

# On The Granulation Simplicity For The Decision Rule Discovery In Databases: EWI Vs. EFI

A Manuscript Submitted By

Chien-Hsing Wu, Ph.D.  
Associate Professor  
Department Of Information Management  
Kun Shan University Of Technology  
949 Da-Wan Road, Yung Kang, Tainan, Taiwan, ROC  
Tel: 886-6-2732726  
Fax: 886-6-2732726  
E-mail: wuch@mail.ksut.edu.tw

## ABSTRACT

*Most intelligent systems require the task of rule discovery in large databases to frequently maintain the current information, at the time the system is placed in service. Granulation is an important process that deals with the conversion of continuous contexts into linguistic ones as far as the rule discovery process is concerned. Existing unsupervised granulation techniques include Equal Width Interval (EWI) and Equal Frequency Interval (EFI). However, the performance difference with respect to the simplicity of the generated decision trees that these two granulation techniques present are hardly ever revealed. This paper opens up the performance in comparison with EWI to EFI via an empirical investigation where 18 real-life continuous datasets and the ID3 algorithm were utilized. Based on the result of nonparametric test, it was found that the EFI performed better than EWI. The research findings would be helpful in support of rule discovery for intelligent systems.*

**Keywords:** Rule discovery, Granulation, Simplicity, ID3

## 1. INTRODUCTION

The advent of modern information technology has been facilitating the use of the past data and concurrently enhancing the value of it in a multitude of applications. The survey conducted by Bose et al. [2001] showed that rule discovery is the most popular technique in the context of data reuse. Eliciting interesting and meaningful knowledge in databases for the modern intelligent systems has also captured an increased attention in the automated knowledge acquisition research community. Major techniques involved entail data management, information representation, and machine learning [Fayyad et al., 1996; Chen et al., 1996; Fayyad et al., 1997; Hirota et al., 1999]. Granulation (or discretization) deals with the conversion of continuous contexts to discrete ones in order for the mechanism employed to proceed with the discovering operation. It greatly affects the final products of rule discovery because of the linguistic representation function for numeric observed data. Many articles have presented various techniques of granulation [Catlett, 1991; Kerber, 1992; Holte, 1993; Quinlan, 1993; Dougherty et al., 1995; Liu, 1997]. They do present a meaningful contribution to the enhancement of rule discovery capability.

Granulation can be in general categorized into three different axes that include static to dynamic, global to local, and supervised to unsupervised [Dougherty et al., 1995]. A static technique assumes that all attributes are independent and consequently sets the number of bins as a constant [Catlett, 1991; Holte, 1993]. In a dynamic method, searching a relevant number of bins by looking at the interdependencies of attributes is performed. It is a technique that substantially needs domain experts' involvement. A global technique simultaneously considers all attributes and generates a conversion function for the entire dataset while a local method only restricts to single attribute [Kerber, 1992; Quinlan, 1993]. The class labels are taken into account while using the supervised technique [Catlett, 1991; Kerber, 1992; Quinlan, 1993; Wu et al., 1999]. This is a well-known entropy-based granulation method that is employed to improve the quality of discovering operation. In contrast, the unsupervised technique does not consider example labels while dividing the observed data space into a number of bins and therefore is regarded as a simpler method to granulate continuous attributes.

When dealing with granulation, a numeric data value will be eventually transformed into a linguistic term, no matter what axes a research focuses on. More

specifically, there must be a transformation function generated to convert continuous data into linguistic one before rule discovery mechanism can launch. For example, any observed numeric data that is less than 37.50°C in the domain of temperature of human body will be converted into ‘normal’, those that are greater or equal to 37.50 °C and less than 39.00 °C will be transformed into the linguistic term ‘high’, and those that are greater than or equal to 39.00 °C will be converted into ‘very high’. The transformation functions are  $T(x_i) = \text{‘normal’}$ , if  $x_i < 37.50$ ,  $T(x_i) = \text{‘high’}$ , if  $37.50 \leq x_i < 39.00$ ,  $T(x_i) = \text{‘very high’}$ , if  $x_i \geq 39.00$ ,  $i = 1, 2, 3, \dots, n$ , where  $x_i$  is the  $i^{\text{th}}$  numeric data and  $n$  is the size of a dataset, respectively.

A transformation function is basically used to compress an interval that theoretically contains infinite elements into a single linguistic term. The compression operation with respect to granulation of continuous contexts is necessary for two reasons. One is the preparation for a discovery mechanism to perform generation task and the other is the requirement of knowledge interpretation. From the technique’s point of view, it is found that almost existing granulation methods are utilized to gather numeric data that are similar, dependent, adjacent, or resemble into a group and to define a linguistic term to symbolize all members of it. Basically, these methods have presented a significant meaningful contribution to the rule discovery capability. However, when more deeply looking at the function of information representation, it is believed that a different technique will produce a different result with respect to its performance.

While a variety of approaches to granulation have shown that a noteworthy one performs better in some particular conditions, but not in all cases, this research focuses only on the unsupervised one. Existing methods include Equal Width Interval (EWI) and Equal Frequency Interval (EFI). The EWI is based on equal-size strategy while EFI chooses equal-volume.

While both techniques are being extensively utilized in the granulation process of rule discovery, their performance with respect to the simplicity of a generated decision tree is hardly ever disclosed, and therefore is the motivation of this study. An empirical investigation is conducted where 18 real-life datasets and the ID3 algorithm are utilized. This study also conducts a nonparametric test to derive conclusions in general.

The remainder of this paper is organized as follows. EWI and EFI information granularity and data transformation are described in section 2. Section 3 describes the employed ID3 rule discovery algorithm that is followed by an illustrated example. The comparison criterion with respect to simplicity is delineated in section 4. Section 5 provides the results of the empirical investigation. Final section addresses the concluding remarks and future research focuses.

## 2. GRANULATION VIA EWI AND EFI

### 2.1 EWI FOR INFORMATION GRANULARITY

EWI has often been applied as a conversion mechanism for producing nominal values from continuous ones. It involves sorting the observed values of a continuous attribute and dividing the range of observed values for the variable into  $n$  equally sized bins, where  $n$  is a parameter predefined by the user. If a variable  $x$  is observed to have values bounded by  $x_{\max}$  and  $x_{\min}$ , then this method computes the equalized bin width as  $(x_{\max} - x_{\min})/n$ . As a result, the set of granules can be expressed as  $R = \{R_1, R_2, \dots, R_n\}$ . The conversion function and the measure of participation strength are defined as follows. Figure 1 illustrates the data elements and information granularity. Each linguistic granule has the same continuous boundary, but the number of instances.

$$C_{EWI}(x_i) = \begin{cases} R_m, & \text{if } x_{\min} + (m-1)d \leq x_i < x_{\min} + md \\ R_n, & \text{if } x_i = x_{\max} \\ null & \text{otherwise} \end{cases}$$

where

$x_i$ : the  $i^{\text{th}}$  data.

$R_m$  : the  $m^{\text{th}}$  granule that  $y$  is grouped into,  $m = 1, \dots, n$ ,  $n$ : the number of granules.

$x_{\min}$ : the minimum of the data.

$x_{\max}$ : the maximum of the data.

$d$ : the equalized interval for  $n$  granules.

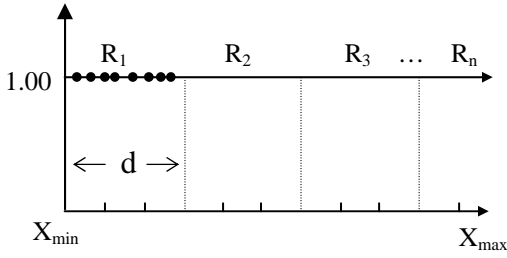


Figure 1: Information granularity via EWI

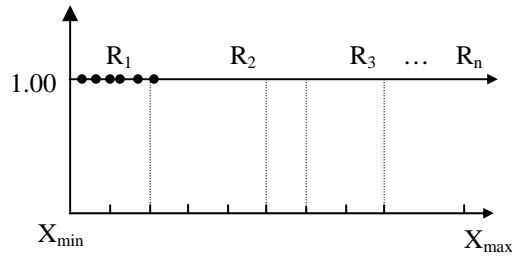


Figure 2: Information granularity via EFI

## 2.2 EFI FOR INFORMATION GRANULARITY

The EFI divides a continuous variable into  $k$  granules. Each granule contains  $m/k$  adjacent values (given  $m$  instances). It also involves sorting the observed values of a continuous feature and dividing its volume into  $n$  equally sized groups, where  $n$  is a user-defined parameter. If a variable  $x$  is observed to have  $N$  instances, the EFI returns the equalized volume as an integer portion of the expression of  $N/n$  for each linguistic granule. Note that the last granule may contain more than  $N/n$  instances if the returned

remainder is not 0. More specifically, the interval boundaries for an attribute of a sorted dataset are determined so that each contains approximately the same number of observed instances. Figure 2 illustrates the data elements and the information granularity. Each granule contains almost the same number of elements, except the interval boundaries. Similar to the EWI, the set of granules can be expressed as  $R = \{R_1, R_2, \dots, R_n\}$ . The conversion function is defined as follows. Note that the last granule may contain a different number of data if  $N$  is not aliquot with respect to the defined  $k$ .

$$C_{EFI}(x_i) = \begin{cases} R_m, & \text{if } (m-1)k < i \leq mk \\ R_n, & \text{if } mk < i \leq N \\ null & \text{otherwise} \end{cases}$$

where

$x_i$ : the  $i^{\text{th}}$  data.

$R_m$ : the  $m^{\text{th}}$  granule that  $x$  is grouped into,  $m = 1, 2, \dots, n$ ,  $n$  is the number of granules.

$k$ : the equalized volume for  $n$  granules.

$N$ : the total number of data.

## 2.3 NUMBER OF GRANULES

The number of granules for a context will greatly affect the information granularity to be defined and final decision rules to be discovered. However, there is a dilemma when making this decision. It is very likely that a granulated dataset contains numerous conflicting instances if it is too small. Note that the conflicting instances are those that have the same condition, except conclusions. If it is too large, a granule may not be able to contain acceptable number of instances, and consequently the discovered results may be too complex to interpret. More specifically, the decision tree to be generated using ID3 is too big to interpret patterns if labeling too many granules. More importantly, the discovered results have to be meaningful concerning performance evaluation. In spite of full freedom for an unsupervised method, the number

of granules should be carefully chosen to eliminate unnecessary problems.

This study does not focus on the optimization of the number of granules with respect to interpretability of a generated decision tree, but a selection of a suitable value of  $n$  by which the study can be empirically conducted in appropriateness. Therefore, the number of granules is determined by considering three conditions as follows: 1) a granule to be labeled can approximately contain more than or equal to 30 instances, 2) a dataset utilized must contain more than 90 instances, and 3) the maximum number of granules to be defined is 7. An algorithm (Binning Algorithm, BA) is a mechanism used to determine the number of granules is generated. It is expressed as follows. For example,  $n$  will be 3 if  $N$  is 100, 5 if 155, or 7 if 334.

**Begin**

**Let N be the number of instances**

**Let n be the number of granules**

**If N > 90**

**n = minimum(7, integer (N/30))**

**Else**

**Error: the dataset does not contain acceptable size of data**

**Terminate**

**Endif**

**End**

### 3. ID3 MECHANISM

The ID3 algorithm introduced by Quinlan [1986] is to help produce a tree that correctly classifies all examples in a given dataset. It is employed as the rule discovery

mechanism in this study. The information computation, expected information, and final gained information for a context A can be determined by formula (1), (2), and (3).

$$I(n_{c_1}, n_{c_2}, \dots, n_{c_n}) = \left(-\frac{n_{c_1}}{M} \log_2 \frac{n_{c_1}}{M}\right) + \dots + \left(-\frac{n_{c_n}}{M} \log_2 \frac{n_{c_n}}{M}\right) \quad (1)$$

$n_{C_i}$  : The number of records that return to class  $C_i$ ,  $i=1,2,\dots,n$ .

$M$  : The total number of records.

$$E(A) = \sum_{i=1}^t \left[ \left(\frac{n_{Vi}}{M}\right) I(a_{iC_1}, a_{iC_2}, \dots, a_{iC_m}) \right] \quad (2)$$

$t$  : The number of different values that attribute A can take on.

$n_{Vi}$  : The total number of records that attribute A takes on value  $V_i$ ,  $i=1,2,\dots,t$ .

$a_{iC_j}$  : the total number of records that attribute A takes on value  $V_i$  and returns to class  $C_j$ ,  $i=1,2,\dots,t$ ,  $j=1,2,\dots,m$ .

$M$  : The total number of records.

$$Gain(A) = I(n_{c_1}, n_{c_2}, \dots, n_{c_n}) - E(A) \quad (3)$$

#### An illustrative example

An artificial set of data collected from clinical records of liver diagnosis is included in Table 1. It is used to demonstrate the generation of decision rules in databases. The dataset contains information as follows:

1) 15 specific instances; 2) 3 attributes (Sgpt, Drinks, and Sgot) that are used to describe the domain; 3) a class that represents the diagnosis result (class attribute); 4) values that each attribute can take on are high, medium, adequate, and low; and 5) values that the class can take on are functional and dysfunctional.

Table 1: An artificial set of data from liver diagnosis

Record #	Sgpt	Drinks	Sgot	Class
1	high	high	medium	Dysfunctional
2	high	medium	high	Dysfunctional
3	high	medium	medium	Dysfunctional
4	high	adequate	low	Dysfunctional
5	high	adequate	adequate	Dysfunctional
6	medium	high	high	Dysfunctional
7	medium	high	low	Dysfunctional
8	medium	low	low	Functional
9	adequate	high	high	Dysfunctional
10	adequate	high	adequate	Dysfunctional
11	adequate	medium	high	Dysfunctional
12	adequate	medium	adequate	Functional
13	low	medium	high	Dysfunctional
14	low	medium	medium	Functional
15	low	low	adequate	Functional

Basically, all instances in Table 1 have to be taken into account because they all have to participate in the generation of decision rules. The discovery process starts partitioning via the value that the attribute takes on to form a decision tree, and consequently the decision rules can be generated [Sestito et al., 1994; Dhar et al., 2000]. Murphy [1998] presents a comprehensive process of generating a decision tree from a set of data. For example, when Sgpt takes on the value of “high”, all records are checked whether or not the value that the class takes on is the same. If so (e.g. all dysfunctional in Table 1), a rule “IF Sgpt = high THEN Class = dysfunctional with 4 supports and 1

condition” can be simply obtained. However, if the conclusion is inconsistent, the second order of attribute (e.g. Drinks) needs to be considered. This process has to repeat until the consistence is reached to return the decision rules. Notice that a single record is considered to be consistent. Figure 3 is the decision tree generated from Table 1, based on the attribute orders of Sgpt, Drinks, Sgot. The returned decision rules are listed in Table 2. The number of supports as well as the number of conditions for each generated rule is also included. Note that the number of supports of a rule is the number of evidences in the dataset.

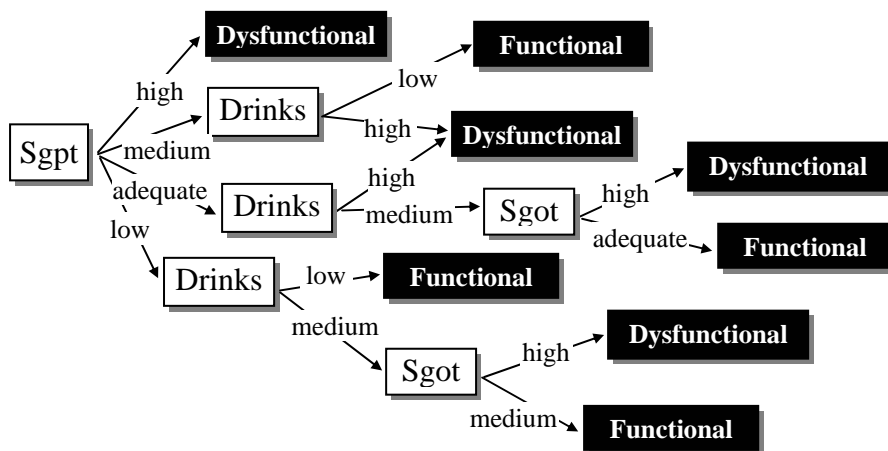


Figure 3: The generated decision tree from Table 1

Table 2: The generated decision rules from Figure 1

Rule #	Contents			Supports	Conditions	
1	IF	Sgpt=high	THEN	Class = Dysfunctional	5	1
2	IF	Sgpt=medium and Drinks=high	THEN	Class = Dysfunctional	2	2
3	IF	Sgpt=medium and Drinks=low	THEN	Class = Functional	1	2
4	IF	Sgpt=adequate and Drinks=high	THEN	Class = Dysfunctional	2	2
5	IF	Sgpt=adequate and Drinks=medium and Sgot=high	THEN	Class = Dysfunctional	1	3
6	IF	Sgpt=adequate and Drinks=medium and Sgot=adequate	THEN	Class = Functional	1	3
7	IF	Sgpt=low and Drinks=low	THEN	Class = Functional	1	3
8	IF	Sgpt=low and Drinks=medium and Sgot=high	THEN	Class = Dysfunctional	1	3
9	IF	Sgpt=low and Drinks=medium and Sgot=medium	THEN	Class = Functional	1	3

#### 4. MEASURE OF GAINED SIMPLICITY

The study then moves to the stage of evaluation. It is realized that the more supports and less conditions a rule has, the stronger it is [Wu et al., 1999]. The evaluation criterion selected for this study is the measure of gained simplicity that considers depth and width of a decision tree. In regard to a rule that has more conditions, the generated decision tree will go deeper. Similarly, given a fixed number of tuples, if a rule has more supports, the returned decision tree will be simpler. A dataset that contains 6 granulated tuples and 4 attributes as well as a class attribute as an

example is used to express this concept in a simpler way. Assume that 3 techniques, denoted T1, T2, and T3, (granulation and/or discovering mechanism) are employed to discover decision rules. The results are given in Table 3 where various numbers of rules with various conditions and supports are included. The notation of (r1, r2, r3, r4) gives the information as follows. 1) the number of rules generated by T1 and T2 is 4; 2) T3 returns 3 rules; 3) the number of supports for a rule that has 1 condition is denoted as r1, 2 conditions as r2, 3 conditions as r3, and 4 conditions as r4. Figure 4 illustrates this manner with a graphical representation for Table 3.

Table 3: The results from T1, T2, and

Techniques	Generated rules	Gained simplicity
T1	(2, 1, 1, 2)	3.3333
T2	(1, 2, 2, 1)	2.9167
T3	(0, 2, 2, 2)	2.1667

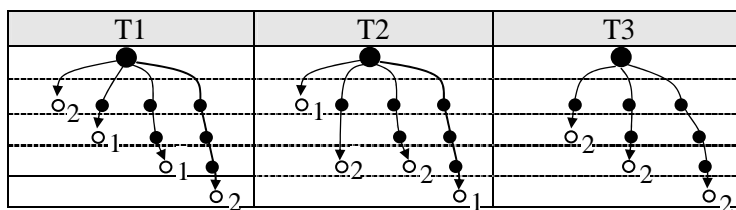


Figure 4: The decision trees via T1, T2, and T3

The gained simplicity of a decision tree is regarded as a function of the number of conditions and the number of supports for the generated rules. It is expressed as formula (4).

$$S_T = \sum_{i=1}^k \left( \frac{1}{n_i} \right) (m_i) \quad (4)$$

where

$S_T$ : the total gained simplicity of a generated decision tree.

$n_i$ : the number of conditions of the  $i^{\text{th}}$  rule,  $i=1, 2, \dots, k$ .

$m_i$ : the number of supports of the  $i^{\text{th}}$  rule,  $i=1, 2, \dots, k$ .

As a result, the gained simplicity of the decision trees generated by T1, T2, and T3 is computed as follows. It

is concluded that T1 returns *simpler* results.

$$S_{T1} = \frac{1}{1} * 2 + \frac{1}{2} * 1 + \frac{1}{3} * 1 + \frac{1}{4} * 2 = 3 \frac{1}{3} = 3.3333$$

$$S_{T2} = \frac{1}{1} * 1 + \frac{1}{2} * 2 + \frac{1}{3} * 2 + \frac{1}{4} * 1 = 2 \frac{11}{12} = 2.9167$$

$$S_{T3} = \frac{1}{1} * 0 + \frac{1}{2} * 2 + \frac{1}{3} * 2 + \frac{1}{4} * 2 = 2 \frac{1}{6} = 2.1667$$

We develop an algorithm (Computing Total Rule Simplicity Algorithm, CTRSA) expressed as follows to

compute the total gained simplicity

**Begin**

**Load a table that contains generated rules**

**Go top**

**Initialize the total gained simplicity denoted as totalGainSmp to be zero**

**Do while not end of file**

**totalGainSmp = totalGainSmp + (1/conditions) \* supports**

**skip**

**Enddo**

**End**

## 5. EMPIRICAL INVESTIGATION

### 5.1 CHARACTERISTICS OF THE EXPERIMENT

The datasets used as the test example were collected from the machine-learning repository cited in <http://www.ics.uci.edu/~mlearn> [Murphy et al., 1994].

The characteristics of experiment for this study are summarized in Table 3 showing the number of datasets used, data type, rule used to deal with missing data, granulation techniques, binning mechanism, discovery mechanism, evaluation criterion, and the experiment objective.

Table 3: Characteristics of the experiment

Items	Characteristics
The number of datasets	18
Data type	All real-life and continuous
Missing data	Be eliminated
Granulation techniques employed	EWI, EFI
Binning rule	BA
Discovery mechanism utilized	ID3
Evaluation criterion	Total gained simplicity by CTRSA
Experiment objective	Simplicity comparison for EWI and EFI

## 5.2 EMPIRICAL RESULTS

The result with respect to the gained simplicity is listed in Table 4. It was found that of the 18 datasets used, 15 via EFI obtained the bigger gained simplicity. However, in order to compare the performance of these two techniques in general, the matched-pairs signed rank test of significance of difference was conducted. [Hamburg, 1991; Nasipuri et al., 1997]. Basically, the

test hypothesis for a small size of samples ( $n < 30$ ) in the test advocates that the population positive and negative differences are symmetrically distributed about a mean of zero. As a result, at the 95% of level of significance, the test result confirmed that EFI performed better than EWI. This implies that EFI can help produce a more concise decision tree for a dataset

Table 4: The characteristics of datasets used and the results of simplicity

Name	Size	No_attr.	No_class	Total gained simplicity	
				EWI	EFI
BC198	194	33	2	76.7187	79.1856*
BC569	569	30	2	288.7177*	282.7245
BC699	699	9	2	277.8240	387.1358*
Bupa	345	6	2	55.3667	108.2666*
Glass	214	9	7	35.9059	135.4500*
Iris	150	4	3	97.6667	117.5667*
Letter	16384	16	26	2479.1380	2914.9140*
Pageblock	5473	10	5	381.9286	2045.906*
Pendigit	3498	16	10	685.0004	738.8256*
Satellite	2000	36	6	558.0693*	434.2257
Segmentation	210	16	7	61.1806	79.1607*
Shuttle	14500	9	7	1957.3190	6079.205*
Sonar	208	60	2	55.5485	64.1127*
Synthetic	600	61	6	136.8064*	129.2234
Vehicle	864	18	4	117.3573	131.9958*
Vowel	900	10	11	262.3219	279.9459*
Waveform	5000	21	3	795.8637	955.8507*
Wine	178	13	3	60.0000	74.4167*

\* the larger total gained simplicity

## 6. CONCLUDING REMARKS

This paper has briefly described two granulation techniques, EWI and EFI, to assist with the transformation of continuous attributes to discrete ones, addressed the simplicity measure for the selected evaluation criterion, and conducted an empirical investigation in comparison of EWI to EFI by using 18 real life datasets. The experimental result showed that EFI significantly performed better than EWI. It is almost impossible to derive all implications via a single study. Issues such as the effect of EWI and EFI on other mining techniques, comparison to statistical techniques, and criteria used to evaluate the outputs can be the future research focuses. Decision rule discovery is indisputably a costly and time-consuming task. Existing tools can only carry out the techniques that have been explored in the preprocessing or discovering stages. New ideas to be conducted require many efforts to obtain the results. It is believed that a methodology that can carry out the whole task ranging from data preprocessing, knowledge elicitation, and performance evaluation would be beneficial to the knowledge discovery endeavor.

## 7. REFERENCES

- Bose, I. and Mahapatra, R.K. (2001), "Business data mining – a machine learning perspective", *Information & Management*, 39, 211-225.
- Catlett, J., (1991), "On Changing Continuous Attributes into Ordered Discrete Attributes," *Proceedings of the European Working Session on Learning*, Berlin, Germany, pp.164-178.
- Chen, M.S., Han, J. and Yu, P.S., (1996), "Data Mining: An Overview from a Database Perspective", *IEEE Transaction on Knowledge and Data Engineering*, Vol. 8, pp. 866-883.
- Dougherty, J., Kohavi, R. and Sahami, M., (1995), "Supervised and Unsupervised Discretization of Continuous Features," *Proceedings of 1995 International Conference on Machine Learning*, pp.194-202.
- Fayyad, U. M, Piatetsky-Shapiro, G. and Smyth, P. (1996), "From Data Mining to Knowledge Discovery in Databases," *AI Magazine*, Vol. 17, pp.37-54.

Fayyad, U.M. and Stolorz, P., (1997), "Data Mining and KDD: Promise and Challenges," *Future Generation Computer Systems*, Vol. 13, No. 2-3, pp. 99-115.

Hamburg, M., (1991), *Statistical Analysis for Decision Making*, Harcourt Brace Jovanovich Inc., New York, pp. 656-658.

Hirota, K. and Pedrycz, W., (1999), "Fuzzy Computing for Data Mining", *Proceedings of The IEEE*, 87, 1575-1600.

Holte, R.C., (1993), "Very Simple Classification Rules Perform Well on Most Commonly Used Datasets," *Machine Learning*, Vol. 11, pp. 63-90.

Kerber, R., (1992), "ChiMerge: Discretization of Numeric Attributes," *Proceedings of the 10<sup>th</sup> National Conference on Artificial Intelligence*, AAAI Press/MIT Press, Menlo Park, pp. 123-128.

Liu, H. and Setiono, R., (1997), "Feature Selection via Discretization," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9, No. 4, pp. 642-646.

Murphy, P.M. and Aha, D.W., (1994), *UCI Repository of Machine Learning Databases*, <http://www.ics.uci.edu/~mlearn/>.

Nasipuri, A. and Tantarantana, S., (1997), "Nonparametric distributed detector using Wilcoxon statistics," *Signal Processing*, Vol. 57, No. 2, pp. 139-146.

Quinlan, J.R., (1986), "Induction of decision tree," *Machine Learning*, Vol. 1, pp. 81-106.

Quinlan, J.R., (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann, California.

Wu, X. and Urpani, D., (1999), "Induction By Attribute Elimination," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, No. 5, pp. 808-812.