other is about information system with missing values.

In the area of link analysis, the celebrated PageRank algorithm has proved to be a very effective paradigm for ranking results of web search algorithms. However, as the web continues to grow, it becomes more impossible for one search engine to crawl all the web page, as a result, the final page ranks computed by PageRank are only based on a subset of the whole web, which is found or visited by a crawler. This results in inaccuracy because of its incomplete information about the web structure. Can we find a way to get as much as information about the web structure based on the limited subset of the whole web so that the inaccuracy of the PageRank can be avoided as much as possible? We try to solve this kind of problem by proposing a new method for ranking pages. The main idea is that during the process of crawling, the information for unknown links can be used for link analysis, more specifically, we can use known information about links to predict the number of inlinks for each page that have already been found, and thus we can predict the information about unknown links and the link structure based on such information. In other words, the web pages already visited or found along with the known links form a known directed graph, whose structure is somewhat certain, and a random graph is formed based on the predicted number of inlinks, whose structure is somewhat uncertain, we then apply the combination of the known graph and the random graph to the PageRank algorithm to get the final page ranks. Experiments show this algorithm achieves encouraging results both in speed and in accuracy.

In the area of information system, the dependency degree, γ is a traditional measure in Rough Set Theory to measure the dependency between the conditional attributes and the decision attributes. However, γ does not express the dependency accurately. More specifically, in extreme cases when there is no deterministic rule between the conditional attributes and the decision attributes, the dependency degree γ becomes zero, but there may exist some kinds of dependency between the conditional attributes and the decision attributes. To avoid such inaccuracy we introduce a generalized dependency degree, γ' between two sets of attributes which counts both deterministic rules and indeterministic rules while γ only counts deterministic rules. Therefore γ' is a generalization of γ. We first give the definition of generalized dependency degree in terms of equivalence relation, then interpret it

in terms of minimal rule, and further find its connection with conditional entropy used in literature of decision tree. In order to obtain a deeper understanding of the generalized dependency degree, we investigate its various properties. Furthermore we can extend its formulation in incomplete information systems based on probabilistic distribution for missing values. After the theoretical study on this measure, encouraged by its simplicity and its good properties, we turn to empirical study by replacing with it the measure used in the well-known C4.5 algorithm such that a new C4.5 algorithm is formed. The original C4.5 algorithm needs the MDL principle and the pruning procedure to achieve better prediction accuracy, while the new C4.5 algorithm discards both. Experiments show that the speed of the new C4.5 algorithm is improved greatly, while the prediction accuracy of the new C4.5 algorithm is a little better than the original C4.5 algorithm.

Contents

Chapter 1

Introduction

Information that we do not know is called missing information. When a crawler crawls the web, it visits one page after another. If the crawler visits all the pages on the web and thus finds all the link information for the web page, then there is no missing link information. However, in most cases, it is impossible for a crawler to visit all the web pages, consequently, the missing link information arises. In a database, if some of the values for some attributes are unknown, then this database is an information system with missing information.

In our work, we do not try to guess exactly what the missing information is. What we do is to process them so that the original algorithms can be changed to new algorithms that perform better. In case of missing information for links, we handle the missing information by predicting web structure, thereby the original PageRank algorithm is changed to Predictive ranking algorithm; in case of missing information for database, the C4.5 algorithm is changed to new C4.5 algorithm.

Because our ways of handling missing information are oriented to algorithms, and because the two algorithms we care belong to different areas, the details for them are very different, and we have to introduce different methods for handling missing information in different chapters. However, the common things of the two chapters are that we estimate the missing information in a similar statistic way, i.e., using the sample to estimate the probability density.

In the area of link analysis, PageRank in [22] gives the relative importance of web page based on the link structure of the web. The intuition behind PageRank is that it uses information which is external to the web pages themselves–their in-links, and that in-links from "important" pages are more significant than in-links from average pages.

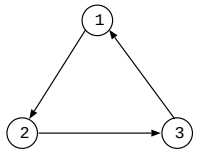
More formally presented in [6], the web is modelled by a directed graph $G = (V, E)$ in the PageRank algorithms, and the rank or "importance" x_i of each for the n pages $i \in V$ is defined recursively in terms of pages

which point to it:

$$x_i = \sum_{(j,i)\in E} a_{ij} x_j \tag{1.1}$$

where a_{ij} is assumed to be $1/d_j$, d_j is the out-degree of page j. Or in matrix terms, $x = Ax$.

This form has three problems. One is that if the matrix A is not a positive matrix (although non-negative), and therefore can not guarantee the convergence of the power iterative method for the linear system. For example, in the graph, the corresponding matrix is



$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

If we use the power iterative method to solve the above page rank problem, then we will suffer the problem of convergence unless the entire initial values of x_i $(i = 1, 2, 3)$ take the value of $1/3$, which usually can not be found in practice.

Another problem is that in practice, the users do not follow the link all the time, and when they become bored, they may jump to some other page, and therefore the model 1.1 can not model the reality accurately.

In [22], these two problems are handled by one technique, i.e., introducing the concept of "random jump" or "teleportation", and therefore avoid the problem of inaccuracy and the problem of convergence (the modified matrix is a positive stochastic matrix, and so one is its largest absolute eignvalue and no other eignvalue whose absolute value is equal to 1, which is guaranteed by the famous Perron Theorem [19]).

Theorem 1: Perrons theorem. The eigenvalue of largest absolute value of a positive (square) matrix A is both simple and positive and belongs to a positive eigenvector. All other eigenvalues are smaller in absolute value.

This technique is presented formally in model 1.

Model 1 : The matrix form of the equation 1.1 is changed to

$$x = [(1-\alpha) f e^T + \alpha A] x, \tag{1.2}$$

where the parameter α is the probability of following the actual link from a page, $(1-\alpha)$ is the probability of taking a "random jump", f is a stochastic vector (i.e. $e^T f = 1$). It has become a standard to use the value $\alpha = 0.85$ and e is the vector of all ones.

The third problem in Equation 1.1 (also in Equation 1.2) is that the matrix A is not column stochastic unless every node has at least one outlink. This kind of problem is called dangling node problem [6]. Pages that either have no outlink or for which no outlink is known are called dangling nodes. In the following, we analyze the reasons that cause the dangling nodes, and classify dangling nodes into 3 classes according to these reasons.

Pages unvisited or not visited successfully are dangling nodes, besides, some pages visited successfully may be dangling nodes, for example, PostScript, Pdf, and TXT files on the web are dangling nodes because such kinds of files have no outlinks. In more details, we explain three reasons that cause dangling pages.

That we can not visit all the page is one reason to produce dangling pages. Pages that are not visited are called dangling pages of class 1. In this paper, we focus on this kind of dangling nodes.

The second reason is that there exit many pages that we try to visit but can not visit them successfully. These pages may exist some time ago, but now is damaged or is in maintenance, or they are protected by a robot.txt. Pages that have been tried but not visited successfully are called dangling pages of class 2.

The third reason is that there are many files on the web that have not hyperlink structure. Pages which have been visited successfully and from which no outlink is found are called dangling pages of class 3.

In [22], the authors suggested simply removing the pages that have no outlink and the links that point to them. After doing so it was suggested that they can be "added back in" without significantly affecting the results. However the situation is changed now and the dangling nodes problem has to be handled more accurately and directly.

On one hands, from model 1, we can see that the PageRank algorithm depends on the web structure. When we only know part of the web structure, is PageRank algorithm still accurate? As the web continues to grow, the visited fraction of the whole web page by a crawler becomes smaller and smaller, unfortunately this becomes a fact. It is hard to sample the entire web. In [22], the authors reported that they have 51 million URLs not downloaded yet when they have 24 million pages downloaded. In [10], dynamic pages are estimated to be 100 times more than static pages, and in [6], the authors point out in their experiment that the number of uncrawled pages still far exceeds the number of crawled pages and that there are an essentially infinite number of URLs which is estimated to be at least 64^{2000}. The database-driven pages may produce many pages at a short time. These experimental results and theoretical analysis mean that in reality, unvisited pages are so many that we have to face them.

On the other hands, some dangling pages worthy of ranking because they contain important information. Moreover, including dangling nodes in the overall ranking may have significant effect on the ranks of non-dangling

pages, this will be shown in the next section.

We handle different class of dangling nodes in different way. For dangling nodes of class 2 and 3, we treat it in traditional way, however we handle dangling nodes of class 1 by predicting the link information about them. Dangling nodes of class 1 exits only because the crawler has not visited them at current time, and they contain some information that we can not know directly. These dangling nodes cause incomplete information about the web, which can fortunately be partly inferred from the known information. Dangling nodes of class 1 become the focus of this paper.

Different from dangling nodes of class 1 that have been found but not visited yet, nodes, which have been not found by the crawler, cause more serious incomplete information problem, however, we have to ignore this kind of nodes because we can do nothing about them.

Our Predictive Ranking Model based on the above consideration can be found in Chapter 3.

In the area of information system with missing values, we handle the missing information more systematically, first we introduce a generalized dependency degree based on a well-known dependency degree in Rough Set Theory, then we extend it to incomplete information system, and finally we change the C4.5 algorithm to new C4.5 algorithm, which can be found in Chapter 4.

According to Rough Set Theory an information system is a four-tuple $S = (U, A, V, f)$, where U represents the universe of objects, A represents the set of attributes or features, V represents the set of possible attribute or feature values, V_a, the domain of the attribute a, is the set of all possible value of attribute a and f is the information function which maps an given object and a given attribute to a value, i.e.,

$$f : U \times A \rightarrow V.$$

By $a(x)$ we denote the value of $f(x, a)$. An information system is represented by an attribute-value table in which rows are labelled by objects of the universe and columns by the attributes. Let P be a subset of A, that is, P is a subset of attributes. The P-indiscernibility relation, denoted by $IND(P)$, defined as

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

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 P-class. For $x \in X$, let $P(x)$ denote the P-class containing x. For any class X where $X \subseteq U$, and for any subset of attributes P, the P-lower approximation of X, denoted by $\underline{P}(X)$, is defined as

$$\underline{P}(X) = \cup\{Y \in U/IND(P) \mid Y \subseteq X\}.$$

Let C and D be two subsets of A, the dependency degree $\gamma(C, D)$ is defined in [23] as

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

$\gamma(C, D)$ expresses the percentage of objects which can be correctly classified into D-class by employing attribute C. It is also the relative number of elements of U which can be described by deterministic rules since each C-class contained in a D-class corresponds to a deterministic rule (and vice versa). Because of this, Gediga, et al [7] consider γ as a traditional measure in Rough Set Theory to evaluate the classification success of attributes in term of numerical evaluation of the dependency properties generated by these attributes. For example, Hassanien [11] uses γ to generate rules in a case study.

However, $\gamma(C, D)$ does not accurately express the dependency among different attributes in any case. The problem of inaccurateness of $\gamma(C, D)$ can be seen more clearly in the extreme cases when there is no deterministic rule. Specifically, when there is no deterministic rule between C and D, the dependency degree $\gamma(C, D)$ will be equal to zero, whereas this may not mean that there is no dependency between C and D. For example, in Table 1.1 where a, b, c, and d represent *headache, muscle pain, body temperature* and *influenza*, respectively, it is easy to calculate that $\gamma(C, D) = 0$ when $C = \{a\}, D = \{d\}$. This happens because 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. This seems contradictory to our intuition.

Here, we propose the generalized dependency degree $\gamma'(C, D)$. Because every rule whose confidence is not equal to zero reflects to a certain degree the relation between the conditional attributes and the decision attributes, theoretically we should not ignore any rule whose confidence is not equal to zero. Based on this consideration, we count in the proposed $\gamma'(C, D)$ all the minimal rules whose confidence are not equal to zero and hence all the rules whose confidence are not equal to zero (since every rule is a join of some minimal rules). The generalized dependency degree $\gamma'(C, D)$ is different from the γ-like statistics introduced by Gediga, et al [7], the idea of which is to count the number of error while $\gamma'(C, D)$ counts every object by a corresponding fraction (as we will explain later, it is equivalent to count all the minimal rules whose confidence are not equal to zero).

In literature of decision tree, Breiman, et al [2] recommends in their CART algorithm adopting the Gini index as an impurity measure and choosing the split that maximizes the decrease in impurity. The Gini index in fact is a special case of γ'.

We try to make a deeper understanding of the generalized dependency degree in Chapter 4, and justify it both theoretically and empirically. Theoretically, we give its various forms and develop its various properties, and extend the definition of the generalized dependency degree to incomplete information system. Empirically, we

	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

Table 1.1. Influenza Data

compare this measure with the conditional entropy by replacing the conditional entropy with the generalized dependency degree in well-known C4.5 classification algorithm.

Chapter 2

Related Work

We first introduce the related work in the area of link analysis, then we show the related work in information system.

For convenience, $\mathbf{1}$ denotes the matrix of all ones, e denotes the vector of all ones. In the original PageRank paper [22] the authors suggested simply removing the links to dangling pages from the graph, calculating the PageRank on the remaining pages, and dangling nodes problem has received relatively less attention in the past.

In [1], an absorbing model was suggested. This model can handle dangling nodes by modifying the original graph. Specifically speaking, It adds additional nodes (called clones), adds links from all the original nodes to their clones on the web, and adds links from all the clones to themselves. As a result, the modified graph has no dangling node and so it is robust against dangling nodes.

In [12], pages whose out-degree is zero are handled by adding jump to a randomly selected page with probability 1 from every dangling node, and then by adding teleportation. More formally, the model 1 is modified as

Model 2:

$$x = [(1-\alpha)E + \alpha P']x, \tag{2.1}$$

where $E = fe^T,\ P' = A + fd^T,\ f = e/n$, and d denotes the $n-$dimensional column vector identifying the dangling nodes:

$$d_i = \begin{cases} 1 & \text{if } i \text{ is a dangling node,} \\ 0 & \text{otherwise.} \end{cases}$$

f is referred as the personalization vector, it models the behavior of users when they get bored in following the link and decide to jump randomly.

Further, Kamavar et. al. [12]speed up the PageRank algorithm by exploiting the block structure of the web.

In [6], dangling pages are handled in similar way, but more computationally efficient though scarifying some kind of accuracy. We reinterpret the model formally as follows.

Model 3:

$$\begin{aligned}\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \alpha C + (1-\alpha)/m \cdot \mathbf{1} & 1/m\mathbf{1} \\ \alpha D & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \\ &= (\alpha A + (1-\alpha)B) \begin{pmatrix} x \\ y \end{pmatrix}\end{aligned} \tag{2.2}$$

where $A = \begin{pmatrix} C & 1/m\mathbf{1} \\ D & 0 \end{pmatrix}$, $B = \begin{pmatrix} 1/m \cdot \mathbf{1} & 1/m\mathbf{1} \\ 0 & 0 \end{pmatrix}$, m is number of nodes that have been crawled successfully,n is the number of nodes that have been found by the crawler, $C = (c_{ij})$, $D = (d_{ij})$ and if d_j is the out-degree of node j,

$$c_{ij} = \begin{cases} d_j^{-1} & \text{if there is a link node } i \text{ node } j, \\ 0 & \text{otherwise.} \end{cases}$$

$$d_{ij} = \begin{cases} d_j^{-1} & \text{if there is a link node } i \text{ node } j, \\ 0 & \text{otherwise.} \end{cases}$$

Respectively by C and D we also denote the set of all nodes that have been crawled successfully and the set of remaining nodes.

In this model, the matrix A models the users' behavior in case of following the actual links and the unknown links from dangling nodes to visited nodes. The matrix B models the users' teleportation. Then the linear convex combination of the matrix A and the matrix B models the total behaviors of the users.

By adding a virtual node $n+1$, the Eq. 2.2 is equivalent to the following

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \alpha C & O & e/m \\ \alpha D & O & 0 \\ (1-\alpha)e^T & e^T & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{2.3}$$

which can be found in [6]. Exploiting this structure, the authors developed the following reduced eigen-system:

$$\begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} \alpha C & e/m \\ (1-\alpha)e^T + \alpha e^T D & 0 \end{pmatrix} \begin{pmatrix} x \\ z \end{pmatrix}$$

After solving the reduced eigen-system iteratively, the vector y can be calculated in one step:

$$y = \alpha D x.$$

While this form of linear equation can be calculated efficiently by exploiting the special structure of the above matrix and the computation complexity is a very important thing, if not most important, in the case of extremely large web, it is not as accurate as the model 2.

From Eq 2.2, we can see what the problem is. For our convenience, we denote the matrix $\begin{pmatrix} C & 1/m\mathbf{1} \\ D & 0 \end{pmatrix}$ as $\begin{pmatrix} X & M \\ Y & N \end{pmatrix}$, the link information about X and Y is already known because the crawler has visited all the nodes in C and therefore all the link from the nodes in C to nodes in C and D have been known by the crawler.

But the information about links from the nodes in D to nodes in C and D is unknown by the crawler because the nodes in D have not been visited yet or have not been visited successfully. Hidden in the matrix $\begin{pmatrix} C & 1/m\mathbf{1} \\ D & 0 \end{pmatrix}$, there is an assumption, in which users will jump randomly and uniformly from every node in D only to nodes in C , and therefore $M = 1/m\mathbf{1}$. This assumption can be improved to be more accurate. In reality, users may jump from nodes in D to nodes in D, and thus the assumption that all the elements in the right-bottom part of the matrix are zero is problematic, and the assumption about the right-top part of the matrix need to be adjusted accordingly. In our model, we assume that users will jump randomly but not uniformly from every node in D to both nodes in C and nodes in D.

Our model is different from the absorbing model in that our model try to predict the unknown link information and therefore handle the dangling node robustly and accurately while the absorbing model is new paradigm for ranking. So the ranking values derived from these two models are not comparable.

Our model is different from model 2 in that we get the information about the unknown part of the matrix by prediction while the model 2 assume the uniform distribution about the unknown part of the matrix. The authors in [12] also suggest re-defining the vector f as non-uniform distribution, however, they only consider the vector f as the personalization factor, which is a subjective factor, and from which the PageRank vector can be biased to prefer certain kinds of pages.

Our model is different from model 3 in two folds:

1. The users will not jump uniformly from every node in D to other nodes.

2. The users will jump from every node in D not only to nodes in C but also to nodes in D.

The authors in [6] further discuss the “link rot” problem and suggest new methods of ranking motivated by the

hierarchical structure of the web. Although we can combine our model with the technique used in solve these kinds of problem, we do not focus them in this paper.

Other related work is the Page Popularity Evolution Model in [3, 4], in which the popularity of a page evolves with the time; it is also a kind of prediction, which looks outside the web structure while our model looks inside the web structure at current time.

A Simple Example

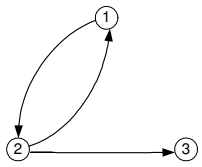


Figure 2.1. A case in which considering dangling node will have significant effect on the ranks of non-dangling nodes

We consider a case in which dangling nodes are so significant that including them in the overall ranking may not only change the rank value of non-dangling nodes but also change the order of the non-dangling nodes.

In the example of figure 2, there are three pages, with one of them being a dangling node with a link from page 2. If we compute pagerank by the model 2, and let $\alpha = 0.85$, the matrix in the model 2 is

$$\begin{pmatrix} 0.05 & 0.475 & 1/3 \\ 0.9 & 0.05 & 1/3 \\ 0.05 & 0.475 & 1/3 \end{pmatrix}$$

By power iteration, the RageRank scores are $(x_1, x_2, x_3) = (0.3032, 0.3936, 0.3032)$. So in model 2, rank for node 2 is much higher than node 1. If we simply remove the dangling node 3, then by Equation 1.1, the PageRAnk scores for nodes 1 and 2 are $(x_1, x_2) = (0.5, 0.5)$, in which the rank for node 1 is same as that for node 2. From this example, we can see that whether we handle dangling nodes will not only change the rank value of the non-dangling nodes but also change their order.

In the area of information system, Nambiar [21], Malvestuto [20], and Lee [15] 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 [8]. The conditional entropy is a well known measure for dependency degree between attributes. Its formulation 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, et al [5] calls the conditional entropy as information dependency measure, denoted by $H_{C\rightarrow D}$. He develops a variety of arithmetic inequalities for this measure.

The formulation of entropy is

$$
H(D) = -\sum_{d}\Pr[d]\cdot\log_2(\Pr[d]).
$$

The third form of the generalized dependency degree $\gamma'(C,D)$, which can be found in Chapter 4, is written as

$$
\begin{aligned}
\gamma'(C,D) &= \sum_{c}\sum_{d}(\Pr[c]\cdot\Pr^2[d|c]) \\
&= \sum_{c}(\Pr[c]\cdot\sum_{d}\Pr^2[d|c]),
\end{aligned}
$$

whose form is similar to that of the conditional entropy. To compute $H(D|C)$, we need to compute $\log_2(\Pr[d|c])$ which is more time-consuming. While in $\gamma'(C,D)$, we do not need to compute such time-consuming logarithm function.

A variation of this probabilistic form is originally defined by Goodman, et al [9] in literature of statistics, later independently by Piatetsky-Shapiro [26] in literature of machine learning. Unfortunately, they do not focus on this measure itself. Goodman, et al [9] do not discuss it further after giving its normalized form while Piatetsky-Shapiro [26] focuses more on its expected value under randomization. Nor they conduct experiments to support this measure. Giannella, et al [8] interpret it as the probability of a correct guess in situation when the drawer is told the value of the conditional attributes, and compares empirically the numerical values of the normalized form of the generalized dependency degree with that of the normalized InD measure IFD, and find that the average of the difference between these two normalized measures tends to be close to zero (this does not mean the difference between these two measures themselves tends to be close to zero). Our theoretical work focuses on the generalized dependency degree itself, not the normalized form, and our empirical work focus on changing a well-known classification algorithm C4.5 by replacing the conditional entropy with the generalized dependency degree.

Breiman, et al [2] recommends in their CART algorithm adopting the Gini index as an impurity measure and choosing the split that maximizes the decrease in impurity. The Gini index in fact is $\gamma'(U \times U, IND(D))$ which is a special case of the generalized dependency degree, and the corresponding decrease in impurity is actually $\gamma'(\{a\}, D) - \gamma'(U \times U, IND(D))$ which is ensured to be larger than zero by Theorem 5 and is called as dependency gain in next section. This fact is also proved, but no further properties are given in [2].

None of the above work formulate the generalized dependency degree in terms of equivalence relation or in terms of minimal rule, which can bring us a better understanding for this measure as we have seen in the previous sections. By formulating it in terms of equivalence relation, we can find its connection to the dependency degree frequently used in Rough Set Theory, by reformulating it we find its connection to the minimal rules, and by interpreting it in terms of probability distribution, we find its connection to the measure that is presented but not fully investigated in the above work.

We give three different forms of the generalized dependency degree, in terms of equivalence relation, minimal rule, and probability respectively. These three different forms can be used in different situations. When we want to extend the measure to more complicated data structure than equivalence relation, or when we want to find some properties about this measure, we can resort to the first two forms of the measure. When we use it in the computing situation, the third form of the measure may be the best choice. In fact, in this paper, we guess its various properties by the first two forms, and in the experiments, we use the third form.

If an information system has some missing values, we call this information system as incomplete information system. For example, there are three missing value in the Table 2.1 with "*" in the space.

	a	b	c	d
e_1	Y	Y	Normal(0)	N
e_2	Y	*	High(1)	Y
e_3	Y	Y	*	Y
e_4	N	*	Normal(0)	N
e_5	N	N	High(1)	N
e_6	N	Y	Very High(2)	Y
e_7	Y	N	High(1)	Y

Table 2.1. Influenza Data

The problem of rule generation from incomplete information systems is considered in literature. The simplest method is to remove examples with unknown values. Replacing every value by set all possible values is another

method [18]. Introducing the similarity relation and completion of an incomplete information system is a more accurate way to handle missing values [13, 14, 16]. To extend the definition of generalized dependency degree to the case of incomplete information systems, we handle missing values by replacing them with their probabilistic distribution at first, then extending the definition of confidence and strength of a rule to incomplete information system.

Chapter 3

Handle Missing Information for Link Analysis

3.1 Predictive Ranking Model

The intuition behind our model is that the link structure in M and N can be estimated from the known information. In general, it is difficult to estimate such link structure accurately; however, some elementary estimation is possible. In this paper, we only estimate the in-degree of each node in the set $C \cup D$, and thus some information about the link structure in M and N can be inferred statistically. The dogma in the predictive ranking is that: the more we know the structure of the web, the more accurate we can infer about the web.

We formulate our model as follows.

1. Suppose that all the nodes V of the graph ($n = |V|$) can be partitioned into three subsets: C, D^1, and D^2, where C ($|C| = m$) denotes the subset of all nodes that have been crawled successfully and have at least one out-link; D^1 ($|D^1| = m_1$) denotes the subset of all dangling nodes of class 3, i.e., the set of those nodes that the crawler has visited successfully but from which no outlink is found; D^2 ($|D^2| = n - m - m_1$) denotes the set of all dangling nodes of class 1, i.e., the set of all nodes that have not been visited but have been found by the crawler through other visited nodes. Dangling nodes of class 2 are ignored here.

2. For every node v_i ($i = 1, 2, ..., m$) in C, the real out-degree $d^+(v_i)$ ($i = 1, 2, ..., m$) has been known since the crawler has found all its outlinks, but the real in-degree $d^-(v_i)$ ($i = 1, 2, ..., m$) is unknown since the crawler has not visited all the nodes in V and there maybe unknown links from the nodes in D^2 to the node v_i.

3. For every node v_i ($i = m + 1, m + 2, ..., m + m_1$) in D^1, since v_i is a dangling node of class 3, the real out-degree $d^+(v_i) = 0$. Again the real in-degree $d^-(v_i)$ ($i = m + 1, 2, ..., m + m_1$) is unknown.

4. For every node v_i ($i = m + m_1 + 1, m + 2, ..., n$) in D^2, neither the real out-degree $d^+(v_i)$ ($i = m + 1, m + 2, ..., n$) nor the real in-degree $d^-(v_i)$ ($i = m + 1, m + 2, ..., n$) has been known since the crawler has not crawled

it yet.

5. We predict the real in-degree $d^-(v_i)$ ($i = 1, 2, ..., n$) by the number of found links $fd^-(v_i)$ ($i = 1, 2, ..., n$) from visited nodes to the node v_i. With the breadth-first crawling method, we assume that the number of found links $fd^-(v_i)$ ($i = 1, 2, ..., n$) from visited nodes to the node v_i is proportional to the real number of links from all nodes in V to the node v_i, and further we assume that

$$d^-(v_i) \approx \frac{n}{(m+m_1)} \cdot fd^-(v_i)(i = 1, 2, \ldots, n)$$

This assumption is meaningful. Although the crawler crawls the web from a given web site to other sites in a definite way, but its ability of finding new link to a given node v_i depends on the density of these links. The density of these links to the node v_i is equal to $\frac{d^-(v_i)}{n}$.

The crawler has found $fd^-(v_i)$ such kind of links when it has crawled m nodes, and we consider $\frac{fd^-(v_i)}{(m+m_1)}$ as an approximate estimate of the density of these links. Following this, the above approximate equality holds.

6. With the approximate in-degree $d^-(v_i)$, we can re-arrange the matrix. All the found links ($fd^-(v_i)$) are from the nodes in C, and the remaining links ($d^-(v_i) - fd^-(v_i)$) are from the nodes in D^2 (it is impossible that some of these links are from the nodes in D^1). Without any prior information about the distribution of these remaining links, we have to assume that they are distributed uniformly from the nodes in D^2 to the node v_i, i.e., these remaining links are shared by all the nodes in D^2. So the column normalized matrix A modelling the uses' behavior of following the actual links and estimated actual links is $A = \begin{pmatrix} C & P & M \\ D & Q & N \end{pmatrix}$, where $\begin{pmatrix} C \\ D \end{pmatrix}$ is defined as in Model 3, and it is used to model the known link structure from C to V, $\begin{pmatrix} P \\ Q \end{pmatrix}$ will be defined later, and $\begin{pmatrix} M \\ N \end{pmatrix}$ is used to model the link structure from D^2 to V, and it is defined as follows:

$$\begin{pmatrix} M \\ N \end{pmatrix} = \begin{pmatrix} l_1 & 0 & 0 & 0 \\ 0 & l_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & l_n \end{pmatrix} \mathbf{1}_{n \times (n-m-m_1)},$$

$l_i = \frac{d^-(v_i) - fd^-(v_i)}{(n-m-m_1)\Sigma}$, ($i = 1, 2, \ldots, n$), $n - m - m_1$ means that the remaining inlinks $d^-(v_i) - fd^-(v_i)$ are shared uniformly by all nodes in D^2 and $\Sigma = \sum_{i=1}^{n} d^-(v_i) - fd^-(v_i)$ is multiplied to the denominator to make the

matrix to be stochastic.

7. When we want to model the users' teleportation, we assume that the users will jump to node v_i ($i = 1, 2, ..., m$) with a probability of f_i when they get bored in following the actual links. So the matrix modelling the teleportation is fe^T. We denote here $(f_1\ f_2\ \ldots\ f_n)^T$ by f. Previous suggestions include the choice of a uniform distribution among all pages, among a set of trusted "seed sites", uniformly among a set of all "top-level" pages of sites, or a personalized set of preferred pages.

8. When the user encounters a dangling node of class 3, there is no outlink that the user can follow. In this case, we assume that the same kind of teleportation as in 6 will happen, and so the matrix $\begin{pmatrix} P \\ Q \end{pmatrix}$ in 5 is used to model the link structure from D^1 to V and it is assumed to be

$$\begin{pmatrix} P \\ Q \end{pmatrix} = \begin{pmatrix} f_1 & 0 & 0 & 0 \\ 0 & f_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & f_n \end{pmatrix} \mathbf{1}_{n \times m_1}.$$

9. We further assume that α is the probability of following an actual out-link from a page, $1-\alpha$ is the probability of taking a "random jump" rather than following a link. Then the rank x_i of ith page should satisfy

$$x = [(1-\alpha) f e^T + \alpha A] x, \tag{3.1}$$

Where x is the vector consisting of x_i. .

Because we use the predictable information to construct the link analysis model, we name it as predictive ranking.

3.2 Block Predictive Ranking Model

The predictive ranking model can be handled more accurate than we have done in the previous section if we see into the block structure of the web. We can assume that there is p blocks found in the web, and they are B_1, B_2, ..., B_p. These p blocks are disjoint with each other, and their union is $C \cup D^1 \cup D^2$. Let $C_j = C \cap B_j$ ($|C_j| = m_j$), $D_j^1 = D^1 \cap B_j$ ($|D_j^1| = m_j'$), $D_j^2 = D^2 \cap B_j$ ($|D_j^2| = n_j - m_j - m_j'$), $|B_j| = n_j$. We only need to change the fifth and sixth points of our previous model. The fifth point of our previous is changed to be

5'. We predict the real number $d_j^-(v_i)$ of links from nodes in block B_j to node v_i ($i = 1, 2, \ldots, n$) by the number of found links $fd_j^-(v_i)$ from nodes in C_j (the visited non-dangling nodes in block B_j) to the node v_i

($i = 1, 2, \ldots, n$). With the breadth-first crawling method, we assume that $fd_j^-(v_i)$ is proportional to $d_j^-(v_i)$ ($i = 1, 2, \ldots, n$), and further we assume that

$$d_j^-(v_i) \approx \frac{n_j}{(m_j + m'_j)} \cdot fd_j^-(v_i) \ (i = 1, 2, \ldots, n)$$

where $m_j = |B_j \cap C| = |C_j|$, $m'_j = |B_j \cap D^1| = |D_j^1|$, $n_j = |B_j|$.

The sixth point of our previous model can be changed to be

6'. With the approximate in-degree $d_j^-(v_i)$, we can re-arrange the matrix. All the found links ($fd_j^-(v_i)$) are from the nodes in C_j, and the remaining links ($d_j^-(v_i) - fd_j^-(v_i)$) are from the nodes in D_j. Without any prior information about the distribution of these remaining links, we have to assume that they are distributed uniformly from the nodes in D_j to the node v_i, i.e., these remaining links are shared by all the nodes in D_j. So the column normalized matrix A modelling the uses' behavior of following the actual links and modelling estimated actual links is

$$A = \begin{pmatrix} C & P & M_1 & M_2 & \cdots & M_p \\ D & Q & N_1 & N_2 & \cdots & N_p \end{pmatrix}$$

where $\begin{pmatrix} C & P \\ D & Q \end{pmatrix}$ has the same meaning as in the previous section, $\begin{pmatrix} M_j \\ N_j \end{pmatrix}$ is used to model the link structure from D_j^2 to V, and

$$\begin{pmatrix} M_j \\ N_j \end{pmatrix} = \begin{pmatrix} l_1^j & 0 & 0 & 0 \\ 0 & l_2^j & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & l_n^j \end{pmatrix} \mathbf{1}_{n \times (n_j - m_j - m'_j)},$$

where $l_i^j = \frac{d_j^-(v_i) - fd_j^-(v_i)}{(n_j - m_j - m'_j)\Sigma_j}$, ($i = 1, 2, \ldots, n$), $n_j - m_j - m'_j$ means that the remaining inlinks $d_j^-(v_i) - fd_j^-(v_i)$ are shared uniformly by all nodes in D_j^2 and $\Sigma_j = \sum_{i=1}^{n} d_j^-(v_i) - fd_j^-(v_i)$.

Note that all the link structure from D_j^1 ($j = 1, 2, ..., p$) to V are modelled together in the matrix $\begin{pmatrix} P \\ Q \end{pmatrix}$.

We can divide all the web pages into blocks by their top level domains (for example, edu), or by domains (for example, stanford.edu), or by the countries (for example, cn) . This block predictive ranking model should be more accurate than our previous one; however, we can not conduct experiment to support this model because of our limitation of resource.

Note that when $p = 1$, this model becomes our previous model.

Time t	1	2	3	4	5	6
Vnum[t]	7712	78662	109383	160019	252522	301707
Tnum[t]	18542	120970	157196	234701	355720	404728
Time t	7	8	9	10	11	
Vnum[t]	373579	411724	444974	471684	502610	
Tnum[t]	476961	515534	549162	576139	607170	

Table 3.1. Description of Data Sets Within Domain cuhk.edu.hk

3.3 Experimental Setup

Due to our limited network and storage resource we had to restrict our experiments to a relatively small subset of the web, the network restricted within the domain cuhk.edu.hk. Another reason we choose this subset of the network is that we can get the relatively complete information about all the pages in this subset so that the relatively accurate pages ranks can be calculated to make an easier comparison.

We also get a small subset of network by crawling outside the domain cuhk.edu.hk. We show these results in the following subsections.

3.3.1 Experiments Within Domain cuhk.edu.hk

Because the importance of a web page is an inherently subjective matter, it is difficult to measure whether a link analysis algorithm is better than another. Due to the success of the PageRank, we consider it as an ideal algorithm in the case that we have complete information about the web structure. However, in case of handling dangling nodes, we consider the model 2 as an ideal algorithm. In real cases when we do not know the whole structure of the web, we use both the predictive ranking algorithms and the modified PageRank algorithm in Model 2 to get the results of the ranks of the web Pages, and then compare both of them to the ranks of the web Pages calculated by the modified PageRank Algorithms in case that we have relatively more information about the web structure. More specifically, we snapshot the 11 matrices during process of crawling the page restricted within the domain cuhk.edu.hk, namely, $A_1,\ A_2,\ \ldots,\ A_{11}$. The numbers $Vnum[t]$ of pages visited successfully at time t and the total numbers $Tnum[t]$ of pages only found at time t are shown in the table 3.1.

By applying both the predictive algorithm and the modified PageRank Algorithm in Model 2 to these 11 data sets, we get different rank results

PreRank[t] t=1,2,. . . ,12;

PageRank[t] t=1,2,. . . ,12.

In our experiment we set $\alpha = 0.85$ and set the personalized factor f to be a uniform distribution in both algorithms. The iterative algorithm stops when the norm $||.||_1$ between the current ranking and the previous ranking is less than 0.000001. The numbers of iteration are shown in the table 3.2.

Time t	1	2	3	4	5	6	7	8	9	10	11
PreRank	5	3	2	2	2	2	2	2	2	2	2
PageRank	12	4	3	2	2	2	2	2	2	2	2

Table 3.2. Numbers of Iterations

From this table, in 3 cases out of 11 cases, the PreRank needs less iterations than PageRank. In the other 8 cases, they needs the same iterations. The computing complexities of both algorithms are Cn^2, where n is the number of all the nodes found.

We then calculate the ranking difference for nodes already found at time t between $PreRank[t]$ ($PageRank[t]$) and $PageRank[11]$ by the formula:

$$Diff1[t] = ||PreRank[t] - Cut(t, PageRank[11])/Sum[t]||_1,$$

$$Diff2[t] = ||PageRank[t] - Cut(t, PageRank[11])/Sum[t]||_1,$$

where $cut(t, PageRank[11])$ means the vector cut from PageRank[11] such that it has the same dimension as PreRank[t] and PageRank[t] , $Sum[t]$ means the sum of values in vector $cut(t, PageRank[11])$.

$Cut(t, PageRank[11])/Sum[t]$ is a normalized vector. Since we have more link information at time 11 than we have at time t ($t \leq 11$), we consider $Cut(t, PageRank[11])/Sum[t]$ as an "ideal" reference.

The value diff1[t] measure the difference between PreRank[t] and PageRank[11];

The value diff2[t] measure the difference between PageRank[t] and PageRank[11];

The results of diff1[t] and diff2[t] are shown in the following table and figure:

Time t	1	2	3	4	5	6
Diff1[t]	1.12	1.42	1.08	0.72	0.47	0.31
Diff2[t]	1.07	1.51	1.21	0.81	0.55	0.39
Time t	7	8	9	10	11	
Diff1[t]	0.19	0.16	0.13	0.11	0.09	
Diff2[t]	0.24	0.19	0.12	0.08	0.0	

Table 3.3. Results Within Domain CUHK Based on PageRank at time 11

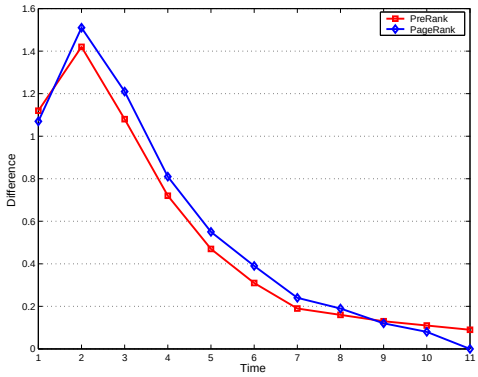


Figure 3.1. Results Within Domain CUHK Based on PageRank at time 11

From the figure, we can see that at time 1, PageRank[1] is closer than PreRank[i] to PageRank[11], this happen because at time 1, the data set is so small that the statistic estimation is not accurate sometimes. But as the time grows, from time 2 to time 8, PreRank[t] is closer than PageRank[t] to PageRank[11]. As we expected, as time t is near to the end 11, PageRank[t] again is closer than PreRank[i] to PageRank[11], this happen because

1. At time 11, the information contained in the data set is still incomplete, so the results from both the algorithms are not accurate.

2. We use the PageRank[11] as comparison reference, it is biased against PreRank[t], therefore it is natural that PageRank[t] is near to PageRank[t].

3. If we use PreRank[11] as comparison reference, at all time t except 1, PreRank[t] is closer to PreRank[11] than PageRank[t] does. The Results can be seen in the following figure:

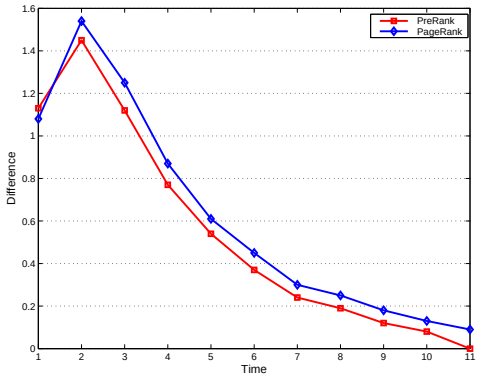


Figure 3.2. Results Within Domain CUHK Based on PreRank at time 11

3.3.2 Experiment Outside Domain cuhk.edu.hk

We also snapshot the 9 matrices during process of crawling the page outside the domain cuhk.edu.hk, namely, $A_1,\ A_2,\ \ldots,\ A_9$. The denotations $Vnum[t]$ and $Tnum[t]$ have the same meaning as the previous subsection. The values of them are shown in the table 3.4.

Time t	1	2	3	4	5
Vnum[t]	4611	6785	10310	16690	20318
Tnum[t]	87930	121961	164701	227682	290731
Time t	6	7	8	9	
Vnum[t]	23453	25417	28847	39824	
Tnum[t]	322774	362440	413053	882254	

Table 3.4. Description of Data Sets Outside Domain cuhk.edu.hk

By applying both the predictive algorithm and the modified PageRank Algorithm to these 9 data sets, we get different rank results

$PreRank[t]\ t = 1, 2, \ldots, 9; PageRank[t]\ t = 1, 2, \ldots, 9;$

Other experiment setting is same as the local experiments. The numbers of iteration are shown in the table 3.5

Time t	1	2	3	4	5	6	7	8	9
PreRank	2	2	2	1	1	1	1	1	1
PageRank	3	3	3	2	2	2	2	2	1

Table 3.5. Numbers of Iterations at Time t

From this table, in most cases, the PreRank needs less iterations than PageRank.

We then calculate the ranking difference for nodes already found at time t between PreRank[t] (PageRank[t])and PageRank[12] by the same procedure as the previous subsection. But the value diff1[t] measure the difference between PreRank[t] and PageRank[9]; The value diff2[t] measure the difference between PageRank[t] and PageRank[9];

The results of diff1[t] and diff2[t] are shown in figure 3.3:

From the figure, we can see that only at time 1, 2, and 3, PreRank[t] is closer than PageRank[t] to PageRank[9]. Although we only win these 3 cases, it is a surprise because the reference PageRank[9] is biased against our model, and PageRank[9] is extremely inaccurate because of the extremely small dataset compared to the extremely large whole web. As the time grows, from time 4 to time 8, PageRank[t] is closer than PreRank[t] to PageRank[9]. This

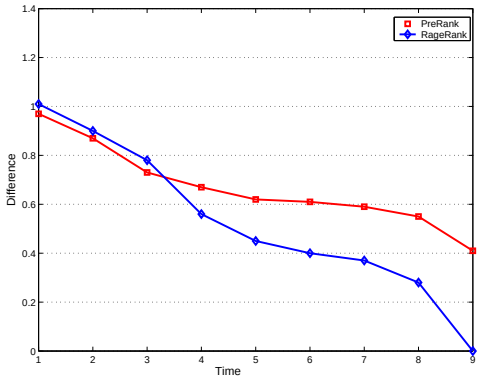


Figure 3.3. Results Outside Domain CUHK Based on PageRank at time 9

result seems that the accuracy in the case of crawling outside the doamin cuhk.edu.hk is worse than in the case of crawling with the domain. But the truth is that the data set obtained at time 9 is too much small compared to the whole web containing more than 4 billions of pages, and so the information at time 9 is extremely incomplete, and therefore both PreRank[9] and PageRank[9] are not accurate. This happens within our expectation.

If we use PreRank[9] as comparison reference, at all time t, PreRank[t] is closer to PreRank[9] than PageRank[t] does. The Results can be seen in the figure 3.4:

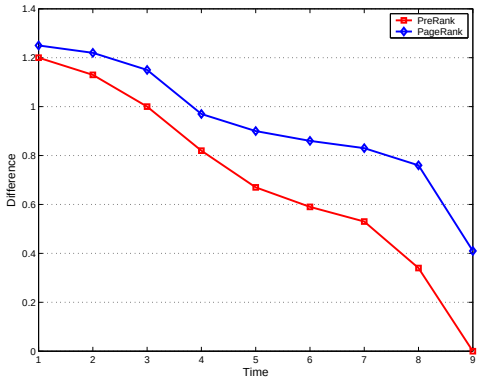


Figure 3.4. Results Outside Domain CUHK Based on PreRank at time 9

3.4 Discussion

In the previous model, we ignore dangling nodes of class 2. Dangling nodes of class 2 is different from dangling nodes of class 3. dangling nodes of class 2 can not be visited successfully while dangling nodes of class 3 have been successfully visited. So it is reasonable to handle them in different way. From the point of view that dangling nodes of class 2 contain useless information (because we can not visit them), our model does not need further

modification. However, dangling nodes of class 2 reflect bad structure about the web, and if a pages contains too many links that point to these nodes, then this page should be penalized. For this reason, dangling nodes of class 2 should not to be ignored simply and we suggest adopting the push-back algorithm in [6], in which dangling nodes of class 2 is called as penalty pages. We believe that push-back algorithm can be combined with our PreRank algorithm.

All the above experimental results except the number of iterations are within our expectation. Because our model mines more information about the web structure, the results of Predictive Ranking is more accurate than PageRank. In the local experiment, even in the case that we consider the results of PageRank as the reference, i.e., even in the case the reference is biased against our model, most early results calculated by PreRank are closer to the reference (calulated by PageRank) than the results calculated by PageRank. In the outside experiment, if we consider the results of PageRank as the reference, then we only win 3 cases; if we consider the results of PreRank as the reference, then we will all the cases. This suggests that PreRank performs better than PageRank in accuracy.

However, that PreRank needs less iteration exceeds our expectation. We can not explain why PreRank can perform better in speed. One possible reason is that the structure in the Matrix A in Predictive Ranking Model contains more accurate information (nearer to the global information) and thus each iteration adds more ”correct” value onto the ranking result, and finally less iteration can get the right answer. This suggests that PreRank performs better than PageRank in speed.

Chapter 4

Handle Missing Information for Decision Tree

4.1 Definition of the Generalized Dependency Degree

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

4.1.1 Formal Language to Describe Decision Rule

The decision language is defined by Pawlak [24, 25]. 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 formulas $For(B)$. 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 object objects $x \in U$ satisfying Φ by $||\Phi||_S$ 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; Φ and Ψ are referred to as condition and decision of the rule respectively. A decision rule $\Phi \rightarrow \Psi$ is called a *deterministic rule* in S if $||\Phi||_S \subseteq ||\Psi||_S$, an *indeterministic rule* otherwise. With every decision rule $\Phi \rightarrow \Psi$ we associate a conditional probability called the *certainty factor* (the *confidence* of the rule $\Phi \rightarrow \Psi$), we denote it by $Con(\Phi \rightarrow \Psi)$ which can be written as

$$Con(\Phi \rightarrow \psi) = \frac{card(||\Phi \wedge \Psi||_S)}{card(||\Phi||_S)}.$$

We denote the *strength* of decision rule $\Phi \rightarrow \Psi$ by $Str(\Phi \rightarrow \Psi)$ which is defined in [24] as:

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

4.1.2 Generalized Dependency Degree

We give our first form of the generalized dependency degree $\gamma'(C, D)$ in terms of equivalence relation as follows,

Definition 1:

$$\gamma'(C, D) = \frac{1}{|U|} \sum_{x \in U} \frac{|D(x) \cap C(x)|}{|C(x)|}, \tag{4.1}$$

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 have defined P-class for any attribute set P).

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

$$\gamma(C, D) = \frac{1}{|U|} \sum_{x \in U \wedge C(x) \subseteq D(x)} \frac{|D(x) \cap C(x)|}{|C(x)|}, \tag{4.2}$$

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 tell 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 be or not equal to 1.

Next we will interpret $\gamma'(C, D)$ from another point of view, i.e., we will change our view from the equivalence class to minimal decision rule.

Example 1: In Table 1, $A = \{a, b, c, d\}$, $U = \{e1, e2, e3, e4, e5, e6, e7\}$. We use Equation (4.1) to calculate $\gamma'(C, D)$ when $C = \{a, b, c\}$, $D = \{d\}$. Since $C(e1) = \{e1\}$, $C(e2) =\{e2\}$, $C(e3) =\{e3\}$, $C(e4) =\{e4\}$, $C(e5) =\{e5\}$, $C(e6) =\{e6\}$, $C(e7) =\{e7\}$, $D(e1) =D(e4) = D(e5) =\{e1, e4, e5\}$, $D(e2) = D(e3) = D(e6) = D(e7) = \{e2, e3, e6, e7\}$, we have
$\gamma'(C, D) = (\frac{|D(e1)\cap C(e1)|}{|C(e1)|}+\frac{|D(e2)\cap C(e2)|}{|C(e2)|}+\frac{|D(e3)\cap C(e3)|}{|C(e3)|}+\frac{|D(e4)\cap C(e4)|}{|C(e4)|}+\frac{|D(e5)\cap C(e5)|}{|C(e5)|}+\frac{|D(e6)\cap C(e6)|}{|C(e6)|}+\frac{|D(e7)\cap C(e7)|}{|C(e7)|})/7 =$
$(1 + 1 + 1 + 1 + 1 + 1 + 1)/7 = 1$.

Example 2: Also in Table 1, We calculate $\gamma'(C, D)$ and $\gamma(C, D)$ when $C = \{a\}$,$D = \{d\}$. Since $C(e1) = C(e2) = C(e3) = C(e7) = \{e1, e2, e3, e7\}, C(e4) = C(e5) = C(e6) = \{e4, e5, e6\}$,$D(e1) = D(e4) = D(e5) = \{e1, e4, e5\}$,$D(e2) = D(e3) = D(e6) = D(e7) = \{e2, e3, e6, e7\}$, we have $\gamma'(C, D) = 1/7(1/4 + 3/4 + 3/4 + 2/3 + 2/3 + 1/3 + 3/4) = 25/42$, while we have $\gamma(C, D) = 0$ according to Equation (4.2).

4.1.3 Connection between the Generalized Dependency Degree and Minimal Rule

We first give the definition of the minimal rule.

Definition 2: Let $C = \{a_1, a_2, a_3, ..., a_n\}$, $D = \{b_1, b_2, b_3, ..., b_m\}$. Then we call the rule

$$a_1 = u_1 \ \wedge \ a_2 = u_2 \ \wedge ... \wedge a_n = u_n \ \rightarrow b_1 = v_1 \ \wedge \ b_2 = v_2 \wedge ... \wedge b_m = v_m$$

a minimal rule, where $u_1 \in V_{a_1}, u_2 \in V_{a_2}, u_3 \in V_{a_3}, \cdots, u_n \in V_{a_n}, v_1 \in V_{b_1}, v_2 \in V_{b_2}, \cdots, v_m \in V_{b_m}$.

If $x \in U$, by $C(x) \to D(x)$ we denote the rule $a_1 = a_1(x) \ \wedge \ a_2 = a_2(x) \ \wedge \ a_3 = a_3(x) \wedge ... \wedge a_n = a_n(x) \ \to b_1 = b_1(x) \ \wedge \ b_2 = b_2(x) \wedge ... \wedge b_m = b_m(x)$, where $C(x)$ is the C-Class containing x, $D(x)$ is the D-class containing x, $a_i(x)$ is the value of x at the attribute a_i and $b_j(x)$ is the value of x at the attribute b_j.

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

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

$$\sum_{r \in MinR(C,D)} Str(r) \cdot Con(r) \tag{4.3}$$

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 $\gamma'(C, D)$ which is defined in terms of minimal rule. We explain this by the following.

Let X be a $(C \cup D)$-class. Then for all $y, x \in X, C(y) = C(x), D(y) = D(x)$, so for $x \in X$ we denote $C(x)$ by $C(X)$, $D(x)$ by $D(X)$, and $|D(x) \cap C(x)|/|C(x)|$ by $|D(X) \cap C(X)|/|C(X)|$. Since $|X| = |D(X) \cap C(X)|$, we have

$$\begin{aligned}
\gamma'(C, D) &= \frac{1}{|U|} \sum_{x \in U} \frac{|D(x) \cap C(x)|}{|C(x)|} \\
&= \frac{1}{|U|} \sum_{X \in U/(C \cup D)} \sum_{x \in X} \frac{|D(x) \cap C(x)|}{|C(x)|} \\
&= \frac{1}{|U|} \sum_{X \in U/(C \cup D)} \sum_{x \in X} \frac{|D(X) \cap C(X)|}{|C(X)|} \\
&= \frac{1}{|U|} \sum_{X \in U/(C \cup D)} |X| \frac{|D(X) \cap C(X)|}{|C(X)|} \\
&= \frac{1}{|U|} \sum_{X \in U/(C \cup D)} \frac{|D(X) \cap C(X)|^2}{|C(X)|} \\
&= \sum_{X \in U/(C \cup D)} \frac{1}{|U|} \frac{|D(X) \cap C(X)|^2}{|C(X)|} \\
&= \sum_{X \in U/(C \cup D)} \frac{|D(X) \cap C(X)|}{|U|} \cdot \frac{|D(X) \cap C(X)|}{|C(x)|} \\
&= \sum_{X \in U/(C \cup D)} Str(C(X) \to D(X)) \cdot Con(C(X) \to D(X))
\end{aligned}$$

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

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

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

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

4.2 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:

$$\gamma'(R_1, R_2) = \frac{1}{|U|} \sum_{x \in U} \frac{|R_2(x) \cap R_1(x)|}{|R_1(x)|}, \tag{4.4}$$

$$\gamma'(R_1, R_2) = \sum_{r \in MinR(R_1,R_2)} Str(r) \cdot Con(r). \tag{4.5}$$

Here the set $MinR(R_1, R_2)$ is the set of all the minimal rules, r is any rule in $MinR(R_1, R_2)$, 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

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

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, et al (1998) extends the rough set model to any binary relation. The Equation (4.4) is a general form, in which the equivalence relations can be understood as any binary relations. This explains why we say that the first form of the generalized dependency degree is a flexible form. In this paper, we only focus on the case of equivalence relation.

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:

$$\gamma(R_1,R_2) = \frac{1}{|U|} \sum_{x\in U \wedge R_1(x)\subseteq R_2(x)} \frac{|R_2(x)\cap R_1(x)|}{|R_1(x)|}, \tag{4.6}$$

$$\gamma(R_1,R_2) = \frac{1}{|U|} \sum_{x\in U \wedge R_1(x)\subseteq R_2(x)} Str(r)\cdot Con(r). \tag{4.7}$$

Throughout the rest of this paper, all the relations we use are all on the finite universe 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 $\gamma(R_1,R_2) \le \gamma'(R_1,R_2) \le 1$ holds.

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. Note that $\sum_{r\in MinR(R_1,R_2)} Str(r) = 1$. If $\gamma(R_1,R_2)$=1, then by Theorem 1, we have

$$1 = \gamma(R_1,R_2) \le \gamma'(R_1,R_2) \le 1,$$

and hence $\gamma'(R_1,R_2) = 1$.

If $\gamma'(R_1,R_2) = 1$, then by Equation (4.4), we have

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

and hence the equality $1 = |R_2(x)\cap R_1(x)|/|R_1(x)|$ holds for any $x \in U$. This yields $R_1(x) \subseteq R_2(x)$ for any $x \in U$(recall that U is a finite set). Therefore we have $\gamma(R_1,R_2) = \gamma'(R_1,R_2)$ by Equation (4.4) and (4.6).

If $\gamma(R_1,R_2) = \gamma'(R_1,R_2)$, also by Equation 4.4) and (4.6), we have $R_1(x) \subseteq R_2(x)$ for any $x \in U$, which yields $\gamma(R_1,R_2) = 1$.

Based on above discussion, the conclusion is true.

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

Proof. According to Equation (4.4), 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 lager equivalence class induced by the decision attribute is, the easier we can classify the objects into new D-class by employing attribute C. For example, in Table 1, $D = \{d\}$, $V_d = \{Y, N\}$, 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 the Table 1.1 becomes Table 4.1. D induces the equivalence relation $IND(D)$, and the set of the equivalence classes is calculated as $U/D = \{\{e1, e4, e5\}, \{e2, e3, e6, e7\}\}$; D' induces the equivalence relation $IND(D')$, and $U/D' = \{\{e1, e2, e3, e4, e5, e6, e7\}\}$. Then we classify object into U/D' easier than into U/D.

	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

Table 4.1. Influenza Data

By Theorem 3, we have

Theorem 4: For any given equivalence relation R_1.

$$\min_{R_2} \gamma'(R_1, R_2) = \gamma'(R_1, I_U),\ \max_{R_2} \gamma'(R_1, R_2) = \gamma'(R_1, U \times U) = 1,$$

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 properties about 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 $a_i \in R$, and $0 < b_i \in R$,$i = 1, 2, \ldots, n$.

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 $\mu_1, \mu_2, \ldots, \mu_n \geq 0, \quad \mu_1 + \mu_2 + \cdots + \mu_n = 1$. 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.

Theorem 5: For any given equivalence relation R_2,

$$\min_{R_1} \gamma'(R_1, R_2) = \gamma'(U \times U, R_2).$$

Proof. Suppose that there are m R_2-classes, and they are $X_1, X_2, X_3, \ldots, X_m$. We will calculate $\gamma'(R_1, R_2)$ when $R_1 = U \times U$. By Equation (4.4), when $R_1 = U \times U$, we have

$$\begin{aligned}
\gamma'(R_1, R_2) &= \frac{1}{|U|} \sum_{X \in U/R_2} \sum_{x \in X} \frac{|X \cap R_1(x)|}{|R_1(x)|} \\
&= \frac{1}{|U|} \sum_{X \in U/R_2} \sum_{x \in X} \frac{|X \cap U|}{|U|} \\
&= \frac{1}{|U|} \sum_{X \in U/R_2} \sum_{x \in X} \frac{|X|}{|U|} \\
&= \frac{1}{|U|} \sum_{X \in U/R_2} \frac{|X|^2}{|U|} \\
&= \frac{1}{|U|^2} \sum_{i=1}^{m} |X_i|^2.
\end{aligned}$$

Then we analyze $\gamma'(R_1, R_2)$ for any given R_1. In order to achieve this goal, we need to see into 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 $i = 1, 2, ..., m$. 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 $\underset{x \in U}{\cup} R_1(x) = U$.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 (4.4), we have

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

Because that $U \supseteq R_1(x_{i1}) \cup R_1(x_{i2}) \cup \cdots \cup R_1(x_{ik_i})$ and that $R_1(x_{i1})$, $R_1(x_{i2})$, $\cdots$, $R_1(x_{ik_i})$ are disjoint with each other, we have

$$|U| \geq |R_1(x_{i1})| + |R_1(x_{i2})| + \cdots + |R_1(x_{ik_i})|.$$

Note that

$$|X_i| = |R_1(x_{i1}) \cap X_i| + |R_1(x_{i2}) \cap X_i| + \cdots + |R_1(x_{ik_i}) \cap X_i|,$$

so by Lemma 1, we have

$$\frac{|X_i|^2}{\sum_{j=1}^{k_i} |R_1(x_{ij})|} \leq \frac{|R_1(x_{i1}) \cap X_i|^2}{|R_1(x_{i1})|} + \frac{|R_1(x_{i2}) \cap X_i|^2}{|R_1(x_{i2})|} + \cdots + \frac{|R_1(x_{ik_i}) \cap X_i|^2}{|R_1(x_{ik_i})|},$$

and hence

$$\begin{aligned}
|X_i|^2 &\leq (\sum_{j=1}^{k_i} |R_1(x_{ij})|)(\frac{|R_1(x_{i1}) \cap X_i|^2}{|R_1(x_{i1})|} + \frac{|R_1(x_{i2}) \cap X_i|^2}{|R_1(x_{i2})|} + \cdots + \frac{|R_1(x_{ik_i}) \cap X_i|^2}{|R_1(x_{ik_i})|}) \\
&\leq \frac{|U||R_1(x_{i1}) \cap X_i|^2}{|R_1(x_{i1})|} + \frac{|U||R_1(x_{i2}) \cap X_i|^2}{|R_1(x_{i2})|} + \cdots + \frac{|U||R_1(x_{ik_i}) \cap X_i|^2}{|R_1(x_{ik_i})|}).
\end{aligned}$$

Therefore $\gamma'(U \times U, R_2) \leq \gamma'(R_1, R_2)$.

This theorem means that the generalized dependency degree is minimal when there is only one equivalence class induced by R_1, therefore from R_1 we can not learn more useful information about R_2 in such case.

Similarly we have

Theorem 6: **(Anti-Partial Order Preserving Property)** For any equivalence relations $R_1,\ R_2,\ R$. If $R_1 \subseteq R$, then $\gamma'(R_1, R_2) \geq \gamma'(R, R_2)$.

Proof. Since $R_1 \subseteq R$, each R-class is the union of some R_1-classes, 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

$$
\begin{aligned}
R(y_{i1}) \cap X_i &= (R_1(x_{i1}) \cap X_i) \cup (R_1(x_{i2}) \cap X_i) \cup \cdots \cup (R_1(x_{ip_1}) \cap X_i), \\
R(y_{i1}) &\supseteq R_1(x_{i1}) \cup R_1(x_{i2}) \cup \cdots \cup R_1(x_{ip_1}), \\
R(y_{i2}) \cap X_i &= (R_1(x_{ip_1+1}) \cap X_i) \cup (R_1(x_{ip_1+2}) \cap X_i) \cup \cdots \cup (R_1(x_{ip_2}) \cap X_i), \\
R(y_{i2}) &\supseteq R_1(x_{ip_1+1}) \cup R_1(x_{ip_1+2}) \cup \cdots \cup R_1(x_{ip_2}), \\
&\ldots, \\
R(y_{il_i}) \cap X_i &= (R_1(x_{ip_{l_i-1}+1}) \cap X_i) \cup (R_1(x_{ip_{l_i-1}+2}) \cap X_i) \cup \cdots \cup (R_1(x_{ip_{l_i}}) \cap X_i), \\
R(y_{il_i}) &\supseteq R_1(x_{ip_{l_i-1}+1}) \cup R_1(x_{ip_{l_i-1}+2}) \cup \cdots \cup R_1(x_{ip_{l_i}}).
\end{aligned}
$$

By the proof of Theorem 5, 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 $a_{ij} = |X_i \cap R_1(x_{ij})|$, $b_{ij} = |R_1(x_{ij})|/|U|$, $i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, k_i$. And

$$\begin{aligned}\gamma'(R, R_2) &= \frac{1}{|U|^2}\sum_{i=1}^{m}\sum_{j=1}^{l_i}\frac{|U|}{|R(y_{ij})|}|X_i \cap R(y_{ij})|^2 \\ &= \frac{1}{|U|^2}\sum_{i=1}^{m}\sum_{j=1}^{l_i}\frac{{a'_{ij}}^2}{b'_{ij}},\end{aligned}$$

where

$$\begin{aligned}
a'_{i1} &= |X_i \cap R(y_{i1})| = a_{i1} + a_{i2} + \cdots + a_{ip_1}, \\
a'_{i2} &= |X_i \cap R(y_{i2})| = a_{ip_1+1} + a_{ip_1+2} + \cdots + a_{ip_2}, \\
&\cdots, \\
a'_{il_i} &= |X_i \cap R(y_{il_i})| = a_{ip_{l_i-1}+1} + a_{ip_{l_i-1}+2} + \cdots + a_{ip_{l_i}}, \\
b'_{i1} &= |R(y_{i1})|/|U| \geq b_{i1} + b_{i2} + \cdots + b_{ip_1}, \\
b'_{i2} &= |R(y_{i2})|/|U| \geq b_{ip_1+1} + b_{ip_1+2} + \cdots + b_{ip_2}, \\
&\cdots, \\
b'_{il_i} &= |R(y_{il_i})|/|U| \geq b_{ip_{l_i-1}+1} + b_{ip_{l_i-1}+2} + \cdots + b_{ip_{l_i}},
\end{aligned}$$

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

$$\begin{aligned}\sum_{j=1}^{k_i}\frac{a_{ij}^2}{b_{ij}} &= \frac{a_{i1}^2}{b_{i1}} + \frac{a_{i2}^2}{b_{i2}} + \cdots + \frac{a_{ip_1}^2}{b_{ip_1}} \\ &+ \frac{a_{ip_1+1}^2}{b_{ip_1+1}} + \frac{a_{ip_1+2}^2}{b_{ip_1+2}} + \cdots + \frac{a_{ip_2}^2}{b_{ip_2}} \\ &+ \cdots \\ &+ \frac{a_{ip_{l_i-1}+1}^2}{b_{ip_{l_i-1}+1}} + \frac{a_{ip_{l_i-1}+2}^2}{b_{ip_{l_i-1}+2}} + \cdots + \frac{a_{ip_{l_i}}^2}{b_{ip_{l_i}}}\end{aligned}$$

$$\geq \frac{(\sum_{j=1}^{p_1} a_{ij})^2}{\sum_{j=1}^{p_1} b_{ij}} + \frac{(\sum_{j=p_1+1}^{p_2} a_{ij})^2}{\sum_{j=p_1+1}^{p_1} b_{ij}} + \cdots + \frac{(\sum_{j=p_{l_i-1}+1}^{p_{l_i}} a_{ij})^2}{\sum_{j=p_{l_i-1}+1}^{p_{l_i}} b_{ij}}$$

$$\geq \frac{{a'_{i1}}^2}{b'_{i1}} + \frac{{a'_{i2}}^2}{b'_{i2}} + \cdots + \frac{{a'_{il_i}}^2}{b'_{il_i}}.$$

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

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 equivalence class induced by the decision attribute is, the more difficult we can classify the objects into new D-class by employing attribute C. For example, in Table 1, let $C = \{c\}$, $V_c = \{0, 1, 2\}$, 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 the Table 1 becomes Table 3. Thus it is harder for us to classify objects into D-class by employing the attribute C'.

	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

Table 4.2. Influenza Data

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 6, $\gamma'(C', D) \leq \gamma'(C, D)$. This means that generally, the less the condition attribute set contains attributes, the harder we classify the objects into D-class by employing the condition attribute set.

Theorem 7: For any given equivalence relation R_2, we have

$$\max_{R_1} \gamma'(R_1, R_2) = \gamma'(I_U, R_2) = 1, \ \min_{R_1} \gamma'(R_1, R_2) = \gamma'(U \times U, R_2).$$

Proof. It is immediate by Theorem 6.

Theorem 8:

$$\min_{R_1,R_2} \gamma'(R_1,R_2) = \frac{1}{|U|},\ \max_{R_1,R_2} \gamma'(R_1,R_2) = 1.$$

Proof. By Theorem 3 and Theorem 6, we only need to verify that $\gamma'(U \times U, I_U) = 1/|U|$. Let $R_1 = U \times U, R_2 = I_U$. According to Equation (4.4), we have

$$\begin{aligned}
\gamma'(U \times U, I_U) &= \frac{1}{|U|} \sum_{x \in U} \frac{|R_2(x) \cap R_1(x)|}{|R_1(x)|} \\
&= \frac{1}{|U|} \sum_{x \in U} \frac{|\{x\} \cap U|}{|U|} \\
&= \frac{1}{|U|} \sum_{x \in U} \frac{1}{|U|} = \frac{1}{|U|}.
\end{aligned}$$

This means that for any two equivalence relations R_1, R_2, R_2 depends on R_1 to some degree at least $1/|U|$. When the number of objects tends to infinity, the minimum tends to zero.

4.3 Probabilistic Form of Generalized Dependency Degree

The third form of the generalized dependency degree $\gamma'(C,D)$ can be rewritten as

$$\begin{aligned}
\gamma'(C,D) &= \sum_c \sum_d (\Pr[c] \cdot \Pr^2[d|c]) \\
&= \sum_c (\Pr[c] \cdot \sum_d \Pr^2[d|c]),
\end{aligned} \tag{4.8}$$

Next we will explain why the generalized dependency degree $\gamma'(C,D)$ can be rewritten as the above form.

Let $C = \{a_1, a_2, ..., a_n\}$,$D = \{b_1, b_2, ..., b_m\}$. Then the minimal rule

$$a_1 = u_1 \ \wedge \ a_2 = u_2 \ \wedge ... \wedge a_n = u_n \ \rightarrow b_1 = v_1 \ \wedge \ b_2 = v_2 \wedge ... \wedge b_m = v_m$$

can be denoted by $c \rightarrow d$, where $c = (u_1, u_2, \ldots, u_n)$, $d = (v_1.v_2, \ldots, v_m)$. Then

$$Con(c \rightarrow d) = \Pr(d|c),$$

$$Str(c \rightarrow d) = \Pr(c,d).$$

According to $\Pr(c,d)/\Pr(c) = \Pr(d|c)$ and Equation (4.3), we have

$$
\begin{aligned}
\gamma'(C,D) &= \sum_{r \in MinR(C,D)} Str(r) \cdot Con(r) \\
&= \sum_{c,d} Str(c \to d) \cdot Con(c \to d) \\
&= \sum_{c,d} \Pr(c,d) \cdot \Pr(d|c) \\
&= \sum_{c} \sum_{d} \Pr(c,d) \cdot \Pr(d|c) \\
&= \sum_{c} \sum_{d} \Pr(c) \cdot \frac{\Pr(c,d)}{\Pr(c)} \cdot \Pr(d|c) \\
&= \sum_{c} (\Pr(c) \cdot \sum_{d} \Pr^2(d|c)).
\end{aligned}
$$

Before we move on to the next section, we give one more property of the generalized dependency degree based on the form of Equation (4.8), i.e., in term of probability.

Theorem 9: If the decision attributes are independent of the conditional attributes, then $\gamma'(C,D) = \gamma'(U \times U, IND(D))$, i.e., the generalized dependency degree $\gamma'(C,D)$ takes the minimal value of $\gamma'(R_1, IND(D))$ among all possible equivalence relation R_1

Proof. If D is independent of C, then $Pr(d|c) = Pr(d)$, and so we have

$$
\begin{aligned}
\gamma'(C,D) &= \sum_{c} (\Pr[c] \cdot \sum_{d} \Pr^2[d|c]) \\
&= \sum_{c} (\Pr[c] \cdot \sum_{d} \Pr^2[d]) \\
&= (\sum_{c} \Pr[c]) \cdot (\sum_{d} \Pr^2[d]) \\
&= \sum_{d} \Pr^2[d].
\end{aligned}
$$

By the proof of Theorem 5, the above formula is exactly $\gamma'(U \times U, IND(D))$, which means that under the condition of independency, $\gamma'(C,D) = \gamma'(U \times U, IND(D))$, and that under the same condition, $\gamma'(C,D)$ takes the minimal value of $\gamma'(R_1, IND(D))$ among all possible equivalence relation R_1.

4.4 Definition of the Generalized Dependency Degree γ' in 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 and using minimal rule.

4.4.1 How to Handle Missing Values in Incomplete Information Systems

Here we introduce a new approach by replacing the missing value by its possible attribution shown in Table 4.3:

	a	b	c	d
e_1	Y	Y	Normal(0)	N
e_2	Y	$\{P_1/Y, P_2/N\}$	High(1)	Y
e_3	Y	Y	$\{S_1/0, S_2/1, S_3/2\}$	Y
e_4	N	$\{Q_1/Y, Q_2/N\}$	Normal(0)	N
e_5	N	N	High(1)	N
e_6	N	Y	Very High(2)	Y
e_7	Y	N	High(1)	Y

Table 4.3. Influenza Data

In the e_2-row, by $\{P_1/Y, P_2/N\}$ we mean that e_2 takes the value Y with a probability P_1, and N with a probability 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 to 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 the column b, i.e., $P_1 = Q_1 = 3/5, P_2 = Q_2 = 2/5$; let S_1, S_2, S_3 take the values of the distribution of 0,1 and 2 in column c, i.e., $S_1 = 2/6, S_2 = 3/6, S_3 = 1/6$.

4.4.2 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 the Equation 4.3

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

as our definition of the generalized dependency degree γ' 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 by example of the Influenza Data in Table 4.3.

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

If $x \in U$ satisfies Φ with a probability of p, then the object x belongs to the set $||\Phi||_S$ with a probability of p, and we write the element x in $||\Phi||_S$ as p/x. For example, in the Table 4.3, let Φ be the formula $b = Y$. $e1$ satisfies the formula $b = Y$ with a probability of 1, the probability of Y in e_1-row, b-column; 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

$$||\Phi||_S = \{1/e_1, 0.6/e_2, 1/e_3, 0.6/e_4, 0/e_5, 1/e_6, 0/e_7\}.$$

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

Then we define $||\Phi||_S$ inductively as follows: If x satisfies Φ with a probability of p, and x satisfies Ψ with a probability of q, then x satisfies $\Phi \wedge \Psi$ with a probability of pq; x satisfies $\sim \Phi$ with a probability of $1 - p$; x satisfies $\Phi \vee \Psi$ with a probability of $1 - (1-p)(1-q)$. Note that our support $||\Phi||_S$ of Φ has the same expression as fuzzy set. So we can also define $||\Phi||_S$ inductively in term of fuzzy set as follows:

$$\begin{aligned}
F1 \quad &: \quad ||a = v||_S = \{p(x)/x | x \in U, P[a(x) = v] = p(x)\} \\
&\qquad \text{for } a \in B \text{ and } v \in V_a \\
F2 \quad &: \quad ||\Phi \vee \Psi||_S = ||\Phi||_S + ||\Psi||_S \\
F3 \quad &: \quad ||\Phi \wedge \Psi||_S = ||\Phi||_S \cdot ||\Psi||_S \\
F4 \quad &: \quad || \sim \Phi||_S = \sim ||\Phi||_S
\end{aligned}$$

where $||\Phi||_S + ||\Psi||_S$ is the algebraic sum of the fuzzy sets $||\Phi||_S$ and $||\Psi||_S$, $||\Phi||_S \cdot ||\Psi||_S$ is the algebraic product of the fuzzy sets $||\Phi||_S$ and $||\Psi||_S$, and $\sim ||\Phi||_S$ is the complement of the fuzzy set $||\Phi||_S$ [29].

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

$$card(||\Phi||_S) = \sum_{x \in ||\Phi||_S} p(x). \tag{4.9}$$

Then as an example we will calculate the generalized dependency degree between $C = \{a, b, c\}$ and $D = \{d\}$ in Table 4.3 by the Equation 4.3. We first need to calculate the confidence and strength of each minimal decision rule using the following definitions in which we have already defined whatever we need.

$$Con(\Phi \rightarrow \Psi) = card(||\Phi \wedge \Psi||_S)/card(||\Phi||_S) \tag{4.10}$$

$$Str(\Phi \rightarrow \Psi) = card(||\Phi \wedge \Psi||_S)/card(U) \tag{4.11}$$

Example 3: We show in the following the process of calculation of confidence and strength of one minimal rule while the results of all minimal rules is listed in Table 4.4. Since $||a = Y \wedge b = Y \wedge c = 0 \wedge d = Y||_S = \{S_1/e_3\}$,$|\{S_1/e_3\}| = S_1 = 2/6$,$||a = Y \wedge b = Y \wedge c = 0||_S = \{1/e_1, S_1/e_3\}$,$|\{1/e_1, S_1/e_3\}| = 1 + S_1 = 1 + 2/6 = 4/3$, we have the minimal rule $a = Y \wedge b = Y \wedge c = 0 \rightarrow d = Y$ with confidence=1/4, strength=1/21;

a	b	c	d	Con	Str	a	b	c	d	Con	Str
Y	Y	0	Y	$\frac{1}{4}$	$\frac{1}{21}$	N	Y	0	Y	0	0
Y	Y	0	N	$\frac{3}{4}$	$\frac{1}{7}$	N	Y	0	N	1	$\frac{3}{35}$
Y	Y	1	Y	1	$\frac{11}{70}$	N	Y	1	Y	0	0
Y	Y	1	N	0	0	N	Y	1	N	0	0
Y	Y	2	Y	1	$\frac{1}{42}$	N	Y	2	Y	1	$\frac{1}{7}$
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	$\frac{2}{35}$
Y	N	1	Y	1	$\frac{1}{5}$	N	N	1	Y	0	0
Y	N	1	N	0	0	N	N	1	N	1	$\frac{1}{7}$
Y	N	2	Y	0	0	N	N	2	Y	0	0
Y	N	2	N	0	0	N	N	2	N	0	0

Table 4.4. Results of All Minimal Rules

So we have $\gamma'(C, D) = \sum_{r \in MinR(C,D)} Str(r) \cdot Con(r) = 1/21 \cdot 1/4 + 1/7 \cdot 3/4 + 11/70 \cdot 1 + 1/42 \cdot 1 + 1/5 \cdot 1 + 3/35 \cdot 1 + 1/7 \cdot 1 + 2/35 \cdot 1 + 1/7 \cdot 1 = 13/14$.

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

Theorem 10: 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 $\sum_{r \in MinR(C,D)} Str(r)$ and there are totally $|U|$ objects, we have $\sum_{r \in MinR(C,D)} Str(r) = 1$. It is obvious that $Con(r) \leq 1$ for all rule r, so we have

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

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

4.5 Experiments

In the above sections, we have given a deeper understanding of the generalized dependency degree by presenting its various forms and developing its various properties. Next, 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. We discard in the new C4.5 algorithm the MDL principal by which the original C4.5 can correct the split selection bias towards the continuous attribute. Since the conditional entropy has the meaning of average code length, it is compatible with the MDL principal in the original C4.5, however the generalized dependency degree means that to what degree the decision attribute depend on the condition attribute, so the new C4.5 using the generalized dependency degree is not compatible with the MDL principal and we discard it. One more thing that we change the original C4.5 is that we let the procedure of building tree stop earlier by a new criterion: in the current node, if for every attribute, the number of the gain cases is less than a given value (< 1.0), then the splitting procedure stops. The number of the gain cases is calculated by multiplying the number of cases in current node and the dependency gain.

C4.5 algorithm uses a divide-and-conquer approach to grow decision trees [27]. Next, we describe the C4.5 algorithm roughly. The readers are recommended to read the book [27] for a better understanding.

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

$$G(D, \{a\}) = H(D) - H(D|\{a\}).$$

Then choose the attribute which has the maximum gain among all the condition attributes, 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 the algorithm is run under the default option, then for every condition attribute a, its information gain ratio is computed by the formula

$$\frac{H(D) - H(D|\{a\})}{H(C)}.$$

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

We replace the information gain in the original C4.5 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 5, $G(D, \{a\}) \geq 0$.

To make it easier for the readers to repeat the experiments, we describe in the Appendix how the new C program is obtained and how our experiments are conducted in details.

Both the original C4.5 and the new C4.5 are applied to all the same eleven data sets with missing values as used in [28]. These eleven data sets are from the UCI Repository. Note that the datasets we use may have slight difference with what Quinlan [28] uses, as Quinlan points out in our private corresponding. For example, the anneal dataset we use has a different order of the cases with what Quinlan uses. The Table 4.5 is a description of the datasets we use. The first column refers to the names of the datasets, the second column refers to the numbers of cases in each datasets, the third column refers to the number of continuous attributes, and the final column refers to the number of discrete attribute.

dataset	Cases	Classes	Cont	Discr
Anneal	898	6	6	32
Auto	205	6	15	10
Breast-w	699	2	9	0
Colic	368	2	7	15
Credit-a	690	2	6	9
Heart-c	303	2	6	7
Heart-h	294	2	8	5
Hepatitis	155	2	6	13
Allhyper	3772	5	7	22
Labor	57	2	8	8
Sick	3772	2	7	22

Table 4.5. Description of the Datasets

The experiments are conducted on the workstation whose hardware model is Nix Dual Intel Xeon 2.2GHz,

whose RAM is 1GB, and whose OS is Linux Kernel 2.4.18-27smp (RedHat7.3).

Both algorithms use ten-fold cross-validations with each task. The figures shown in the Table 4.6 is the mean error rate of ten-fold cross-validations.

dataset	O-g-s(%)	**pruned**(%)	**N-g-s**(%)	pruned (%)
Anneal	3.9	**4.6**	6.1	7.9
Auto	20.5	**22.0**	**22.0**	22.5
Breast-w	5.7	4.3	**4.2**	4.5
Colic	19.8	**16.0**	16.3	15.4
Credit-a	19.7	17.1	**15.2**	15.6
Heart-c	22.4	**21.4**	23.4	23.1
Heart-h	24.2	22.8	**20.7**	21.1
Hepatitis	20.0	19.9	**19.3**	19.3
Allhyper	1.4	1.4	**1.1**	1.2
Labor	24.7	26.3	**15.7**	19.3
Sick	1.2	1.1	**1.0**	1.0

Table 4.6. Mean error rates of the original C4.5 and the new C4.5 on the data sets with missing values

The second column and third column in Table 4.6 are the results before pruning and after pruning respectively obtained by running the command

xval.sh filestem 10 –g –s

in the original C4.5 system. Which means the gain criteria (not the gain ratio) is used, and the MDL principle is used to correct the bias towards continuous attributes with numerous distinct values. Moreover the grouping method is used. The fourth column and the fifth column are the results before pruning and after pruning respectively obtained by running the same command in the new C4.5 system. Which means the new gain criteria based on the generalized dependency degree is used, and the MDL principle is not used. The grouping method is also used. The final row refers to the sum of results of the experiments on the twenty datasets.

The figures shown in the Table 4.7 is about the average run time of ten-fold cross-validations. The time unit in Table 4.7 is 0.01 second. The second column in Table 4.7 refers to the average run time of the original procedure C4.5 in the ten-fold cross-validations. The third column and the forth column refer to the average run time and the reduced time rate based on the second column of the changed C4.5 procedure without and with the

pruning procedure respectively. Note that the run time does not include the run time for data preparation for cross-validation and the run time for final result report in both C4.5 systems.

dataset	O-g-s	N-g-s unpruned	reduced rate(%)	N-g-s pruned	reduced rate(%)
Anneal	6.800	4.600	32.4	5.100	25.0
Auto	9.700	2.600	73.2	2.600	73.2
Breast-w	1.600	1.000	37.5	1.000	37.5
Colic	4.100	1.500	63.4	1.500	63.4
Credit-a	8.300	2.400	71.1	2.500	69.9
Heart-c	2.000	0.700	65.0	0.900	55.0
Heart-h	1.900	0.700	63.2	0.600	68.4
Hepatitis	0.900	0.600	33.3	0.700	22.2
Allhyper	45.000	18.500	58.9	18.500	58.9
Labor	0.400	0.100	75.0	0.400	0.0
Sick	38.100	17.100	55.1	20.800	45.4

Table 4.7. Average run time of the original C4.5 and the new C4.5

The experiments show that the generalized dependency degree $\gamma'(C, D)$ is a useful measure in incomplete information systems. We compare in three folds the new C4.5 algorithm using the generalized dependency degree with the original C4.5 algorithm using the conditional entropy:

1. Theoretical complexity: The generalized dependency degree $\gamma'(C, D)$ itself is somewhat complete. In the new C4.5 it does not need the MDL principal to correct the bias towards the continuous attribute, while the conditional entropy needs the MDL principal to achieve the competitive prediction accuracy. Moreover, from the experiment, we find that the pruning procedure in the original C4.5 algorithm can be omitted in the new C4.5 algorithm. Table 4.6 and Table 4.7 show that in the new C4.5 algorithm, omitting the pruning procedure can achieve a better performance both in speed and prediction accuracy.

2. Speed: To compute $\gamma'(C, D)$, we only need to compute the square of the frequency, while the computation of the often used conditional entropy needs to compute the time consuming logarithm of the frequency. Furthermore, the building tree procedure in the new C4.5 algorithm stops earlier. Omitting the pruning procedure can also save us an amount of time. This explained why the new C4.5 procedure with pruning procedure runs much faster than the original C4.5 procedure. In fact, the new C4.5 procedure run at about half of the time run by the original C4.5 procedure, and the new C4.5 procedure without pruning procedure can run a little faster further. Note that the

original C4.5 algorithm is the fastest algorithm in training time among the thirty-three old and new classification algorithms [17].

3. Prediction accuracy: The original C4.5 algorithm performs best by using the option –g –s and using the pruning procedure, while the new C4.5 algorithm also performs best by using the same option, but it does not need the extra pruning procedure. So we only compare the third column with the fourth column in Table 4.6, i.e., the result after pruning of original C4.5 algorithm with option –g –s with the result before pruning of the new C4.5 algorithm with the same option. In the third column and fourth column, the less one is written in bold. And we find that the new C4.5 algorithm performs better than the original C4.5 algorithm in prediction accuracy in these eleven data sets with missing values. Note that the prediction accuracy of the original C4.5 algorithm is not statistically significantly different from POL whose prediction accuracy is best among the thirty-three old and new classification algorithms [17]. The new C4.5 algorithm seems more successful in the dataset labor, on which the algorithm achieve a 15.7% prediction error rate while the original has 26.3% error rate.

We believe that after further investigation on the new C4.5 algorithm, the overall performance of the new C4.5 algorithm will perform better.

Chapter 5

Conclusions and Future Work

Using simple statistic method to handle missing values in the area of link analysis and in the area of decision tree achieves better performance both in accuracy and in speed.

Using known information to predict the number of inlinks for each page that have already been found is an efficient way to predict the unknown web structure in link analysis. Our experimental results suggest that PreRank need less iteration and performs better than PageRank in accuracy. Continue this work, there is much work we can do:

1. Speed up the Predictive ranking algorithm. Because the matrix A has special structure, it is possible to exploit this special structure to speed up the algorithm without losing accuracy. Moreover, the special web structures, such as block structure, hierarchy structure and directory structure, can also be used to speed up the algorithm.

2. Look both inside and outside a single matrix to get a more accurate model. In this work, we only predict a kind of information contained in the single data. In fact, we also can predict more by exploiting the information changed dynamically, for example, use the ARMA model to predict the future rank. Moreover, we can use the history information to adjust the link density $d^-(v_i)/n$ (or $d_j^-(v_i)/n$).

3. Conduct experiment on large real data set to support the block predictive ranking model.

Using the estimated probabilistic distribution as a method to extend the generalized dependency degree to the case of incomplete information system is a natural idea. This way work well in decision tree. Moreover, the generalized dependency degree is a good measure, it has three different kind of form, and it has many properties.

Among our three different forms of the generalized dependency degree, the first form (in terms of equivalence relation) of the measure is most important. Besides its simplicity, the first form is flexible, and therefore can be extended not only to equivalence relation but also to arbitrary relation. Moreover, it bridges the gap between the dependency degree γ defined in terms of rough set and the probabilistic form of the generalized dependency

degree γ'. The first form (in terms of equivalence relation) and the second form (in terms of minimal rule) share the advantage of being easily understood while the third form (in terms of probability) of the measure is computing-efficient. So these three forms of the measure can be used in different situations. When we want to extend the measure to more complicated data structure (such as partial order relation, totally order relation or others) than equivalence relation, or when we want to find some properties about this measure, we can resort to the first two forms of the measure. When we use it in the computing situation, the third form of the measure may be the best choice.

The generalized dependency degree γ' has good properties, such as Partial Order Preserving Property and Anti-Partial Order Preserving Property. Besides, its value is between zero and one. Therefore, it can be served as an index to measure how well decision attributes depend on conditional attributes. Its ability of being used in incomplete information systems is its another advantage. Furthermore, because it is more accurate than the dependency degree γ, it can be a substitute of the dependency degree in Rough Set Theory, and because it is less time-consuming and simpler than the conditional entropy, it can be a possible substitute as an information measure. Our experiments only show one possible such substitute in the field of decision tree. While the new C4.5 algorithm using the generalized dependency degree performs better in run time and in precision accuracy in these 11 data sets with missing values, we point out that we seem not to exploit fully in our experiments the properties of the generalized dependency degree γ'. For example, the generalized dependency degree is defined as the relation between two sets of attributes, which suggests that we can split the node on more than two attributes while in the current version of C4.5 algorithm, we only split the node only on one attribute.

Further study on the new C4.5 algorithm and on the possible application of the generalized dependency degree will be our future work.

The success of the Predictive Ranking Algorithm in Chapter 3 and the new C4.5 algorithm in Chapter 4 suggests that it is possible to deepen and widen the current work. For example, we can estimate the web structure more accurately (Block Predictive Ranking Model in Section 3.2 is just one possible), we can branch on more than one attribute to improve the current new C4.5 algorithm, and we need to investigate other areas in which such simple processing missing information method can work.

Bibliography

[1] G. Amati, I. Ounis, and V. Plachouras. The dynamic absorbing model for the web. Technical Report TR-2003-137, University of Glasgow, Apr. 2003.

[2] L. Breiman, F. J. H., R. A. Olshen, and S. C. J. *Classification and regression trees*. Belmont: Wadsworth International Group, 1984.

[3] J. Cho and R. E. Adams. Page quality: In search of an unbiased web ranking. Technical report, UCLA Computer Science Department, Nov. 2003.

[4] J. Cho and S. Roy. Impact of search engines on page popularity. In *Proceeding of the 13th World Wide Web Conference*, pages 20–29, 2004.

[5] M. Dalkilic and E. Robertson. Information dependencies. In *Proceedings of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principals of Database Systems*, pages 245–253, Dallas, Texas, 2000.

[6] N. Eiron, K. S. McCurley, and J. A. Tomlin. Ranking the web frontier. In *Proceeding of the 13th World Wide Web Conference*, pages 309–318, 2004.

[7] G. Gediga. Rough approximation quality revisited. *Artificial Intelligence*, 132:219–234, 2001.

[8] C. Giannella and E. Robertson. On approximation measures for functional dependencies. *Information Systems*, 29(6):483–507, 2004.

[9] L. A. Goodman and W. H. Kruskal. Measures of association for cross classifications. *Journal of the American Statistical Association*, 49(268):732–764, 1954.

[10] S. Handschuh, S. Staab, and R. Volz. On deep annotation. In *Proceeding of the 12th World Wide Web Conference*, pages 431–438, 2003.

[11] A. Hassanien. 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, 2004.

[12] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub. Exploiting the block structure of the web for computing pagerank. Technical report, Stanford University, 2003.

[13] M. Kryszkiewicz. Rough set approach to incomplete information systems. *Information Sciences*, 112:39–49, 1998.

[14] M. Kryszkiewicz. Rules in incomplete information systems. *Information Sciences*, 113:271–292, 1999.

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

[16] Y. Leung and D. Li. Maximal consistent block technique for rule acquisition in incomplete information systems. *Information Sciences*, 153:85–106, 2003.

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

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

[19] C. R. MacCluer. The many proofs and applications of perron's theorem. *SIAM Review*, 42(3):487–498, 2000.

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

[21] K. Nambiar. Some analytic tools for the design of relational database systems. In *Proceedings of the Sixth International Conference on Very Large Databases*, page 417C428, Montreal, Quebec, Canada, 2000.

[22] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report Paper SIDL-WP-1999-0120 (version of 11/11/1999), Stanford Digital Library Technologies Project, 1999.

[23] Z. Pawlak. Rough classification. *International Journal of Human-Computer Studies*, 51:369–383, 1999.

[24] Z. Pawlak. Rough sets and intelligent data analysis. *Information Sciences*, 147:1–12, 2002.

[25] Z. Pawlak. Rough sets, decision algorithms and bayes' theorem. *European Journal of Operational Research*, 136:181–189, 2002.

[26] G. Piatetsky-Shapiro. Probabilistic data dependencies. In J. M. Zytkow, editor, *Proceeding of the ML-92 Workshop on Machine Discovery*, pages 11–17, Aberdeen, UK, 1992.

[27] J. Quinlan. *C4.5: Programs for machine learning*. San Mateo: Morgan Kaufmann, 1993.

[28] J. R. Quinlan. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.

[29] H. Zimmerman. *Fuzzy Set Theory and its Application*. Kluwer Academic Publishers, 2001.