ON APPLICATIONS OF ROUGH SETS THEORY TO KNOWLEDGE DISCOVERY

by

Frida R. Coaquira Nina

A thesis submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY
in
COMPUTING AND INFORMATION SCIENCES AND ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2007

Approved by:

Date

Edgar Acuña, Ph.D.
President, Graduate Committee

Raúl Macchiavelli, Ph.D. Date
Member, Graduate Committee

Tokuji Saito, Ph.D. Date
Member, Graduate Committee

Fernando Vega, Ph.D. Date
Member, Graduate Committee

Mario Córdova, Ph.D. Date
Representative, Office of Graduate Studies

Nestor Rodriguez, Ph.D. Date
Director, CISE Doctoral Program

ABSTRACT

Knowledge Discovery in Databases (KDD) is the nontrivial extraction of implicit, previously unknown and potentially useful information from data. Data preprocessing is a step of the KDD process that reduces the complexity of the data and offers better conditions to subsequent analysis. Rough sets theory, where sets are approximated using elementary sets, is another approach for developing methods for the KDD process.

In this doctoral Thesis, we propose new algorithms based on Rough sets theory for three data preprocessing steps: Discretization, feature selection, and instance selection. In Discretization, continuous features are transformed into new categorical features. This is required for some KDD algorithms working strictly with categorical features. In Feature selection, the new subset of features leads to a new dataset of lower dimension, where it is easier to perform a KDD task. When a dataset is very large, an instance selection process is required to decrease the computational complexity of the KDD process. In addition to that, we combine a partitioning clustering algorithm with the Rough sets approach obtaining comparable results to a hierarchical clustering algorithm used along with rough sets.

The new methods proposed in this thesis have been tested on datasets taken from the Machine Learning Database Repository at the University of California at Irvine.

RESUMEN

Descubrimiento del conocimiento en bases de datos (KDD por sus siglas en inglés) trata de la extracción no trivial de conocimiento implícito y de información muy útil previamente desconocida, que se obtiene a partir de los datos. El preprocesamiento de datos es un paso dentro del proceso de KDD, que reduce la complejidad del conjunto de datos y ofrece mejores condiciones para análisis posteriores. La teoría de “Rough sets”, en donde conjuntos son aproximados por conjuntos elementales. es otra manera de desarrollar los métodos para el proceso KDD.

En esta tesis doctoral se proponen nuevos algoritmos basados en teoría de “Rough sets” para tres etapas del preprocesamiento de datos: discretización, selección de variables y selección de casos. En discretización, las variables continuas son transformadas en nuevas variables categóricas. Este proceso es necesario en muchas tareas de KDD, dado que algunos algoritmos están diseñados para trabajar sólo con datos de tipo categórico. En selección de variables, las variables seleccionadas darán origen a un conjunto de datos de menor dimensión, en donde se ejecutarán las tareas de KDD más fácilmente. Cuando el tamaño del conjunto de datos es muy grande, se requiere un proceso de selección de instancias para reducir la complejidad computacional del proceso KDD. En esta tesis, se seleccionan los mejores casos de un conjunto de datos desde una perspectiva de la teoría de “Rough sets”. Adicionalmente, combinamos un algoritmo de conglomerados usando particionamiento con la teoría “Rough sets” y obtenemos resultados comparables a los obtenidos usando un conglomerado jerárquico junto con “Rough sets”. Los nuevos métodos propuestos en esta tesis han sido probados utilizando conjuntos de datos tomados del “Machine Learning Database Repository” disponible en la Universidad de California en Irvine.

.

To my family.

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Edgar Acuña for his collaboration and suggestions in this work.

To my graduate committee members: .Dr. Tokuji Saito, Dr. Raúl Macchiavelli, Dr. Fernando Vega and Dr. Mario Cordova, for their valuable suggestions.

To the Doctoral program CISE (Computation and Information Science and Engineering) of the University of Puerto Rico, at Mayagüez Campus.

To the CASTLE research group of the Mathematical Sciences Department.

To the administrative staff and faculty members of the Mathematical Sciences Department.

To my family for supporting me constantly in every step of my life, and to my friends for making my stay in Puerto Rico more pleasant.

I would like also to thank the Office of Naval Research (ONR) – Grant (N00014-03-1-0359) for their financial support to this project.

To the Mathematical Sciences Department for giving me financial support through the teaching assistant program.

TABLE OF CONTENTS

List of Tables ….…………………………………………………...……….....……...x
List of Figures ……………..…………………………………………...…..……........ix

1. INTRODUCTION……………………………………………………...…..…..1

    1.1. Classifiers ………………………………………………………………..…..3

    1.2. Objectives of the thesis……………………………………………….…..…..4

    1.3. Thesis structure……………………………………..…………………...5

2. ROUGH SETS……………………………………………………..….........….7

    2.1. Introduction………………………………………………………..………7

    2.2. Rough sets Concepts ……………………………………………....……..…7

        Equivalence Relation……………………………………………..……...7

        Partition……………………………………………………………………..…8

        Indiscernability relation……………………………………………….....8

        Lower Approximation of a subset…………………………………..……...….9

        Upper Approximation of a subset………………………….………………..…..9

        Boundary Region………………………………………………………….10

        Rough set…………………………………………………………………….10

        Crisp………………………………………………………………………….10

        Positive region of a subset………………………………………..…...….10

        Negative region of a subset………………………………………......…10

        Discernibility Matrix…………………………………………….………..11

        Dispensable and Indispensable features………………..……………………....…12

        Reduct ……………………………………….………………………..….12

        The Core…………………………………………..……………….……...…14

        Dependency coefficient…………………...16

Accuracy of the approximation……………..………………………..…………...17

Dependency relation matrix………………..…………………………..…..…..18

2.3. Rule discovery based on rough set theory……..……………………..……...…19

2.4. Relationship between Fuzzy sets and Rough sets………..……………..……..……19

2.5. Exploring Results with Rough sets theory………………..…………….…...…..…..21

3. DISCRETIZATION…………………………………………..……….…….…...27

3.1. Introduction……………………………………………………………..........…27

3.2. Discretization methods……………………………………..…………..…...31

3.2.1. Equal width intervals……………………………………..…………..…...31

3.2.2. One R……………………………………………………….……....…32

3.2.3. Entropy……………………………………………………………..........32

3.2.4. Chi-merge………………………………………………………………..…..34

3.3. A Discretization methods based on Rough sets theory…………………...……..…..37

3.4. Results and Discussion ………………………………………………...……...38

3.5. Conclusions…………………………………….……………………….……......40

4. FEATURE SELECTION ……………………………..……………………….…...42

4.1. Introduction……………………………………..………………………..….42

4.2. Filter Methods………………………………………..……………………...45

4.3. Wrapper methods…………………………………..………………………..45

4.4. Feature Selection method based on Rough sets theory……………..…………...…46

4.5. Feature selection by ranking according to the dependency measure…..……………49

4.6. An hybrid feature selection method……………………………………..………..50

4.7. Results and Discussion ………………………………………………..……51

4.8. Conclusions……………………………………………………………..………...56

5. INSTANCE SELECTION…………………… ……………………………….……..…...….57

5.1. Introduction……………………………………………………………..…......57

5.2. Instance selection using Rough sets……………………………………..….59

5.3. A new algorithm for case selection using Rough Sets theory……………..……..60

5.4. Results and Discussion…………………………………………………….….…62

5.5. Conclusion……………………………………………………………….....…65

6. UNSUPERVISED LEARNING BASED ON ROUGH SETS………..……..……….…66

6.1. Introduction……………………………………………………..……….……..66

6.2. Rough sets based clustering algorithms ……………………………………..67

6.3. Dissimilarity measures…………………………………………………………68

6.4. Similarity measures ……………………………………..………..……….70

6.4.1. Dissimilarity matrix. ……………………..…………………. …………70

6.5. PAM(Partitioning Around Medoids) algorithm………..…………….……........71

6.6. A Rough sets-based Cluster Algorithm…………………………..……….........71

6.7. External Criteria Measures…..…………………………………..……..…...72

6.7.1. Rand Index………………..……………………………………….………....73

6.7.2. Jaccard Coefficient……………………………………………..….............73

6.7.3. Fowlkes and Mallow Index…………………………………………………..74

6.7.4. Hubert's Statistic………………………………………..…..…......…74

6.8. Results and Discussion …………………………………………………...….....74

6.9. Conclusion……………………………………………………..………..…......77

7. ETHICS……………………………………………………..………..…………..78

8. CONCLUSIONS……………………………………..……………….............80

9. FUTURE WORK…………………………………………….……………….......81

REFERENCES……………………………………………….…………….................82

APPENDIX A DATASETS DESCRIPTION……………………...……….……………….…90

APPENDIX B: DESCRIPTION OF R FUNCTIONS CREATED IN THIS THESIS….….92

LIST OF TABLES

Tables **Page**

Table 2.1 Example of reduct features ……………………………………….………..…..14
Table 2.2 Example of features that are not in Reduct……………………….……………14
Table 2.3 Example of a feature that belong to the core…….…………………….……...…15
Table 2.4 Example of a feature that does not belong to the core…..…………….………...…16
Table 2.5 Dataset…...…………………………………………………..………..…16
Table 2.6 The discernibility matrix corresponding to the dataset in table 2.5…..…..……....17
Table 2.7 Rules example…………………………………………………...……..…20
Table 2.8 Number of partitions generated for the Colon dataset according to the number of top genes considered………………..…………………………………..22
Table 2.9 The 15-subset partition of the Colon dataset and its respective representative Instance…………………………………………………………..….23
Table 2.10 Members of each of the fifteen partition sets with its respective class label.........24
Table 2.11 Partition sets without considering class label………………………............25
Table 3.1 Comparison of the number of cut points for feature using five discretization methods ……………………...……………...………………………..38
Table 3.2 Misclassification Error rate for the LDA classifier using discretized features…...39
Table 3.3 Comparison of the Misclassification Error rate for the KNN classifier using discretized features and without performing discretization ……………………..40
Table 4.1 Subsets of features selected using Rough sets criterion with four discretization, and features selected by two wrapper methods……….…………………….…52
Table 4.2 Comparison of Misclassification error rate for the LDA classifier with and without feature selection ………………………...……………………………...53
Table 4.3 Misclassification error rate for the KNN classifier after feature selection……..…53
Table 4.4 Subset of features selected using the hybrid method based on rough sets with three different discretization methods, and features selected by the SFS wrapper method along with the LDA classifier. ……………………………………………….54
Table 4.5 Subsets of features selected using the hybrid method based on rough sets with three different discretization methods and feature selected by the SFS wrapper method along with the KNN classifier. ……………………………………………….54
Table 4.6 Misclassification error rate for the LDA classifier after feature selection using the hybrid method……………………………………………………..…55
Table 4.7 Misclassification error rate for the KNN classifier after feature selection using the hybrid method……………………………………………………….55

Table 5.1 Misclassification error rate for the KNN classifier before and after case selection 60
Table 5.2 Misclassification error rate for the LDA classifier before and after case selection 63

Table 5.3 Misclassification error rate after ranking Feature selection for the KNN and LDA classifier before and after case selection considering Rough Set method discretization……………………..……………………..……………………….64
Table 5.4 Misclassification error rate after ranking Feature selection for the KNN and LDA classifier before and after case selection considering Rough Set method discretization. …………… ……………….……………………………………65
Table 6.1 External measures for Rough Cluster algorithm (PAM) ……………………..…..75
Table 6.2 External measures for Rough Cluster algorithm (HCLUST)………...…………...76
Table 6.3 External measures for Rough Cluster algorithm (PAM) Hamming distance……. 76

LIST OF FIGURES

Figure **Page**

Figure 1.1 The KDD process……….……………………………………….…………..…….3
Figure 2.1 Example of a Partition of a universe set U. ……………………………………….9
Figure 2.2 Representation of the data partitioning for a subset X………….………………..12
Figure 2.3 Algorithm to calculate the dependency ……………………………...………..18
Figure 2.4 Rough Set illustration considering three classes……………..…………......…19
Figure 2.5 Example of a fuzzy membership function…….…………………….……......21
Figure 3.1 Discretization algorithm based on rough sets……………………...……......…37
Figure 4.1 Feature Reduction Approach………………...………………….……………......42
Figure 4.2 Feature Selection Problem………...…………………………...………………44
Figure 4.3 The Quick Reduct algorithm…...…………………….……..……..……..48
Figure 4.4 Algorithm for feature selection based on Rough sets..…………..………………49
Figure 4.5 An hybrid feature selection algorithm……………………………………......…50
Figure 4.6 Dependency for each feature on Diabetes dataset……………………………..…51
Figure 4.7 Dependency for each feature on Glass dataset……………………………….…..51
Figure 5.1 Case Reduction in KDD…………………………………………...……..…58
Figure 5.2 Algorithm for case selection using Rough sets…………………..…..………..61
Figure 6.1 The Rough Cluster algorithm……………………….…………..…………..72

CHAPTER 1

INTRODUCTION

Rough sets theory was introduced by Z. Pawlak in 1982 [65] as a mathematical tool for data analysis. Since then it has been used to handle uncertain knowledge in Artificial Intelligence applications. Rough sets theory has many applications in the field of Knowledge Discovery in Databases (KDD) among them discretization [13, 15, 51, 76], feature selection [13, 17, 32], instance selection [14, 21,35, 54, 71], and clustering [49,50,83].

The vagueness and uncertainty of information can be seen as a property of sets imprecisely specified. Uncertainty can be attributed to set elements through the usage of the rough membership function, in a similar manner as the fuzzy membership function.
Fuzzy methods and Rough set methods are macroscopic, descriptive and numerical methods, but Fuzzy methods are deductive whereas Rough sets methods are inductive [60, 67].

Discretization is a data preprocessing task applied to datasets containing continuous features. This process is done prior to the application of several KDD methods. Discretization is an interactive process, and it is implemented based on the partitioning of the values of the continuous attributes. The dimension of a dataset can be reduced by eliminating irrelevant (dispensable) features. Rough sets can be used to find subsets of relevant (indispensable) features [18, 63].

Other application of rough sets is instance (case) selection. Equivalence relations can be found among several instances of the dataset and some of them can be selected to form a new subset to be used in future analyses. Thus, instance selection involves extracting elementary blocks from the dataset based on an equivalence relation.

In this thesis, we have conducted research on the application of rough sets theory in several Knowledge Discovery tasks. Thus, two feature selection methods based on Rough sets are proposed. The first one uses only the class label information to create the indiscernibility relation. Therefore it can be considered as a filter feature selection method. The second one is a hybrid method where a classifier needs to be used. We have developed an efficient discretization method using rough sets. An algorithm for instance selection, where a random sampling is performed on the positive region, has also been constructed. Finally, we have implemented a clustering method using the discernibility matrix as a similarity matrix. All the methods developed in this thesis have been applied to datasets coming from the Machine learning databases repository available at the University of California at Irvine (UCI).

Bioinformatics is an interesting area where many knowledge discovery tasks can be applied, among them, supervised and unsupervised classification (clustering). All methods developed using Rough sets theory may be applied to analyze gene expression data coming from microarray experiments[63]. These datasets have a very large number of features. In this thesis, a gene expression microarray data was used to illustrate concepts of Rough sets theory.

Fig 1.1, shows the steps of the KDD process. All the work done in this thesis, except clustering, is related to the data preprocessing step. Clustering is a data mining task.

1.1. Classifiers

A classifier is a decision rule constructed using the available data, called the training sample, and its goal is to assign classes to instances of a new dataset, called the test sample. The features (variables) of the dataset are used to design the classifier. Examples of classifiers are: linear discriminant analysis (LDA), k-nearest neighbors (KNN) classifiers, kernel density classifiers, decision trees, logistic regression classifiers, neural networks, support vector machine, etc.

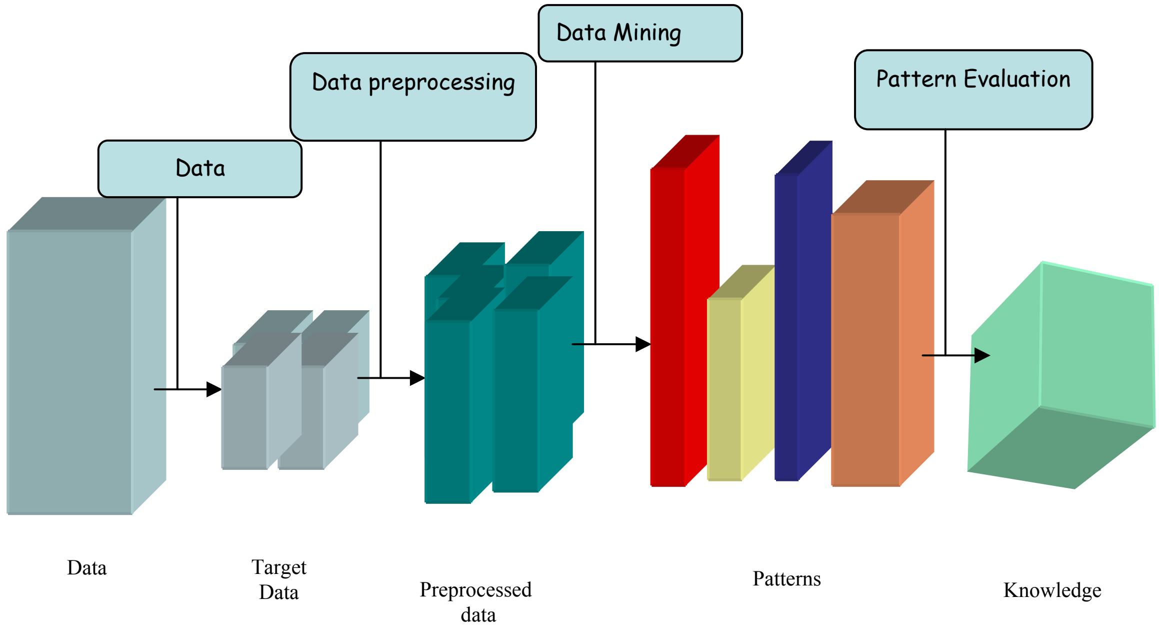

Figure 1.1. The KDD process.

The proportion of future instances assigned to incorrect classes for the classifier is called the misclassification error rate. This error can be estimated using the training and test data sets. K- fold cross validation is the most common method used to estimate the misclassification error rate. A combination of multiple classifiers, called an ensemble, is sometimes used to improve the performance of a single classifier. Bagging and Boosting are examples of ensembles.

A brief description of the classifiers to be used in this thesis is given below:

The Linear discriminant analysis (LDA) classifier. It is a commonly used parametric classification method that assumes multivariate normality for the features in each of the classes, and equal covariance matrices of the features between the classes. This method maximizes the ratio of the between-class variance to the within-class variance in any particular dataset, thereby guaranteeing maximum separability. LDA tries to draw a decision region between the classes of the dataset using the rule: assign the instance $\boldsymbol{x}$ to the class *j,* which has the closest mean to $\boldsymbol{x}$.

The k-nearest neighbors (KNN) classifier. This is a non-linear and nonparametric classifier. The KNN classifier finds for each instance of the dataset, the k-nearest instances using a distance measure, and its classification is decided by majority vote, with ties broken at random. If there are ties for the k-th nearest neighbor, all candidates are included in the vote.

1.2 Objectives of the thesis.

In this research we have accomplished the following objectives:

1. Develop an efficient discretization method using Rough sets theory.
2. Find efficient feature selection algorithms based on Rough sets in a supervised classification context.
3. Build a new algorithm for choosing the best instances of a dataset in order to carry out Knowledge Discovery in an efficient way.
4. Implement a partitioning clustering algorithm based on Rough sets theory.

The algorithms for discretization, feature selection, and case selection based on rough sets are used along with the LDA and KNN classifiers to evaluate their effect on the misclassification error.

Rough set analysis methodology in KDD tasks can be applied only to datasets containing features with nominal values. Therefore, data discretization is required for datasets containing features with continuous values. We compare our proposed discretization methods based on Rough sets theory with other existing discretization techniques such as: Chi-Merge, Entropy, Equal width binning, and 1R.

1.3 Thesis structure

The structure of this thesis is as follows:

Chapter 1: It contains an introduction to the work done in our thesis. A brief description of the classifiers used in the thesis is given. Finally the thesis' objectives are detailed.

Chapter 2: This chapter describes some basic concepts related to Rough sets theory. Definitions involving Rough sets are detailed, and illustrated with examples. Some graphics are used to illustrate the concepts in a friendly way. Finally, Rough sets concepts are applied to two real data sets, and the results are analyzed.

Chapter 3: We describe how Rough sets theory can be used to select the cut points in the discretization process. Some KDD algorithms are designed only for non-continuous features, but there are many datasets containing different types of features: binary, nominal, ordinal, and continuous. Therefore, we are interested in the discretization process of continuous features. First, we use the Scott's criterion to determine the upper bound for the number of intervals defined by the cut points, and then we find the optimal cut points using our proposed algorithm.

Chapter 4: It contains a description of the feature selection problem. Filter and wrapper feature selection algorithms most commonly used are explained briefly. Two algorithms using Rough sets criteria are proposed and some results on real data sets are shown. Finally, the results are discussed and conclusions are given.

Chapter 5: The instance selection problem in the KDD process is described. We use Rough sets criteria to carry out this task, and an algorithm for this purpose is implemented and applied to real datasets.

Chapter 6: It includes a description of the clustering problem. Rough sets theory is used to build a partitioning clustering algorithm based on the robust PAM algorithm. The clusters formed are evaluated using some external measures. A comparison of these measures with those obtained from hierarchical clustering based on Rough sets is carried out. This chapter closes with a discussion of the experimental results.

Chapter 7: This chapter contains ethical aspects related to KDD methods and Rough sets theory. Some problems for the analysis and interpretation of the results are considered from an ethical point of view.

Chapter 8: This chapter presents conclusions of the findings obtained in this thesis.

Chapter 9: Future work using the concepts of rough sets on KDD methods that may be investigated is detailed.

CHAPTER 2

ROUGH SETS

2.1 Introduction

Rough sets theory was proposed by Z. Pawlak (1982) [65]. Rough sets makes an approximation of sets using a collection of elementary sets. Methodology based on Rough set does not require external parameters to analyze data and to draw conclusions from them. It offers many opportunities for developing Knowledge Discovery methods using partition properties and the discernibility matrix.

Rough sets theory provides a mathematical tool that can be used to find out all possible feature subsets [55, 56, 65, 66]. In the feature selection problem the principal idea is to recognize the dispensable and indispensable features, using the discernibility matrix [59, 66, 93]. The purpose of using Rough sets is to find the **Core,** that is, the set of all indispensable features.

2.2 Rough sets concepts

In this section, we will define some concepts related to Rough sets theory.

Definition 1. Let U be a non-empty set and let x, y, and z be elements of U. Consider R such that xRy if and only if (x,y) is in R. R is an **equivalence relation** if it satisfies the following proporties:

i) Reflexive Property: (x, x) is in R for all x in U.
ii) Symmetric Property: if (x, y) is in R, then (y, x) is in R.
iii) Transitive Property: if (x, y) and (y, z) are in R, then (x, z) is in R.

Definition 2. A **partition** *P* of U is a family of nonempty subsets of U such that each element of U is contained in exactly one element of *P*.

i) $U = \bigcup_{i=1}^{n} U_i$,

ii) $U_i \bigcap U_j = \phi$, for all $i \neq j$

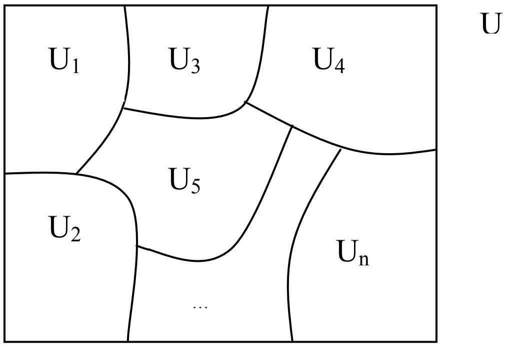

Fig 2.1. Example of a Partition of a universe set U.

Definition 3. The Indiscernibility relation

Rough sets theory is based on the Indiscernibility relation. Let $T = (U, A, C, D)$ be a decision system data, where U is a non-empty finite set called the universe, A is a set of features, C and D are subsets of A, named the conditional and decisional attributes subsets respectively. The elements of U are called objects, cases, instances or observations. Attributes are interpreted as features, variables or characteristics conditions. Given a feature *a*, such that:
$a : U \rightarrow V_a$ for $a \in A$, V_a is called the value set of a.

Let $a \in A$, $P \subseteq A$, the **indiscernibility relation** *IND*(*P*), is defined as follows:

$$IND(P) = \{(x, y) \in U \times U : \textit{for all} \;\; a \in P,\; a(x) = a(y)\}$$

In simple words, two objects are indiscernible if we can not discern between them, because they do not differ enough.

The indiscernibility relation defines a *partition* in U. Let $U/IND(P)$ denote a family of all equivalence classes of the relation $IND(P)$, called elementary sets. Two other equivalence classes $U/IND(C)$ and $U/IND(D)$, called conditional and decisional classes respectively, can also be defined.

The decisional attribute D determines the decisional classes $U / IND(D) = \{X_1, X_2, ..., X_{r(D)}\}$ of the universe U, where $X_k = \{x \in U : D(x) = k\}$ for $1 \leq k \leq r(D)$ is called the *k-th* decisional class of decision system data T.

The equivalence classes of the discernibility relation, which are the minimal blocks of the information system, can be used to approximate these concepts, then a set X could be approximate using upper and lower approximation..

Definition 4. Lower approximation of a subset

Let $R \subseteq C$ and $X \subseteq U$, the R-lower approximation set of X, is the set of all elements of U which can be with certainty classified as elements of *X.*

$$\underline{R}X = \cup\{Y \in U / R : Y \subseteq X\}$$

According to this definition, we can see that R-Lower approximation is a subset of X, thus $\underline{R}X \subseteq X$.

Definition 5. Upper approximation of a subset

The R-upper approximation set of X is the set of all element of U, that can possibly belong to the subset of interest X.

$$\overline{R}X = \cup\{Y \in U / R : Y \cap X \neq \phi\}$$

Note that X is a subset of the R-upper approximation set, thus $X \subseteq \overline{R}X$.

Definition 6. Boundary Region.

It is the collection of elementary sets defined by:

$$BN(X) = \overline{R}X - \underline{R}X$$

These sets are included in R-Upper but not in R-Lower approximations.

Definition 7. A subset defined through its lower and upper approximations is called a **Rough set**. That is, when the boundary region is a non-empty set $(\overline{R}X \neq \underline{R}X)$.

Definition 8. A subset is called **Crisp** when its boundary region is empty ($\overline{R}X = \underline{R}X$).

Definition 9. Positive region of a subset

It is the set of all objects from the universe U which can be classified with certainty to classes of *U/D* employing attributes from *C*.

$$POS_C(D) = \underset{X \in U/D}{\cup} \underline{C}X$$

where $\underline{C}X$ denotes the lower approximation of the set X with respect to C. The **positive region** of the subset *X* belonging to the partition *U/D* is also called the **lower approximation** of the set *X*. The positive region of a decision attribute with respect to a subset C represents approximately the quality of C.

The union of the positive and the boundary regions constitutes the **upper approximation**. The upper approximation contains all data that can possibly be classified as belonging to the set *X* (see Fig. 2.2).

Definition 10. Negative region of a subset

The negative region consists of those elementary sets that have no predictive power for a subset X given a concept R. They consist of all classes that have no overlap with the concept. Thus is,

$$NEG_R X = U - \overline{R}X$$

Figure 2.2, illustrates the lower and upper approximations and the boundary region corresponding to the set X.

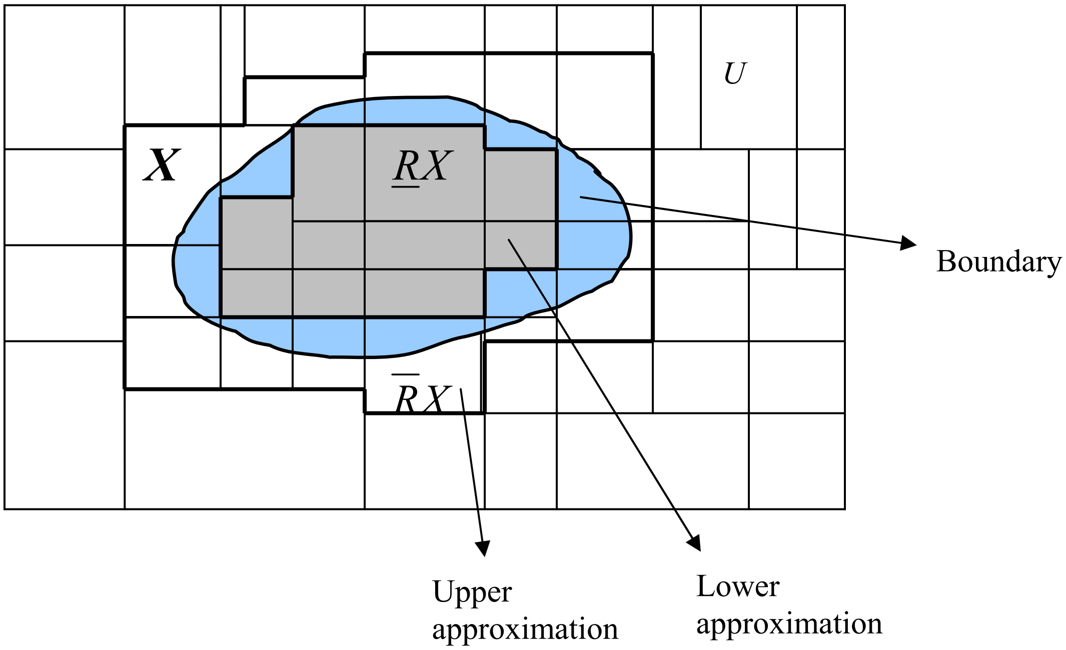

Figure 2.2. Representation of the data partitioning for a subset X.

Definition 11. The Discernibility Matrix

Let U ={x_1, x_2, x_3, … x_n} be the universe on a decision table. The *Discernibility* matrix is defined by: $m_{ij} = \{a \in C : (a(x_i) \neq a(x_j)) \wedge (d(x_i) \neq d(x_j), d \in D)\}$ for $i, j = 1,2,3,...,n$

where, m_{ij} is the set of all attributes that classify objects x_i and x_j into different decision classes in U / D partition.

Definition 12. Dispensable and Indispensable Features

Every dataset contains conditional and decision features. Some of these features are indispensable which are very important in the analysis [31, 93]. The problem of feature selection is searching for indispensable features and eliminating the dispensable features.

Let $c \in C$, C is the set of conditional features. A feature *c* is *dispensable* in the information dataset T if $POS_{\{C-\{c\}\}}(D) = POS_C(D)$; otherwise feature c is indispensable in T and should be considered in the final best subset of feature. The main purpose in the feature selection process is to retain all indispensable features that cause the decision system data T to be consistent [93]. Thus, if c is an indispensable feature, deleting it from T will cause T to be inconsistent. In the other hand, if a feature is dispensable, it could be eliminated from the dataset and in this way the dimensionality of the dataset will be reduced [55].

Definition 13. Reduct

A system $T = (U, A, C, D)$ is independent if all c in C are indispensable. A set of features $R \subseteq C$ is called the reduct of C if $T' = (U, A, R, D)$ is independent and $POS_R(D) = POS_C(D)$. Furthermore, there is not $T \subset R$ such that

$$POS_T(D) = POS_C(D)$$

A Reduct is a minimal set of features that preserves the indiscernibility relation produced by a partition of C. There could be several subsets of attributes like R. Similar or indiscernible objects may be represented several times on an information table, some of the attributes may be superfluous or irrelevant, and they could be removed without loss of classification performance.

Table 2.1 shows that features a_1 and a_2 simultaneously classify well the instances into the classes of D, therefore $\{a_1, a_2\}$ is a Reduct of $C = \{a_1, a_2, a_3, a_4\}$.

Table 2.1. Example of reduct features.

U	D	a_1	a_2	a_3	A_4
1	1	3	1	1	2
2	1	3	1	1	2
3	1	3	1	1	2
4	2	2	3	1	1
5	2	2	3	1	1
6	2	2	1	2	2
7	2	1	2	2	3
8	3	1	1	2	2

Table 2.2 shows features a_1 and a_3 jointly do not classify well, since an instance with values a_1=1 and a_3=2 may belong to either class 2 or class 3. Therefore, $\{a_1, a_3\}$ is not a Reduct of $C = \{a_1, a_2, a_3, a_4\}$.

Table 2.2. Example of features that are not in Reduct.

U	D	a1	a2	a3	a4
1	1	3	1	1	2
2	1	3	1	1	2
3	1	3	1	1	2
4	2	2	3	1	1
5	2	2	3	1	1
6	2	2	1	2	2
7	2	1	2	2	3
8	3	1	1	2	2

Definition 14. The Core

The set of all the features indispensable in C is denoted by CORE(C). The Core is the set of all single element entries of the discernibility matrix, that is,

$$CORE(C) = \{a \in C : m_{ij} = \{a\} \text{ for some } i, j\}.$$

We have

$$CORE(C) = \bigcap RED(C)$$

where $RED(C)$ is the set of all reducts of C. Thus, the Core is the intersection of all reducts of an information system. The Core does not consider the dispensable features and it can be expanded using Reducts. The feature subset obtained is good enough to make information induction.

Table 2.3 shows that instances 6 and 8 are ambiguous upon removal of a_1. Hence a_1 should be part of the Core.

Table 2.3. Example of a feature that belong to the Core

U	D	a1	a2	a3	a4
1	1	3	1	1	2
2	1	3	1	1	2
3	1	3	1	1	2
4	2	2	3	1	1
5	2	2	3	1	1
6	2	2	1	2	2
7	2	1	2	2	3
8	3	1	1	2	2

Table 2.4 shows that there are no ambiguous observations upon removal of a_2, and hence a_2 is not part of the Core.

Table 2.4. Example of a feature that does not belong to the Core

U	D	a_1	a_2	a_3	a_4
1	1	3	1	1	2
2	1	3	1	1	2
3	1	3	1	1	2
4	2	2	3	1	1
5	2	2	3	1	1
6	2	2	1	2	2
7	2	1	2	2	3
8	3	1	1	2	2

Example of a Core: Consider the dataset given in table 2.5, where $U = \{x_1, x_2, ..., x_7\}$ is the universe set, $C = \{a_1, a_2, a_3, a_4\}$ is the conditional features set, and $D = \{0,1,2\}$ is the decision features set.

Table 2.5. The Dataset

	a_1	a_2	a_3	a_4	D
x_1	1	0	2	1	1
x_2	1	0	2	0	1
x_3	1	2	0	0	2
x_4	1	2	2	1	0
x_5	2	1	0	0	2
x_6	2	1	1	0	2
x_7	2	1	2	1	1

The corresponding Discernibility matrix is as follows:

Table 2.6. The discernibility matrix corresponding to the dataset in table 2.5.

	x_1	x_2	x_3	x_4	x_5	x_6
x_2	-					
x_3	$\{a_2, a_3, a_4\}$	$\{a_2, a_3\}$				
x_4	$\{a_2\}$	$\{a_2, a_4\}$	$\{a_3, a_4\}$			
x_5	$\{a_1, a_2, a_3, a_4\}$	$\{a_1, a_2, a_3\}$	-	$\{a_1, a_2, a_3, a_4\}$		
x_6	$\{a_1, a_2, a_3, a_4\}$	$\{a_1, a_2, a_3\}$	-	$\{a_1, a_2, a_3, a_4\}$	-	
x_7	-	-	$\{a_1, a_2, a_3, a_4\}$	$\{a_1, a_2\}$	$\{a_3, a_4\}$	$\{a_3, a_4\}$

Then, Core(C) = $\{a_2\}$. The partition produced by Core is

$$U/\{a_2\} = \{\{x_1, x_2\}, \{x_5, x_6, x_7\}, \{x_3, x_4\}\},$$

and the partition produced by the decision feature *d* is

$$U/\{d\}=\{\{x_4\}, \{x_1, x_2, x_7\}, \{x_3, x_5, x_6\}\}$$

Definition 15. The Dependency coefficient

Let $T = (U, A, C, D)$ be a decision table. The Dependency Coefficient between the conditional attribute C*,* and the decision attribute D is given by

$$\gamma(C,D) = \frac{card(POS(C,D))}{card(U)}$$

where, *card* indicates cardinality of a set. The dependency coefficient varies between 0 and 1, since it expresses the proportion of the objects correctly classified with respect to the total, considering the conditional features set. If γ=1, D depend totally on C, if 0<γ<1, the D depends partially on C, and if γ=0, then D does not depend on C. A decisional attribute depends on the set of conditional features if all values of decisional feature D are uniquely determined by values of conditional attributes. i.e. there exists a dependency between values of decisional and conditional features. An algorithm to calculate de Dependency coefficient is given below

```
i. Create the partition  of the  Dataset  D without considering the class feature.

ii.  Set Positive equal to zero, where Positive represents the cardinality of the Positive region.

iii. Search for Elementary sets that only belong to a unique class.
iv. For i=1 to the number of elementary sets

   If card(class(elementarySet[i] )) =1 then

        P = Card(elementarySet[i])

        Positive = positive + P

iv. Finally calculate dependency as follows:

        $Dependency = \frac{card(Positive)}{card(data)}$
```

Figure 2.3 Algorithm to calculate the dependency.

In the worst case the order of the algorithm is $O(n^2 \times p)$, where n is the number of instances and p is the number of attributes. Since the creation of the partition is of order $O(n^2p)$ and the computation of the positive is of order O(n) in the worst case.

Definition 16. Accuracy of the approximation

The accuracy of the approximation to the set X from the elementary subsets is measured as the ratio of the lower and the upper approximation size. The ratio is equal to 1, if no boundary region exists, which indicates a perfect classification. In this case, deterministic rules for the data classification can be generated.

$$\alpha(X) = \frac{Lower(X)}{Upper(X)}$$

Thus, a set X with accuracy equal to 1 is crisp. otherwise X is rough.

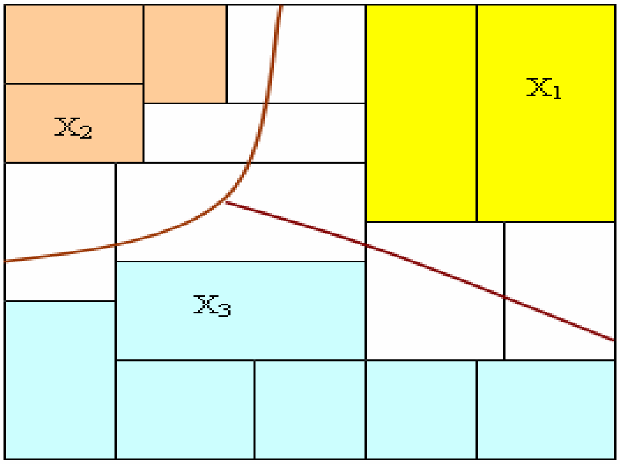

Fig 2.4 : Rough Set illustration considering three classes.

Definition 17. Dependency relation matrix

Given the information table, we can calculate the Dependency Matrix for each couple of feature a_i and a_j according to the class feature.

$$D(a_i, a_j \mid C) = \sum_{a_i, a_j, Y_c} \frac{\mid Pos_{a_i}^{Y_c}(a_j) \mid}{card(Y_c)}$$

$Pos_{a_i}^{Y_c}(a_j)$ represents the positive region of attribute a_j relative to attribute a_i within the class value y_c [63].

2.3 Rule discovery based on Rough sets theory

Rule discovery is an important problem since data relationships in the form "if A then B" do not necessarily reflect real rules of the application domain, and other problems could arise. Therefore, there is a need to eliminate incorrect rules.

Mitra[58] proposed some criteria to evaluate the created rules considering the correct classification percentage provided by the rules on a test set. The percentage of examples from a test set for which no rules are fired is used as a measurement of the uncovered region.

The construction of a rule discovery algorithm with estimated error rates of classification could be developed by the Rough sets theory [67, 80].

Example 1 . Given the dataset in Table 2.7 , then some decision rules that can be obtained appears in the table aside.

Table 2.7. Rules example

a1	a2	a3	a4	D
0	0	0	1	0
0	0	1	3	0
0	1	0	2	0
0	1	1	0	1
1	1	0	2	2

Rules
If(a2=0)=>(D=0)
If(a1=0)&(a4=2)=>(D=0)
If(a4=0)=>(D=1)
If(a1=1)=>(D=2)

2.4 Relationship between Fuzzy sets and Rough sets

Fuzzy sets introduced by Zadeh in 1965 and Rough sets introduced by Pawlak in 1982 are methods that can be viewed as representations of uncertainty regarding set membership.

Fuzzy sets use the membership function to give a degree of membership. A fuzzy set on a classical set X is defined as follows:

$$\widetilde{A} = \{(x, \mu_A(x)) \mid x \in X\}$$

$\mu_A(x)$ denotes the fuzzy membership function. A subset A is a fuzzy set when its membership in X is not crisp, but it is subject to gradation; formally this is expressed in the interval [0,1] by the fuzzy membership function. The membership function $\mu_A(x)$ quantifies the grade of membership of the elements x to the fundamental set X. An element mapping to the value 0 means that the member is not included in the given set, 1 describes a fully included member. Values strictly between 0 and 1 characterize the fuzzy members.

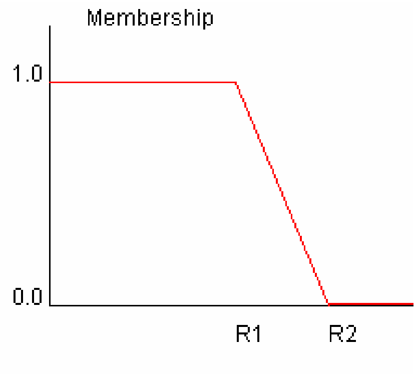

Figure 2.5. Example of a fuzzy membership function

Sometimes, a more general definition is used, where the membership function takes on values in an arbitrary fixed algebra or structure L. This generalization was first considered by Joseph Goguen (1967).

The fuzzy set B, where B={(3,0.3), (4,0.7), (5,1), (6,0.4)} would be enumerated as B={0.3/3, 0.7/4, 1/5, 0.4/6} using standard fuzzy notation. Note that any value with a membership grade of zero does not appear in the expression of the set. The standard notation for finding the membership grade of the fuzzy set B at 6 is $\mu_B(6) = 0.4$.

In Rough sets the equivalence classes generate the lower and upper approximations for a subset X. Rough sets theory does not work with crisp sets. A Crisp set has a clear cut point, hence does not reflect uncertainty about membership. For this reason, Crisp sets are used to formally characterize a concept. Rough sets are used to approximate sets. The rough membership function quantifies the degree or relative overlap between the set X and the equivalence class to which the current argument belongs to.

2.5 Exploring results with Rough sets theory

In this section, we apply the concepts described before to two datasets. The first one is coming from Bioinformatics and it contains gene expression data obtained from microarray experiment. The second one is a medical dataset which is very well known in the Machine Learning community.

Example 1. The Colon dataset

This well known dataset contains 2000 features (genes) and 62 instances classified in two classes: Normal (40 instances) and Tumor (22 instances). Before applying Rough sets theory the dataset is pre-processed. First, feature selection (gene selection) is performed using the Recursive Feature elimination method, which is available in the RFE library of R. After that, two discretization methods are applied: Entropy and Chi-Merge using the dprep library of R [2]. Data analysis using Rough sets theory is applied to the processed Colon dataset to calculate the lower approximation, upper approximation, and boundary regions.

Table 2.8. Number of partitions generated for the Colon dataset according to the number of top genes considered.

Number of top Genes	Number of partitions
50	61
30	51
20	36
10	19
5	15

The top five genes are 792, 1221, 1924, 1893 and 1060. After gene selection, only the Chi-Merge discretization method is used. Finally, the RSES [8] program is applied on the discretized data to obtain a partition with 15 subsets which are shown in Table 2.9.

Table 2.9. The 15-subset partition of the Colon dataset and its respective representative instance

	a_1	a_2	a_3	a_4	a_5	Y
1	2	1	1	1	1	1
2	1	1	2	1	1	1
3	1	2	2	1	1	1
4	2	1	2	1	1	1
5	2	2	2	1	1	1
6	1	1	1	1	2	1
7	1	1	1	1	1	1
8	1	2	1	1	2	1
9	1	2	1	1	2	2
10	1	2	1	1	1	2
11	1	2	2	1	2	2
12	1	1	1	1	1	2
13	2	2	1	1	2	2
14	1	1	1	1	2	2
15	2	1	1	1	2	2

Different colors are used to indicate that there exists relationships among the rows.

Table 2.10 shows the instances in each subset of the partition. The representative instances of each set of the partition appear in the last row of each cell, as a 5-uple of values for each gene, for example (**2 1 1 1 1**).

In constructing the partition sets only the unique cases are considered and the class label 1 is not taken in account. Twelve subsets of the partition (see Table 2.11) are obtained. Note that some of these contain inconsistent data which can be removed.

Table 2.10. Members of each of the fifteen partition sets with its respective class label.

class 1 1 2 6 8 16 21 22 **2 1 1 1 1**	Class 1 3 4 9 10 12 **1 1 2 1 1**	class 1 5 **1 2 2 1 1**	class 1 7 11 17 19 **2 1 2 1 1**	class 1 13 **2 2 2 1 1**	class 1 14 **1 1 1 1 2**	class 1 15 18 **1 1 1 1 1**	class 1 20 **1 2 1 1 2**
class 2 23 33 34 36 38 39 40 41 44 46 47 50 52 53 54 55 58 59 62 **1 2 1 1 2**	class 2 24 57 **1 2 1 1 1**	class 2 25 43 **1 2 2 1 2**	class 2 26 27 29 30 31 32 42 45 48 49 **1 1 1 1 1**	class 2 28 35 51 **2 2 1 1 2**	class 2 37 56 61 **1 1 1 1 2**	class 2 60 **2 1 1 1 2**	

Rough sets analysis for the Colon dataset produces:

Lower(C1) = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 16, 17, 19, 21, 22 }

Upper(C1)= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 16, 17, 19, 21, 22, 14 , 15, 18, 20, 23, 26, 27, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 40, 41, 42, 44, 45, 46, 47, 48, 49, 50, 52, 53, 54, 55, 56, 58, 59, 61, 62 }

Boundary(C1)={14 , 15, 18, 20, 23, 26, 27, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 40, 41, 42, 44, 45, 46, 47, 48, 49, 50, 52, 53, 54, 55, 56, 58, 59, 61, 62 }

Table 2.11. Partition sets without considering class label.

class c2	class c2	class c2	class c2	class c2	class c2	class c2	class c2
1 1	3 1	5 1	7 1	13 1	14 1	15 1	20 1 23 2
2 1	4 1		11 1		37 2	18 1	33 2 34 2
6 1	9 1		17 1		56 2	26 2	36 2 38 2
8 1	10 1		19 1		61 2	27 2	39 2 40 2
16 1	12 1					29 2	41 2 44 2
21 1						30 2	46 2 47 2
22 1						31 2	50 2 52 2
						32 2	53 2 54 2
						42 2	55 2 58 2
						45 2	59 2 62 2
						48 2	
						49 2	
2 1 1 1 1	**1 1 2 1 1**	**1 2 2 1 1**	**2 1 2 1 1**	**2 2 2 1 1**	**1 1 1 1 2**	**1 1 1 1 1**	**1 2 1 1 2**
classs c2	class c2	class c2	classs c2				
24 2	25 2	28 2	60 2				
57 2	43 2	35 2					
		51 2					
1 2 1 1 1	**1 2 2 1 2**	**2 2 1 1 2**	**2 1 1 1 2**				

Lower(C2) ={ 24, 25, 28, 35, 43, 51, 57, 60 }

Upper(C2) ={ 24, 25, 28, 35, 43, 51, 57, 60, 14 , 15, 18, 20, 23, 26, 27, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 40, 41, 42, 44, 45, 46, 47, 48, 49, 50, 52, 53, 54, 55, 56, 58, 59, 61, 62 }

Dependency degree including inconsistent instances

$$\gamma = (18+8)/62 = 0.4194$$

Dependency degree without inconsistent instances

$$\gamma = (18+8+10+19)/62 = 0.8871$$

γ is the proportion of objects of data which can be correctly classified with the knowledge given by rough sets analysis.

Example 2. Diabetes dataset

The Diabetes dataset has 768 instances and 8 attributes. Applying the RSES program[8] to the discretized data gives 28 reduct sets, and produces 21481 rules in the Rule set, many of these with only one match. Therefore, it is very convenient to perform a KDD process using Rough sets theory.

Discretization using Chi-Merge

The proportion of positive region is 0.997

Reduct = {pregnant, glucose, diagnostic, triceps, mass, pdf, age}

Some of the 4994 rules are:

(pregnant=2)&(triceps=2)&(age=1)=>(Class=1[56])

(mass=2)=>(class=1[48])

(pregnant=2)&(glucose=2)&(triceps=2)=>(class=1[40])

Size of rule	1	2	3	4	5	6
Rules	32	722	2355	1632	250	3

Discretization using Entropy

The proportion of positive region is 0.32

Reduct={ pregnant, glucose, insulin, mass, pdf, age}

Some of the 105 rules

(insulin=1)&(mass=2)&(pdf=1)&(age=2)=>(class=1 [60], 2 [58])

(pregnant=1)&(insulin=1)&(mass=2)&(age=2)=>(class=1[48],2[48])

(pregnant=1)&(glucose=1)&(mass=1)=>(class=1[73])

size of rule	3	4	5
Rules	19	76	10

The Chi-Merge dicretization yielded a Positive Region of .997 whereas Entropy based discretization method gave a Positive Region of .32. According to this, Chi- Merge seems to be a better discretization method than the one based on entropy. However, a larger number of rules is obtained using Chi-Merge, 499, whereas Entropy gives only 105. This is similar to the overfitting problem in a linear statistical model.

CHAPTER 3

DISCRETIZATION

3.1 Introduction

Discretization is the process for transforming continuous features into qualitative features. Firstly, continuous feature values are divided into subintervals. Then each interval is mapped to a discrete symbol (categorical, nominal or symbolic). These discrete symbols are used as new values of the original features.

Rough sets theory is based on decision tables. According to this, we need to discretize continuous features of a dataset before applying data mining methods based on Rough sets. There are plenty of discretization methods [13, 16, 53]. Two of the most commonly used are one based on the entropy measure and another one based on the Chi-square statistics. There are also other simple methods based on equal width intervals or equal number of instances in each interval. Discretization methods using the class label are referred as supervised discretization methods [16, 53].

A real value c, within the range of a continuous feature, that partitions the interval [a, b] into two subintervals [a, c] and (c, b], is called a cut point. A continuous feature with many cut points can make the learning process longer, while a very low number of cut points may affect the predictive accuracy negatively [53]. After the discretization process, the data complexity decreases, since the number of values of continuous feature is reduced. Hence, the learning process speeds up by discarding some unnecessary information [76]. A given number k could be considered as an upper bound for the number of cut points. In practice, k is set to be much less than the number of instances, assuming there is no repetition of continuous value for a feature [53].

The number of decision rules is affected by the number of values of the attributes. If many attributes have many values, the number of decision rules increases. Therefore, the number of cut points has to be evaluated carefully in the discretization process [16].

There are numerous discretization methods available in the literature. They consist of the following steps:

i) Sort the continuous values of the feature to be discretized.

ii) Split or merge intervals of continuous value according to some criterion, and

iii) Stop the discretization if some optimization criterion is optimized. The Minimun Description length (MDLP) is one of the stopping criteria most commonly used in discretization methods,

Dougherty et al. [16] classify discretization methods in three dimensions: Supervised versus unsupervised, global versus local and static versus dynamic.

a) Local discretization.

Local methods use only a subset of instances for the discretization process. It is related to dynamic discretization. A single attribute may be discretized into different intervals. But local techniques may result in the discovery of more useful cut points.

b) Global discretization

Global methods use the whole space of instances for the discretization process. Global techniques are more efficient, because only one discretization is used throughout the entire data mining process[16]. However, the significance of each feature is not equal for preserving the information of the original data. Also redundancy appears in global discretization [76].

Obviously global discretization is more complex than local discretization. Local discretization methods are restricted to single continuous features, while global methods are used when several continuous features need to be converted into qualitative features simultaneously.

c) Dynamic discretization.

Some classification algorithms have built-in mechanisms to discretize continuous attributes (for instance, decision trees: CART, C4.5). The continuous features are discretized during the classification process. Thus, these methods take into account the interdependencies between features.

d) Static discretization.

The continuous features are discretized prior to the classification task. That is, the features are discretized independently of each other. There is not a clear advantage of static discretization over dynamic discretization[16].

e) Supervised discretization

Supervised methods are only applicable when the data is divided into classes. These methods use the class information when selecting discretization cut points. Supervised methods can be further characterized as *error-based*, *entropy-based* or *statistics-based*. Error-based methods apply a learner to the transformed data and select the intervals that minimize a error measure on the training data. In contrast, entropy-based and statistics-based methods assess, respectively, the class entropy or some other statistic regarding the relationship between the intervals and the class.

Many discretization methods, such as equal-width-intervals and equal-frequency-intervals methods, do not use the class information during the discretization. These methods are called **unsupervised methods.**

Shi and Fu [76] proposed a global discretization algorithm based on rough sets. It modifies the criterion in selecting the best cut points of the entropy discretization, and makes it a global discretization method by introducing an inconsistency checking based on Rough sets theory instead of information gain. This preserves the behavior of the original data and overcomes the drawback of local discretization method. Then, the reduction of cut points is performed, which will not change the consistency level and lead to small size learning model. Using three evaluation criteria; the total number of intervals， the number of inconsistencies, and predictive accuracy, simulation results showed that the proposed global algorithm is superior to the entropy-based discretization method. Tay and Shen [81] proposed a discretization method using the Chi-square statistic along with an inconsistency measure based on Rough sets. Experimental results carried out on 11 datasets showed that this method does not improve the performance of the Entropy-based discretization over decision tree classifiers, but it does improve the performance of RoughSOM, a clustering algorithm that combines the features of Self organizing maps (SOM) clustering with features of Rough sets theory [80].

According to Shi and Fu [76] there are three important evaluation criteria for a discretization method:

i) The total number of intervals, since a smaller number of cut points gives better discretization results.

ii) The number of inconsistencies caused by discretization should not be much higher than the number of inconsistencies of the original data before the discretization.

iii) The predictive accuracy. The discretization process must not have a major effect in the misclassification error rate.

A good discretization method is obviously one with high performance on each of these criteria.

In this thesis, a discretization method based on Rough sets theory for finding the optimal cut points is introduced. We use the dependency measure based on Rough sets to evaluate the cut points proposed in each iteration. Considering that some conditional features are continuous and others are discrete, we only need to apply the Discretization process to continuous conditional features before continuing with further steps of the KDD process. The discretization algorithm proposed here uses the class information to find the optimal cut points.

In the next section, we describe the discretization methods used in this thesis.

3.2 Discretization methods used in this thesis.

3.2.1 Equal width intervals

In this discretization method, the range of the continuous feature is divided into k equal sized bins, where *k* is a user supplied parameter. Each bin must include at least a threshold number of instances.

The main advantage of this method is its simplicity, but it is vulnerable to the presence of outliers, since these may affect the range. This discretization process ignores the class information.

If a and *b* are the lowest and highest values of the continuous attribute, the width of intervals will be

$$W = (b\text{-a}) / \text{k}$$

The interval boundaries (cut points) are at

$$a+W,\ \text{a}+2W,\ \dots ,\ \text{a} + (k\text{-}1)W$$

There are three ways to determine the k value:

i) Sturges' Formula:

$$\text{k}=\log_2(n+1)$$

where *n* is the number of observations.

ii) Friedman-Diaconis' Formula:

$$\text{W}=2*\text{IQR}*n^{-1/3} ,$$

where IQR=Q3-Q1 is the interquartile range. Then, k=(b-a)/W

iii) Scott's Formula:

$$\text{W}=3.5 \text{ s n}^{-1/3},$$

where s is the standard deviation.

Equal frequency intervals is a related method that divide the range of the continuous feature in k-bins having approximately the same number of instances.

3.2.2 One R (1R).

One R was developed by Holte in 1993. It is a supervised discretization method using binning. After sorting the data, the range of the continuous attribute is divided into a number of disjoint intervals and the boundaries of those intervals are adjusted based on the class labels associated with the values of the feature. Each interval should contain a given minimum of instances (6 by default) with the exception of the last one. The adjustment of an interval boundary continues until the next value belongs to a different class to the majority class in the adjacent interval.

3.2.3 Entropy

Entropy-based discretization, was proposed by Fayyad and Irani in 1993. It uses the class information present in the data. The entropy (or the information content) is calculated on the basis of the class label. Thus, this is a supervised discretization method that uses the class information entropy of candidate partitions to select the bin boundaries. Intuitively, it finds the best split so that the bins are as pure as possible, i.e. the majority of the values in a bin correspond to having the same class label. Formally, it is characterized by finding the split with the maximal information gain.

Let S be a set of instances, a feature A, and a partition boundary T, the class information entropy of the partition induced by T, denoted E(A,T;S) is

$$E(A,T;S) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2)$$

where, $Ent(S) = -\sum_k p_k \log(p_k)$, with p_k denoting the proportion of instances belonging to the *k-th* class. Also, |S| denotes cardinality of the set S. For a given feature A, the boundary $T_{\min}$ which minimizes the entropy function over all possible partition boundaries is selected as a binary discretization boundary. Then, this method can be applied recursively to both partitions induced by $T_{\min}$ until some stopping condition is achieved. In this way, multiple intervals for the feature A are created.

Recursive partitioning within a set of values S stops if and only if

$$Gain(A,T;S) < \frac{\log_2(N-1)}{N} + \frac{\Delta(A,T;S)}{N}$$

where, N is the number of instances in the set S, and Δ is defined as

$$\Delta(A,T;S) = \log_2(3^k - 2) - [kEnt(S) - k_1 Ent(S_1) - k_2 Ent(S_2)]$$

where, k is the number of class labels represented in the set S, and k_i is the number of class labels represented in S_i. This is called the Minimum Description Length Principle (MDLP).

In order to illustrate the method, let us suppose that we have the following (attribute-value, class) pairs. Let S denote the 9 pairs given by: S = (0,Y), (4,Y), (12,Y), (16,N), (16,N), (18,Y), (24,N), (26,N), (28,N).

Let $p_1 = 4/9$ be the fraction of pairs with class = Y, and $p_2 = 5/9$ be the fraction of pairs with class = N.

The Entropy (or the information content) for S is defined as:

$$\text{Ent(S)} = - p_1 * \log_2(p_1) - p_2 * \log_2(p_2)\ .$$

In this case Entropy(S)=.991076.

If the entropy is small, then the set is relatively pure. The smallest possible value is 0. If the entropy is large, then the set is mixed. The largest possible value is 1, which is obtained when $p_1=p_2=.5$.

The cut points T are chosen from the midpoints of the attributes values. Thus, the first cut point must be chosen from the set, {2, 8, 14, 16, 17, 21, 25, 27}.

For instance if T=14

S1= (0,Y), (4,Y), (12,Y) and S2= (16,N), (16,N), (18,Y), (24,N), (26,N), (28,N)

E(S,T)=(3/9)*E(S1)+(6/9)*E(S2)=3/9*0+(6/9)* 0.6500224

E(S,T)=.4333

Information gain of the split, Gain(S,T) = Entropy(S) - E(S,T).

Gain=.9910-.4333=.5577

Similarly, for T=21 one obtains

Information Gain=.9910-.6121=.2789. Therefore T=14 is a better partition.

3.2.4 Chi-Merge

The Chi-Merge is a supervised discretization method introduced by Kerber in 1992. The basic idea is to merge neighboring intervals if the class information is independent of the interval. Two adjacent intervals should not have similar relative class frequencies, otherwise should be merged.

The algorithm is as follows:

Input: the original data

i. Sort the data for the given feature in ascending order.

ii. Construct initial intervals so that every value of the feature is in a separate interval.

iii. Compute χ^2 for each pair of adjacent intervals. χ^2 is given by

$$\chi^2 = \sum_{i=1}^{2}\sum_{j=1}^{k}\frac{\left(A_{ij} - E_{ij}\right)^2}{E_{ij}}$$

where:

k: the number of the classes.

A_{ij} : number of instances in the i-th interval, j-th class.

$E_{ij} = R_i C_j / N$ Expected frequency of examples in i-th interval, j-th class,.

R_i : number of instances in i-th interval = $\sum A_{ij}$, j= 1,2,…k.

C_j ; number of instances in the j-th class = $\sum A_{ij}$, i= 1,2.

N : Total number of instances.

If E_{ij}=0 then set E_{ij} to an small value, for instance 0.1

iv. Merge the pair of adjacent intervals with the lowest $\chi 2$ value

v. Repeat steps iii and iv until no χ^2 of any two adjacent intervals is less than a threshold value corresponding to a χ^2 with 1 degree of freedom and a significance level α.

The significance level α has been set to 0.10. Choosing a smaller value of α will generate fewer cut points.

The Chi2 algorithm introduced by Liu and Setiono [56] is a modification to the Chi-Merge method. It automates the discretization process by introducing an inconsistency rate as the stopping criterion and it automatically selects the significance value. Furthermore, Chi2 performs feature selection through discretization. However, the Chi2 algorithm does not consider the inaccuracy inherent in Chi-Merge merging criterion. The user-defined inconsistency rate also brings about inaccuracy to the discretization process. To overcome these two drawbacks, Tay and Shen (2002) proposed a modified Chi2 by using an inconsistency based on rough sets theory.

3.3 A Discretization method based on Rough sets theory

The main purpose in the discretization process is to transform data containing continuous attributes into a more simple data containing attributes with a limited number of values. This is done by considering cut points. After the transformation, the new dataset has to maintain the discriminating power of a classifier. There are several manners to obtain the cut points in a discretization algorithm. One of them is Equal width intervals. This method takes some cut points and gives a value in each interval. Rough sets theory can be applied to compute a dependency measure considering the partitioning generated by these cut points and the decisional feature in order to obtain a better set of cut points. We propose an algorithm to do so using the Scott's formula to obtain an upper bound for the number of cut points. Since the algorithm depend only of the computation of the dependency measure, then in the worst case the order of the algorithm is $O(n^2p)$, where n is the number of instances and p is the number of attributes. The algorithm is given below.

```
Input: The original dataset with n instances and f features
For each continuous feature $v_i$ (i=1,…,p)
        For $j$ in 2:$m_i$ ($m_i$ is nclass.scott($v_i$))
            Calculate the partition considering j equal intervals
            Evaluate each partition using an association measure based on Rough sets
                    $\gamma_j(v_i) = \frac{card(Pos(v_i, d))}{n}$
   Stopping criteria: select the optimal number of partition $p_i$
                    $p_i = \arg\max_j \gamma(v_i)$
        End For
  Divide the range of $v_i$ considering $p_i$ intervals.
  End For
  Output: A new data matrix with discrete values
```

Fig 3.1. Discretization algorithm based on Rough sets.

3.4 Results and Discussion

Our discretization algorithm based on rough sets theory is applied to eight datasets coming from the Machine Learning Database Repository available at the University of California at Irvine. A brief description of these datasets is given in the appendix A of the thesis. Some of these datasets have only continuous features and others have both continuous and discrete feature. The algorithm is applied only to the continuous features.

Table 3.1: Comparison of the number of cut points per feature using five discretization methods.

Dataset	Rough Method	Equal width Bound Scott	Entropy Method	1R method	Chi-Merge
Iris	6 4 4 4	7 9 6 5	3 3 3 3	3 3 3 3	7 5 4 4
Glass	9 14 2 10 12 15 10 10 7	13 14 6 11 13 18 13 11 9	3 2 2 3 1 4 4 2 1	6 9 8 3 9 6 4 6 8	15 7 7 9 8 8 9 5 2
Diabetes	5 13 16 8 19 21 16 6	14 17 17 17 20 23 19 14	2 4 1 1 3 2 2 2	8 6 6 10 14 16 8 15	5 14 4 9 41 45 80 9
Heartc* (1,4,5,8,10)	8 6 17 6 10	11 12 17 11 11	2 1 1 2 2	5 8 9 2 4	6 6 32 18 8
Ionosphere	7 10 2 2 8 2 6 5 2 8 2 9 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	9 10 8 9 9 8 8 9 8 9 7 9 7 9 7 9 7 8 7 8 7 8 7 8 8 8 7 8 8 8 8 9	4 5 4 6 3 5 5 4 5 5 6 4 5 5 6 3 6 3 5 5 5 3 5 3 3 3 5 3 5 3 5 5	3 5 5 5 6 5 5 5 5 5 7 5 7 6 6 5 7 5 7 5 7 5 5 5 2 3 6 6 6 5 5 5	17 50 23 31 23 24 40 33 24 32 39 45 35 44 42 51 41 33 58 34 30 35 34 28 47 38 37 32 42 26 51
Crx * (2,3,6,8,14, 15)	2 13 2 2 17 35	14 14 8 21 30 48	2 2 3 2 2 2	12 11 15 10 14 9	10 6 6 4 11 7
Vehicle	10 2 2 14 23 20 9 9 3 13 15 13 9 18 2 2 2 2	16 12 13 19 32 32 13 13 13 14 17 13 14 28 13 13 14 11	5 4 4 3 4 4 5 5 5 5 4 7 4 3 2 2 5 2	18 26 16 20 30 18 20 19 15 21 24 20 27 28 29 33 22 22	6 5 15 11 6 4 9 10 6 6 13 9 4 5 3 3 5 7
German* (2,5,13)	6 13 2	17 19 15	2 2 1	9 6 5	10 27 2

*In the datasets: Heartc, Crx, and German, discretization has been applied only to the features appearing between parentheses because these are continuous.

The Dependency measure used in the algorithm reflects the relation between each conditional feature and the decisional feature. A comparison between the number of intervals generated for the continuous features in each dataset applying five different discretization methods, i.e., Rough sets, Equal width intervals, Entropy method, 1R method, and Chi-Merge, is shown in the Table 3.1. We wrote an R function for the first method, the second method comes with R, and the last three are available in the Dprep library of R [2].

In general, the number of cut points generated by the rough set discretization method is smaller than the ChiMerge and the Scott's method, slightly higher than the 1R method, but is larger than the Entropy discretization.

In order to prove the efficiency of the discretization process, we calculate the misclassification error of the LDA classifier applied to the discretized data. The misclassification error was estimated by 10-fold cross-validation. The experimentation was carried out considering the Entropy method, the 1R method, the Chi-Merge method, and the Rough set algorithm.

Table 3.2 shows the misclassification error rate using the LDA classifier and the discretized data using all the methods showed in table 3.1 except by the equal width intervals method.

Table 3.2: Comparison of the Misclassification Error rate for the LDA classifier using discretized features and without performing discretization

Dataset	Rough Method	Entropy Method	1R method	ChiMerge	Without discretization
Iris	0.0833	0.040	0.0333	0.0333	0.0200
Glass	0.4200	0.3009(5,9)	0.2560	0.3228	0.3803
Diabetes	0.2305	0.2083(3,4)	0.2385	0.2282	0.2273
Heartc	0.1616	0.1649(4,5)	0.1481	0.1515	0.1515
Ionosphere	0.1336	0.1364	0.1373	0.1558	0.1421
Crx	0.1347	0.1347	0.1356	0.1349	0.1349
Vehicle	0.3070	0.302	0.3332	0.3990	0.2199
German	0.2432	0.2351(13)	0.2337	0.2414	0.2422
MEAN	0.2142	0.1903	0.1895	0.2084	0.1898

In four datasets: Glass, Diabetes, Heartc, and German, the Entropy-based method discretized some features only to one value. These features can not be considered in the LDA classifer. Actually, these features are irrelevant since they do not present variation, and they are shown in Table 3.2, within parentheses. Therefore, the entropy-based method has this advantage with respect to other methods since features without variability must be eliminated from the model, reducing in this way the dimensionality of the data.

Table 3.3 shows misclassification error rate using the KNN classifier and the discretized data using all the methods shown in table 3.1 except by the equal width intervals method. Here, all the features have been used in the construction of the classifier.

Table 3.3: Comparison of the Misclassification Error rate for the KNN classifier using discretized features and without performing discretization

Dataset	Rough Method	Entropy Method	1R method	ChiMerge	Without discretization
Iris	0.1140	0.053	0.0400	0.0460	0.0340
Glass	0.5331	0.4434	0.2808	0.4261	0.4733
Diabetes	0.2520	0.2227	0.2856	0.2908	0.2851
Heartc	0.2265	0.1898	0.2063	0.2582	0.3397
Ionosfera	0.1264	0.1678	0.1962	0.2196	0.1552
Crx	0.1998	0.1770	0.2404	0.2577	0.3068
Vehicle	0.2685	0.2807	0.3393	0.3065	0.3478
German	0.2529	0.2727	0.2588	0.2994	0.3465
MEAN	0.2466	0.2252	0.2309	0.2630	0.2860

3.5 Conclusions

Discretization is an important step in the KDD process, since many algorithms have been developed only for features representing categorical data.

From our experimental results we arrive to the following conclusions:

i) Our proposed discretization algorithm based on a Rough sets criteria yields on average similar number of cut points than the 1R, but greater than the entropy discretization, and smaller than the Equal width and ChiMerge.

ii) According to the misclassification error rate using the LDA classifier on the discretized data, our algorithm gives similar results than Chi-Merge, but it is not much better than either Entropy or 1R discretization. Also, the LDA classifier tends to give higher misclassification error rates using the discretized data instead of the original data. Our Rough sets discretization methods performs poorly on the Glass dataset, perhaps affected by the large number of classes. Glass has six classes.

iii) According to the misclassification error rate using the KNN classifier on the discretized data, our algorithm gives better results than the Chi-Merge method but is outperformed by both the Entropy and the 1R discretization. Also, the KNN classifier tends to give lower misclassification error rates using the discretized data instead of the original data.

CHAPTER 4

FEATURE SELECTION

4.1 Introduction

The dimension reduction of a dataset can be done in two different manners; through feature selection considering a subset of the original feature or by feature extraction transforming the original features to extract a smaller amount of new features. Figure 4.1 shows the taxonomy of dimensionality reduction methods.

Dimension reduction is needed when the dataset has a large number of features. Classification and regression algorithms could present problems in their general behavior if redundant and irrelevant features are considered. This is a main reason for many investigators to search for different methods to detect these features. In reducing the number of features it is expected that the ones that are redundant and irrelevant will be deleted.

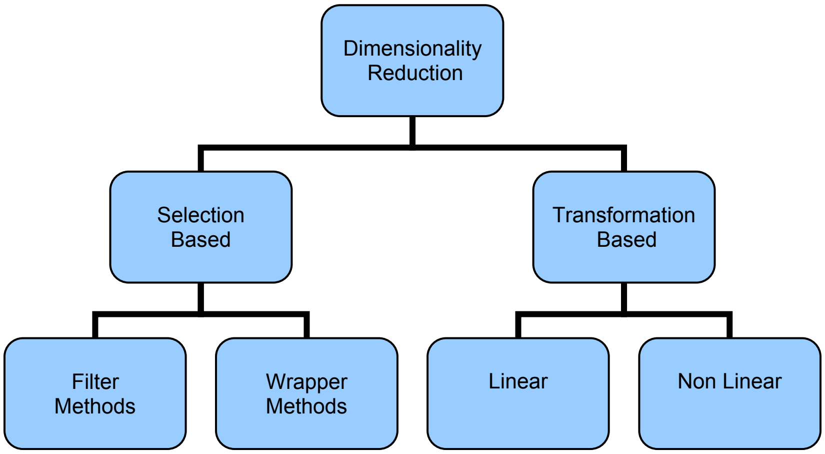

Fig 4.1. Feature Reduction Approach

Lesh et al. [46] adapted data mining techniques to act as a preprocessor to construct a set of feature to be used for classification purposes.

Similarity relations can be used to evaluate a subset of features [73, 77]. Thus, Rough sets theory offers a new alternative to select the subset of dispensable features. Furthermore, an evaluation function for feature selection based on Rough sets theory has been proved to be monotonic [75].

Our principal purpose is to find the minimal attribute subset with respect to class attribute D. Intuitively, the feature selection problem could be solved enumerating all the candidate subsets and apply the evaluation measure to them. This is called exhaustive search, and it is almost infeasible to be done. The number of possible subsets is 2^N, so the time complexity of searching all of them is $O(2^N)$. Heuristic methods for searching avoid the brute-force search, but at the same time take the risk of losing optimal subsets.

The problem of feature selection consists of the search *d* features from a given set of *m* (*d*<*m*) features, which will provide a similar o better performance for a classifier based on a smaller number of features. In others words, feature selection methods determine an appropriate feature subset such that the classification error is optimal [26, 39]. The chosen features permit that pattern vectors belonging to different categories occupy compact and disjoint regions in an m-dimensional feature space. Figure 4.2 shows the steps of the feature selection problem.

There are two main reasons to keep the dimensionality of the feature space as small as possible: cost minimization and classification accuracy [15]. Cost minimization is achieved because after feature selection the classifier's computation will be faster and use less amount of memory [27]. A careful choice of the features is needed since a bad reduction may lead to a loss in the discrimination power and thereby a decrease in the accuracy of the resulting classifier.

The feature selection methods depend on the way that the subset is generated and on the evaluation function used to evaluate the subset under examination [27, 33]. There are three types of procedures for feature subset generation: Complete, heuristic and random [43, 68]. The evaluation function may be a consistency measure, a distance measure, an information gain measure or the misclassification error.

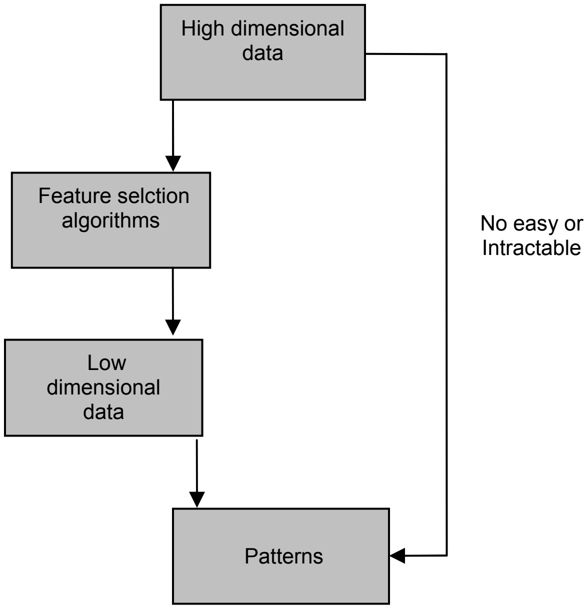

Fig 4.2. Feature Selection Problem

Let J be a feature evaluation function. Assume that a higher value of J indicates a better feature subset. The function J has the *monotonic property* if given two features subsets X_1 and $X_{2,}$ if $X_1 \subset X_2$, then $J(X_1)<J(X_{2`})$. Thus, the performance of a feature subset should improve whenever a feature is added to it. Many evaluation functions do not satisfy this monotonic property, one example is the error rate.

Depending on the generation procedure and the evaluation function [26], the feature selection methods could be divided in two types: filter methods and wrapper methods. A brief description of both methods is given in the next section.

4.2 Filter Methods

Filter methods do not require the use of a classifier to select the best subset of features. Instead these methods use general characteristics of the data to evaluate features. Among the most important filter methods are: the RELIEF [40], Las Vegas Filter (LVF) [42] and FINCO, a procedure introduced by Acuña [1]. In the first and the third a relevance weight is given to each feature in the dataset. This weight is changed iteratively according to a feature relevancy's feature. In the LVF method a relevancy is given to a subset of features. Choosing randomly a large number of subsets one expects to obtain a subset with the highest relevancy. All these methods are computationally cheap and preserve only the necessary information to perform Knowledge Discovery techniques.

4.3 Wrapper methods

Wrapper methods use the misclassification error rate of a given classifier as the evaluation function [40]. In this thesis, two classifiers will be used, linear discriminant analysis (LDA), and the k-nearest neighbor (KNN) classifier, which were described in section 1.1. There are three main approaches to wrapper methods: s*equential forward selection* (SFS), *Sequential Backward selection* (SBS), and the s*equential floating forward selection* (SFFS). A brief description of them follows:

Sequential forward selection (SFS). This method selects the best feature and then adds the next best feature, such that in combination with the previously selected features maximizes the criterion function. Once that a feature is selected, it cannot be discarded in a later step. It is computationally attractive since in order to select a subset of size two, it examines only (m - 1) possible subsets, where m denotes the number of features.

Sequential Backward Selection (SBS). The principal idea is to see if the classifier can maintain its accuracy by removing one feature at a time until there is only one feature or until accuracy deteriorates to an intolerable level. This method considers initially all the set of variables and then discards the worst feature based on the loss of classification accuracy.

Once that a feature is deleted, it can not be taken into account at a later step. A particular case of SBS is **Recursive Feature Elimination (RFE)** introduced by Guyon and Elisseff [26]. It uses as underlying classifiers the Support Vector machine (SVM) classifier. At each stage the feature with the smallest squared coefficient in the SVM model is deleted. RFE can be generalized by eliminating more than one feature at each stage.

Sequential Forward Floating Search (SFFS). This method is a generalization of the *plus-l* and *away-r* method. First, the feature subset is enlarged by *l* features using forward selection and then *r* features are deleted from the new subset using backward selection. In SFFS, the values of *l* and *r* are determined automatically and updated dynamically. This method provides a close to optimal subset with an affordable computational cost [70].

4.4 Feature selection method based on Rough set theory

Rough sets have been used as a feature selection methods by many researchers among them Jensen and Schen [37, 38], Zhong et al [93], Wang [86] and Hu et al. [33]. The Rough set approach to feature selection consists in selecting a subset of features which can predict the classes as well as the original set of features. The optimal criterion for Rough set feature selection is to find shortest or minimal reducts while obtaining high quality classifiers based on the selected features. Other criterion can be the number of rules generated by the reducts. There are many rough rough sets algorithms for feature selection. The most basic solution for finding minimal reducts is to generate all possible reducts and choose any with minimal cardinality, which can be done by constructing a kind of discernability function from the dataset and simplifying. It has been shown that the problem of minimal reduct generation is NP-hard and the problem of generation of all reducts is exponential. Therefore heuristic approaches have to be considered[93].

Hu et al. [33] consider a ranking of the features based on the indiscernability matrix. Thus, a weight w(a) is assigned to every attribute a. The weight is initialized to zero, and every time one feature appears in an entry of the matrix its relevance increases. The shorter the entry of the matrix is, the more relevant the features in such entry might be. If an entry contains only one feature, then such feature must be considered in the core set. Unfortunately, this algorithm has a high complexity since it needs to compute the Discernability matrix. For large datasets, the authors used sampling to reduce the computational burden required on the computation of the Discernibility matrix. Theirs experimental study include 25 medium size datasets and 9 large datasets. Five of these data sets are considered in this thesis.

Deogun, et al.[15] developed four feature selection algorithms in the context of Rough sets methodology, but instead of using the positive region as significance of the attribute set, their algorithm uses upper approximation. The algorithms uses sequential backward feature elimination to reduce the search space. The authors performed an experimental study using thirteen datasets. Three of them; *Iris*, *Glass* and *Breastw*, are considered in the experiments of this thesis.

Rough sets attribute reduction (RSAR) technique has been applied in supervised and unsupervised classification [75]. QuickReduct is the most well known algorithm for feature selection using Rough sets. Its pseudo-code algorithm is shown in Figure 4.3. This is an incremental procedure, where in each step a feature is added to the Reduct, in such way that dependency measure increases. The procedure stops when the dependency measure of the set of features being considered is equal to the dependency measure using all the conditional features. However, it has been proved that this method does not always generate a minimal reduct since the dependency measure is not optimal. It does result in a close-to-minimal reduct, though, which is still useful in greatly reducing dataset dimensionality. Adittionally to not being a non-optimal heuristic, the algorithm also does not take in account the information lost in the discretization procedure [36].

```
QuickReduct (C, D)
Input: C, the set of all feature attributes; D, the set of class attributes.
    $R \leftarrow \{\}$
    do
      $T \leftarrow R$
      for each $a \in (C - R)$
          If $\gamma(R \cup \{a\}, D) > \gamma(T, D)$
              $T \leftarrow R \cup \{a\}$
      $R \leftarrow T$
    Until $\gamma(R, D) = \gamma(C, D)$
return $R$
Output: R, the attribute reduct, $R \subseteq C$
```

Fig 4.3. The Quick Reduct algorithm.

Duntsch and Gediga [17] consider first binary transformation for the information system and then apply rough sets theory to extract relevant features. This method is considered a data filtering.

Stepaniuk[77] applied a wrapper method considering a decision rule discovery (classifier) based on rough sets theory. The procedure was applied on a medical dataset related to Diabetes Mellitus containing 12 conditional attributes on 107 patients.

Zhong ate al. [93] proposed an algorithm which combine Rough set theory with greedy heuristics for feature selection. They applied the algorithm to nine datasets, one of the breast cancer is considered in this thesis.

In this thesis, Rough sets theory is applied to find an optimal subset of features as suggested in [55, 77]. We have two proposals. In our first proposal, we find the best features by ranking them according to its dependency measure γ. Attributes with largest dependency coefficient are selected as members of the best feature subset, which is a sub-optimal subset of features. On the other hand, using the positive measure, we could relate simultaneously various conditional attributes with the decisional attribute. Following a forward sequential selection method, similarly to the QuickReduct, we could obtain a subset of features with a high positive measure in each step.

In the second method, we introduce an hybrid method by applying the Sequential Forward feature selection algorithm on the ranked features according to the dependency measure based in Rough sets theory. Two classifiers are used the LDA and the KNN classifier.

4.5. Feature selection by ranking according to the dependency measure.

A basic filter algorithm to perform feature selection based on rough sets is shown in Figure 4.4. This algorithm calculates the dependency between every conditional feature considering the decisional feature. After ranking the features by their dependency measure only the features with higher dependency values are included in the final subset of best features. The order of the algorithm is O(n^2p),), where *n* is the number of instances and p is the number of attributes. A plot can help to distinguish the most relevant features (see Figure 4.6 for the Diabetes dataset and Figure 4.7 for the Glass dataset).

```
Input: Dataset containing C conditional features and a decisional features D.
       Initialize the best subset of features B as the empty set
       For  i  in 1:number of conditional features
       Apply some evaluation measure based on dependency of Rough  sets.
       End for
       b)  Order the features according to dependency measure
       c)  Select only the feature with high dependency measure.

       Output: A subset B of best features.
```

Fig. 4.4. Algorithm for feature selection based on Rough sets.

4.6. An hybrid feature selection method.

This is a sequential forward feature selection method used along with a classifier. We start with the feature giving the highest positive measure. The misclassification error obtained with this feature is taken as a basis error. In each step, we include the next ranked feature that along with the previous one yields the lowest misclassification error for the given classifier. The omitted ranked features are not considered in the next steps. The procedure stops when either we exhaust the list of ranked features or when the misclassification error does not decrease. The order of this algorithm is $O(n^2 \times p) + O(p \times O(classifier))$. The algorithm is given in Figure 4.5.

```
Input: A discretized dataset D with T ranked conditional features and a
decisional feature D, a classifier L.
        Initialization: Let $B=T_1$, the first feature in T.
Let ME (B)=ME($T_1$)=misclassification error of the first feature in T.
         For i in 2:card(T)
              a) Compute ME(BU{$T_i$}) the misclassification error of BU{$T_i$} .
              b) If ME(BU{$T_i$}) < ME(B). Then B=BU{$T_i$},
              and ME(B)=ME(BU{$T_i$}).
          Output: B, the best subset of features
```

Fig 4.5. An hybrid feature selection algorithm.

4.7 Results and Discussion

The next table contains the subset of features selected using a rough set criteria, after considering previously four different types of discretization : Rough Set method, 1R method, Equal width interval, and Chi-Merge.

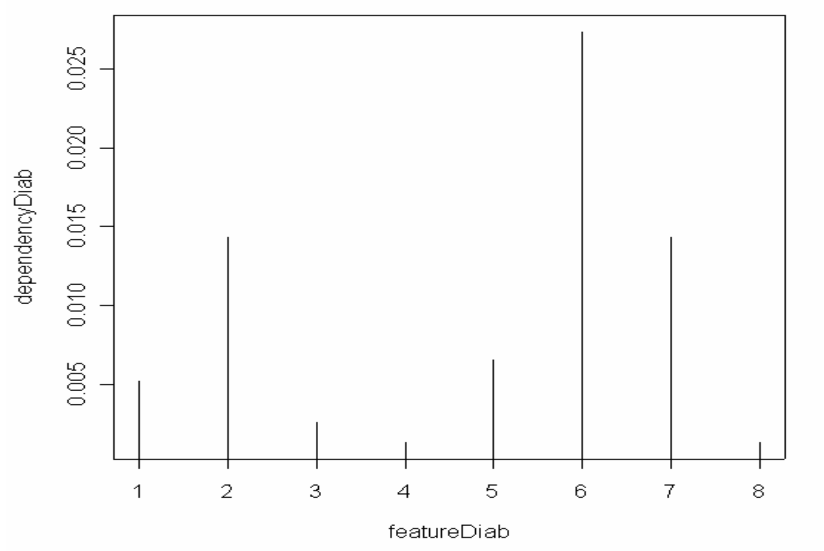

Fig 4.6 Dependency for each feature on Diabetes dataset.

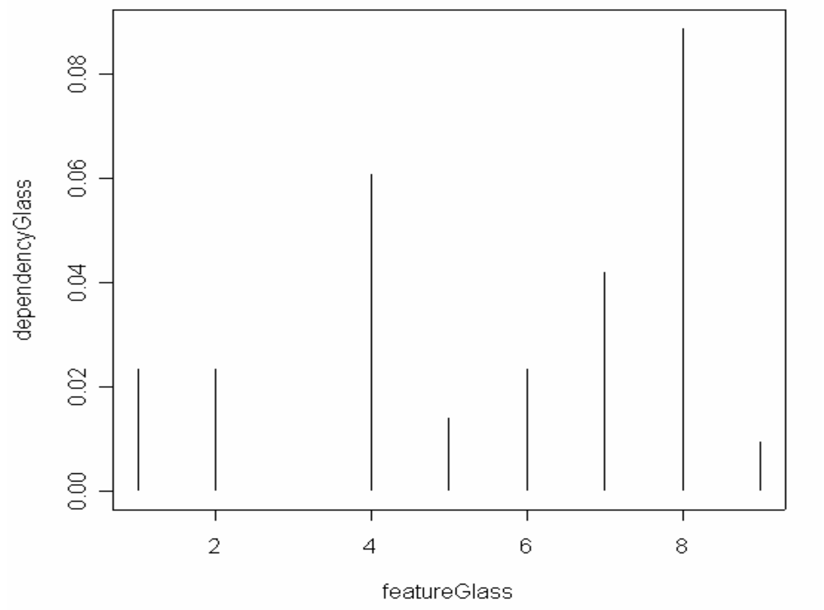

Fig 4.7. Dependency for each feature on Glass dataset.

Table 4.1. Subsets of features selected using Rough sets criterion with four discretization methods, and features selected by two wrapper methods.

Dataset	Rough Method	1R method	Equal width	ChiMerge	SFS - LDA	SFS - KNN
Glass	8 4 7 1 2 6	8 9 3 7	8 4 7 1 2	8 4 7	4 3 8 9 2	8 4 3
Bupa	4 1 5 3 6	6 1 2 3	4 5 3 6 1	2 5 4 3	3 4 5 6	1 3 5
Heartc	1 5 8 10 4	5 1 10	5 8 4 10 1	5 8 4 1 10	3 9 12 13	2 12 13
Ionosphere	3 1 5 7 8 12 2 10	3 1 19 7 16 6 12 4 14 28	3 1 8 5 7 12 2 10	27 3 31 4 1 6 21 25 19 13 11 29 7 15 26 23 22 8 9 2 12 5 16	3 6 19	1 3 4 14
Diabetes	6 2 7 5	6 8 7	6 2 7	7 6 5 4 8	2 6 7	2 6 7
Vehicle	8 12 7 9 11 6 10 1 5 13 14	8 7 12	12 7 8 9 11 6 14	7 12 11 8 9 13 3	1 3 4 5 6 8 10 11 17 18	2 5 6 8 9 10

The entropy-based discretization method was not included because very often yields a gross discretization. This means that a continuous variable is discretized into a categorical variable with few different values. Sometimes, even a continuous variable is discretized to one integer value. This means that this variable is not a relevant feature. For instance, for these datasets: Bupa, Glass and Diabetes, the entropy discretization yields irrelevant features.

The first four columns of Table 4.2 contain the misclassification error rate for LDA considering only the selected features using the rough set criterion with four discretization methods, the fifth column contains the misclassification error using the well known SFS wrapper feature selection method and the last column contains the misclassification error for the classifier without performing feature selection. Table 4.3 is similar to table 4.2 but using the KNN classifier instead of the LDA.

Table 4.2 : Comparison of misclassification error rate for the LDA classifier with feature selection after four discretization methods, wrapper feature selection, and without feature selection

Dataset	Rough Method	1R method	Equal width	Chi-Merge	SFS	No Feat Sel.
Glass	0.4112	0.4588	0.3995	0.4327	0.3789	0.3808
Bupa	0.3176	0.4475	0.3173	0.3286	0.3257	0.3182
Heartc	0.2777	0.3097	0.2767	0.2804	0.1569	0.1663
Ionosphere	0.1723	0.1658	0.1720	0.1586	0.1831	0.1433
Diabetes	0.2294	0.3083	0.2300	0.3105	0.2295	0.2273
Vehicle	0.2925	0.5855	0.4085	0.3858	0.2426	0.2202
MEAN	0.2835	0.3793	0.3007	0.3161	0.2528	0.2427

Table 4.3 : Comparison of misclassification error rate for the KNN classifier with feature selection after four discretization methods, wrapper feature selection, and without feature selection

Dataset	Rough Method	1R method	Equal width	Chi-Merge	SFS	No Feat Sel.
Glass	0.4963	0.4878	0.4859	0.5313	0.4827	0.4813
Bupa	0.3521	0.4657	0.3579	0.3562	0.3455	0.3402
Heartc	0.3569	0.4414	0.3558	0.3609	0.1831	0.3474
Ionosphere	0.1301	0.1390	0.1324	0.1384	0.0806	0.1547
Diabetes	0.2845	0.3380	0.2722	0.3208	0.2714	0.2859
Vehicle	0.3823	0.4601	0.4111	0.4248	0.2971	0.3503
MEAN	0.3337	0.3887	0.3359	0.3554	0.2767	0.3266

Table 4.4 shows the features selected by the hybrid method using three discretization methods and the LDA classifier and Table 4.5 shows the features selected by the hybrid method using three discretization methods and the KNN classifier. The 1R discretization method was not considered in these tables due to its poor performance on the feature ranking procedure.

Table 4.4. Subsets of features selected using the hybrid method based on rough sets with three different discretization methods, and features selected by the SFS wrapper method along with the LDA classifier.

Dataset	Rough Method	Equal width	Chi-Merge	SFS - LDA
Glass	8 4 7 2	8 4 7 2	8 4 7	4 3 8 9 2
Bupa	4 1 3 6	4 3 6	2 5 3	3 4 5 6
Heartc	1 8 10 4	5 8 4 10	5 8 4 1 10	3 9 12 13
Ionosphere	3	3	27 3 31 4 6 21 25 19 13 11 16	3 6 19
Diabetes	6 2 7	6 2 7	7 6 4 8	2 6 7
Vehicle	8 12 6 10 1 5 13 14	12 7 8 9 11 6 14	7 12 11 9 13 3	1 3 4 5 6 8 10 11 17 18

Table 4.5. Subsets of features selected using the hybrid method based on rough sets with three different discretization methods and feature selected by the SFS wrapper method along with the KNN classifier.

Dataset	Rough Method	Equal width	Chi-Merge	SFS - KNN
Glass	8 4 1	8 4 7 2	8 4 7	8 4 3
Bupa	4 1	4 5 3 6	2	1 3 5
Heartc	1 8	5 8 1	5 8 1 10	2 12 13
Ionosphere	3 1 5 8 2	3 1 8 7	27 3 4 1 25 26	1 3 4 14
Diabetes	6 2	6 2 7	7 6 8	2 6 7
Vehicle	8 12 7 6 1 5 14	12 7 8 9 11 6 14	7 12 8 9 3	2 5 6 8 9 10

The first three columns of Table 4.6 contain the misclassification error rate for LDA considering only the selected features using the hybrid method based on Rough sets with three different discretization methods, the fourth column contains the misclassification error using the well known SFS wrapper feature selection method and the last column contains the misclassification error for the LDA classifier without performing feature selection. Table 4.7 is similar to table 4.6 but using the KNN classifier instead of the LDA.

Table 4.6. Misclassification error rate for the LDA classifier after feature selection using the hybrid method

Dataset	Rough Method	Equal width intervals	Chi-Merge	SFS	Without Sel.
Glass	0.3999	0.3869	0.3822	0.3920	0.3780
Bupa	0.3553	0.3489	0.4188	0.3257	0.3182
Heartc	0.2606	0.2757	0.2814	0.1569	0.1663
Ionosphere	0.1715	0.1715	0.1418	0.1831	0.1433
Diabetes	0.2287	0.2299	0.3053	0.2295	0.2273
Vehicle	0.3069	0.4074	0.4260	0.2426	0.2202
MEAN	0.2872	0.3034	0.3259	0.2550	0.2422

Table 4.7. Misclassification error rate for the KNN classifier after feature selection using the hybrid method

	Rough Method	Equal width intervals	Chi-Merge	SFS	Without Sel.
Glass	0.5242	0.4906	0.5219	0.4827	0.4808
Bupa	0.4046	0.3388	0.4130	0.3371	0.3324
Heartc	0.3107	0.3619	0.3552	0.1851	0.3575
Ionosphere	0.1054	0.0757	0.0843	0.0780	0.1544
Diabetes	0.2723	0.2692	0.3382	0.2752	0.2848
Vehicle	0.3566	0.4133	0.3846	0.2978	0.3490
MEAN	0.3290	0.3249	0.3495	0.2760	0.3265

Comparing our results with those from the Deogun et al’s study [15], we have in common only the *Glass* datasets. Theirs algorithm selects two feature whereas our algorithms selects between two and six features. However the accuracy of the classifiers used is much better considering the selected features for our algorithms.

4.6 Conclusion

Our experimental results show that feature selection using Rough sets theory is a good option for data preprocessing. Misclassification error rates for two classifiers, LDA and KNN, constructed considering only the selected features by rough sets based methods along with several discretization methods gives the best results when rough discretization is used. For the LDA classifier, the misclassification error using features selected with the hybrid method are higher than using the selected features by the wrapper method and without performing feature selection. The same result holds when the KNN classifier is used. However, the misclassification error rate after feature selection is lower than when all the features are used.

CHAPTER 5

INSTANCE SELECTION

5.1 Introduction

An instance or case is a collection of values taken from an observation considering all the features (conditional and decisional). It is also named a t-uple, sample or data point. Some of the instances in a dataset appear more than once or could be very similar to others, then these could be eliminated since they are redundant. The elimination of similar instances tackle down the redundancy problem. The instance selection problem reduces the training data by searching for the optimal instances and reaching high accuracy of Knowledge Discovery on the unseen data (see Fig. 5.1). Instance selection has the purpose of selecting high quality cases, eliminating noisy data, and inconsistent data. This will produce a reduction of the storage requirement and a speed-up of the computation of posterior KDD tasks [71].

There are various strategies for drawing a representative subset of samples from a dataset. The size of a suitable subset is determined by taking into account the cost of computation, memory requirement, accuracy of the estimator, and other characteristics of the algorithm and dataset. In general, a subset size is determined in such way that the estimates for the entire data set do not differ by more than a stated error margin in more than δ of the samples (Kantardzic, 2003) [40]. It is considered that a KDD task can be executed efficiently using the chosen subset of the original data set.

Liu (2002) [54] considers the following reasons to carry out instance selection:

a) Enabling: every data mining algorithm is somehow limited by its capability in handling data in terms of sizes, types, and formats. When a data set is too large, it may not be possible to run a data mining algorithm. A data mining task cannot be effectively carried out without data reduction.

b) Focusing: the data includes almost everything in a domain. Instance selection is a natural and sensible way to focus on the relevant part of the data.

c) Cleaning: most datasets present inconsistency and missing values that may affect the KDD process. Doing instance selection we can remove them.

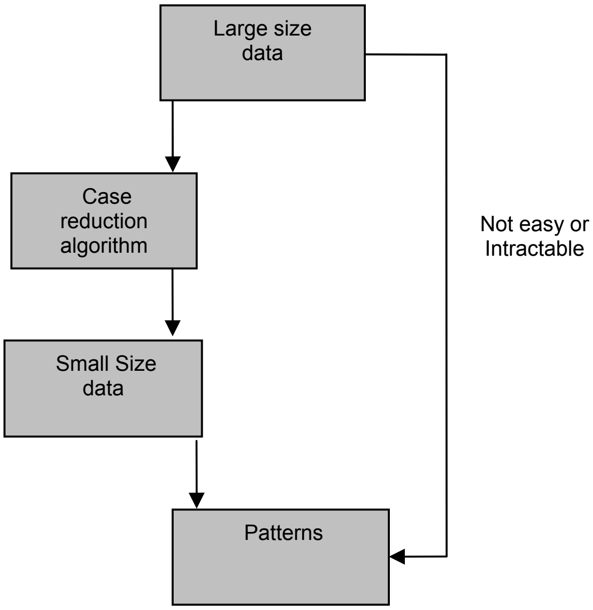

Figure 5.1. Case reduction in KDD.

Yu et al. (2002) [91] proposed four instance selection algorithms to select training instances for memory-based collaborative filtering, a data mining used to make personalized recommendations. These algorithms reduce the time complexity, and the prediction performed over the reduced training set is more efficient than current methods [91].

Cano et al. (2003) [9] classify instance selection methods in four groups: a) methods based on Nearest Neighbors, b) methods based on ordered removal, c) methods based on evolutionary algorithms, and d) methods based on Random sampling [9].

Wilson and Martinez [87] propose three algorithms, called Integrated Decremental Instance-Based Learning, where they select bad instances for exclusion rather than select good instances for inclusion. In an experimental study, Jankowski and Grochowski [35] compare eighteen instance selection methods on six datasets. They conclude that using prototypes, that represent of subset of several instances, is the most effective instance selection algorithm. This is very similar to a clustering algorithm.

Instead of using simple random sampling, some property coming from Rough sets theory could be applied to select a good subset of instances, which can be used to posterior KDD tasks. This property is related to the equivalence relation making elementary sets. The negative region concept of Rough sets theory is a good criterion to determine the low quality cases and these can be eliminated in a first step.

In this thesis, the equivalence relation helps to obtain elementary subsets. Then, we select a representative sample, drawing randomly a 100*p*% instances from the positive region. The value of p is chosen to be 0.6, but it could be a higher value.

5.2 Instance selection using Rough sets

Each group of instances has an instance as a representative of elementary blocks. Then, extracting a subset of instances is related to finding out weights for each elementary sets produced by Rough sets theory. The extraction of some interesting instances from the positive region could help to the posterior analysis of a large dataset, since it decreases its computational complexity. Thus, the computation time of executing some KDD tasks is also reduced since a smaller dataset is used instead of the original one.

Some instances in the dataset are inconsistent, because they might have all their feature values similar to other instances, but lie in a different class. These instances must be considered very carefully. Elementary sets formed using the set C of conditional features help to identify the weight class where there should be inconsistent instances.

Some criteria to eliminate instances are redundancy and incompleteness [54]. When we use the Rough sets theory, some of the instances are inconsistent with others. If each elementary set contains similar observations considering every conditional feature, then some elementary set could contain observation from different decisional classes. These elementary sets are part of the Boundary Region. Then, the Negative region is a good criterion to find out the low quality cases and they can be eliminated in a first step of the preprocessing process.

Salamó and Golobardes (2001) [71] propose two algorithms based on Rough sets theory to reduce the data in case based reasoning (CBR) systems, where solutions to similar problems are stored as cases in a case memory. These methods are; Accuracy Rough sets Case Memory (AccurCM) and Class Rough sets Case Memory (ClassCM). Both techniques use the information of reducts and core to extract the relevant cases. Empirical results on twelve datasets show that both reduction techniques produce an improvement in the misclassification error rate compared with other instance selection methods. However, the percentage of reduction of instances is low.

Geng and Hamilton (2002) [20] propose the ESRS algorithm, based on extended similarity-based Rough sets theory, which selects a reasonable number of instances while maintaining good classification accuracy. Empirical results on nine datasets indicate that the misclassification error rates using the reduced dataset are better than the ones obtained with reduced datasets using other techniques. However the algorithm requires two user-specified parameters, the consistency and similarity thresholds.

5.3 A new algorithm for instance selection using Rough Sets theory

It is clear that a good sampling scheme will reduce the computational complexity of a data mining algorithm. Our proposed instance selection algorithm combines Rough sets theory with a random sampling   of instances as it is considered by Cano et al. [9].
First, we discard the inconsistent data that usually lie in the Boundary region. After that, we select a random sample of instances from the positive region.

. The order of the algorithm is O($n^2 \times p$),), where n is the number of instances and p is the number of attributes.

The algorithm is presented in Fig. 5.2. These instances are consistent and could be used in a future learning algorithm of KDD.

```
Input: The original dataset and the percentage 100$p$% to be sampled from the positive region.
The dataset may contain some continuous conditional feature.

        i. Discretize continuous features of the dataset.

        ii. Calculate the elementary sets (make partition according to conditional and
           decisional features).

        iii. Calculate the positive region to eliminate the inconsistent cases.

        iv. Select 100$p$% instances from the positive region and save their labels in
            a list L.

        v. Extract specific cases from the original dataset according to list L.

Output: The set of cases to be selected.
```

Figure 5.2. Algorithm for case selection using Rough sets.

Sampling the dataset randomly can be improved by using a structured algorithm, Rough sets criteria applied for this purpose can be effective for avoiding the inconsistent instances.

After the case selection process, two classifiers, LDA and KNN are applied on the new dataset. The datasets used in this chapter are the ones used in previous chapters.

5.4 Results and Discussion

The experiments were carried out on 10 datasets. First, the dataset was discretized following two approaches: a) our proposed Rough sets method, and b) Entropy method. Then, the case selection algorithm was applied.

To prove the goodness of this algorithm we selected 70 percent of the data as training sample, and 30 perent as test sample. The misclassification error was calculated using the test sample; this process was repeated ten times. Finally, the average misclassification error on ten random samples (test samples) representing a 30% of the original sample was considered. The data were discretized using two methods; the rough discretization and entropy-based dsicretization. To evaluate the effect of the algorithm, the misclassification error before and after instance selection considering the KNN classifier on the discretized data was computed. The results are presented in table 5.1.

Table 5.1: Misclassification error rate for the KNN classifier before and after case selection.

	Rough Disc.		Ent. Disc.	
	Before	After	before	After
Iris	0.0333	0.0488	0.0377	0.0533
Sonar	0.2419	0.3096	0.2129	0.3290
Heartc	0.1505	0.1573	0.1662	0.1978
Ionosphere	0.1542	0.1514	0.1638	0.1676
Crx	0.3107	0.3364	0.3159	0.3333
Breastw	0.0372	0.0362	0.0303	0.0490
Diabetes	0.2782	0.2886	0.2726	0.2908
Vehicle	0.3494	0.3762	0.3613	0.4221
Glass	0.3781	0.4093	0.3656	0.5593
MEAN	0.2148	0.2348	0.2140	0.2669

When rough discretization is used there is not much change on the misclassification error rate after the instance selection process. The greater change occurs for the sonar dataset. However, when entropy based discretization is used , there is a significant change on the misclassification error rate. In particular, for sonar, vehicle and glass datasets.

Table 5.2 is similar to Table 5.1, but using the LDA classifier. The behavior of this classifier after instance selection is more stable compared to the KNN classifier. Once again, performing instance selection after Rough discretization gives better results than doing so after entropy-based discretization.

Table 5.2: Misclassification error rate for the LDA classifier before and after case selection.

Dataset	Rough Before	Rough After	Ent. Before	Ent. After
Iris	0.0266	0.0311	0.0311	0.0266
Sonar	0.3193	0.3354	0.2645	0.3967
Heartc	0.1506	0.1528	0.1820	0.2022
Ionosphere	0.1657	0.1647	0.1580	0.1638
Crx	0.1497	0.1517	0.1292	0.1312
Diabetes	0.2295	0.2360	0.2321	0.2447
Vehicle	0.2260	0.2347	0.2185	0.2351
Glass	0.3781	0.4421	0.4156	0.4796
MEAN	0.2056	0.2185	0.2038	0.2350

Geng and Hamilton (2002) [20] carried out an experimental study similar to ours, but they used decision trees and neural networks classifiers. Four or our datasets; *Iris*, *Breastw*, *Glass*, and *Diabetes*, appear in Geng and Hamilton's study. Regarding the misclassification error, our results are comparable to theirs, except for Glass, where our result is not good. With respect to the percentage of reduction of instances, we have pre-determined to select only a 60% of the positive region. Therefore, the data reduction process leave us with about 60% of the original data, whereas using the Geng and Hamilton's methodology the new dataset is only 21% of the original data.

Six of the datasets used in this thesis: *Iris*, *Breastw*, *Glass*, Sonar, Ionosphere, and Vehicle appear also in the Salamó and Golabardes. Comparing our results with them, we get similar results regarding the percentages of instances selected. Theirs proposed technique called AccurCM selects about 54% of cases. With respect to the misclassification error, for *Iris*, and Vehicle, we get better results than them, but for *Sonar* and *Glass* our results are not good compared to them.

The next two tables show the misclassification error before and after feature selection but performing first feature selection with the two algorithms described in Chapter 4. Only our proposed discretization method based on Rough sets is used.

Table 5.3: Misclassification error rate after ranking feature selection for the KNN and LDA classifier before and after case selection considering Rough Set method discretization.

Dataset	KNN before	KNN after	LDA before	LDA after
Iris	0.0400	0.0400	0.0178	0.0311
Heartc	0.3404	0.3516	0.2876	0.2808
Bupa	0.3669	0.3689	0.2970	0.3708
Ionosfera	0.1447	0.1676	0.1752	0.1771
Diabetes	0.2834	0.2817	0.2304	0.2321
Vehicle	0.3684	0.4008	0.2135	0.2269
Glass	0.3406	0.4265	0.4125	0.4750
MEAN	0.2699	0.2916	0.2366	0.2575

For both classifiers the effect on the misclassification error rate after instance selection performing first feature selection based on ranking is quite similar.

Table 5.4. Misclassification error rate after the hybrid feature selection method for the KNN and LDA classifier before and after case selection considering Rough Set method discretization.

	KNN before	KNN after	LDA before	LDA after
Heartc	0.3213	0.3483	0.2505	0.2662
Bupa	0.3970	0.4349	0.3417	0.3650
Ionosfera	0.1295	0.1295	0.1676	0.1752
Diabetes	0.2852	0.3600	0.2539	0.2574
Vehicle	0.3683	0.4003	0.2308	0.2490
Glass	0.3281	0.5625	0.3906	0.4968
MEAN	0.3049	0.3726	0.2725	0.3016

The KNN classifier seems to be more affected than the LDA classifier after performing the hybrid feature selection followed by instance selection.

5.5 Conclusions

For the LDA classifier, the misclassification error rate after instance selection based on rough sets increases more for entropy discretization than for Rough sets discretization.

Similarly, for the KNN classifier, the misclassification error rate after instance selection based on rough sets increases more for entropy discretization than for Rough sets discretization.

The performance of the KNN classifier deteriorates more than the LDA classifier after instance selection based on rough sets.

Performing first feature selection followed by instance selection deteriorates the classifier accuracy. In particular, when the hybrid feature selection method is used.

CHAPTER 6

UNSUPERVISED LEARNING BASED ON ROUGH SETS

6.1 Introduction

A clustering algorithm forms groups trying to minimize the distance of objects belonging to the same group while maximizing the distance of these objects to objects belonging to other groups. Thus, clustering, also known as unsupervised learning, can be considered as an optimization problem.

There are two major types of clustering algorithms: Partitioning and hierarchical algorithms. The principal difference between them relies in the initial step. Hierarchical methods generate a succession of clusters. This structure is represented using trees. There are two approaches for hierarchical clustering algorithms: agglomerative and divisive. Agglomerative hierarchical algorithms begin considering each element as a cluster. At each step, the number of clusters is reduced by combining them. These algorithms are considered bottom-up. Divisive hierarchical clustering methods begin by considering that all the objects of the dataset belong to only one cluster. In each step, clusters are divided into two new clusters, increasing in this way the number of clusters. These algorithms are considered top-down. In this thesis, we have used a hierarchical agglomerative and a partitioning clustering algorithms. In particular, we have used the function hclust of the R library stats to perform hierarchical agglomerative clustering. The function agnes from the library cluster also performs hierarchical clustering, and the Partitioning Around Medoids (PAM).

Partitioning methods assume beforehand a given number k of clusters. Then, each observation is assigned iteratively to each cluster until a stopping criterion (i.e. the sum of squares within the clusters) is satisfied. The most frequently used partitioning clustering algorithm are: K-means, Partitioning around medoids (PAM), Self organizing maps (SOM) and clustering based on mixture models. In this thesis, we have use the PAM clustering algorithm.

Unsupervised classification or clustering describes the subdivision of the universal set of all possible categories into a number of distinguishable categories. Clustering methods use a similarity measure [12].

6.2 Rough sets based clustering algorithms

Vinterbo and Ohm (1997) were the first authors to introduce a metric based on rough sets in order to perform clustering [83]. Lingras (2002) combined Rough sets theory along with Genetic Algorithms to perform clustering in web mining [48]. A comparison of classical clustering algorithms with k-means based on rough sets appears in Lingras et al. [49]. An application to market research of a clustering algorithm combining Kohonen networks with Rough sets is detailed in Lingras et al. [51]. More rough sets clustering methodologies and applications can be found in [52]. A rough set-based Hierarchical clustering algorithm for categorical data has been recently proposed by Chen at al [10]. Their algorithm was applied to three datasets containing only discrete attributes.

In generating the description of the main characteristics of each cluster, the lower approximation of a rough set contains objects that only belong to that cluster, and the upper approximation contains objects that may belong to more than one cluster [49,50,52].

In this thesis, an existent clustering algorithm is modified by applying it on representative of elementary sets obtained using Rough set theory. In the proposed algorithm, the clustering process is initialized by considering that the dataset is previously discretized. Then a partition { $E_1, E_2, ..., E_n$ } of elementary sets from the discretized data formed according to conditional features is obtained. Then, a distance matrix is defined using the distances between every pair of representative of the elementary sets obtained using Rough sets, finally a cluster algorithm is applying using this Distance matrix.

6.3 Dissimilarity measures

Dissimilarity Measures are used to find dissimilar pair of objects among X. Let i and j be two observations described by m attributes. Then, the dissimilarity coefficient, $d(i,j)$, is small when objects i and j are alike, otherwise, $d(i,j)$ would be large. A dissimilarity measure needs to satisfy the following conditions:

- $d(i,j) \geq 0$
- $d(i,i) = 0$
- $d(i,j) = d(j,i)$
- $d(i,j) \leq d(i,k) + d(k,j)$

Most of the clustering algorithms use dissimilarity measures to join, or to separate, objects. Examples of dissimilarity measures:

Euclidean Distance

The Euclidean Distance between point $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ is given by:

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Manhattan Distance

The Manhattan Distance between points $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ is given by :

$$d(x,y) = \sum_{i=1}^{n} | x_i - y_i |$$

Chebychev distance

The Chebychev distance between points $x=(x_1,x_2,...,x_n)$ and $y=(y_1,y_2,...,y_n)$ is given by :

$$D_\infty(x,y)=\max_{1\le i\le m}|x_i-y_i|$$

Minkowski Distance

The Minkowski distance between points $x=(x_1,x_2,...,x_n)$ and $y=(y_1,y_2,...,y_n)$ is given by :

$$D_p(x,y)=\left(\sum_{i=1}^{m}|x_i-y_i|^p\right)^{1/p}$$

This distance includes the Manhattan distance when p=1, the Euclidean distance when p=2 and the Chebyschev distance when p=∞.

Canberra Distance

The Canberra distance between points $x=(x_1,x_2,...,x_n)$ and $y=(y_1,y_2,...,y_n)$ is given by :

$$D_{Can}(x,y)=\sum_{i=1}^{m}\frac{|x_i-y_i|}{|x_i+y_i|}$$

When the x_i and y_i are both zero, then the *i-th* term is considered as 0.

Distance measure between instances containing only discrete atributes

Let $d(a,b)=\sum_{j=1}^{m}\delta(a_j,b_j)$

where

$$\delta(a_j,b_j)=\begin{cases}0 & (a_j=b_j)\\ 1 & (a_j\ne b_j)\end{cases}$$

$d(a,b)$ indicates the number of distinct components of the vectors *a* and *b*. This is called the Hamming distance. Let **x** and **y** two vectors of the same dimension and with values on Ω={0,1,2,…….k-1}, then the Hamming distance among them is defined as the the number of different entries between the two vectors. For instance, if x=(0,1,3,2,1,0,1) and y=(1,1,2,2,3,0,2) then DH(**x,y**)=4. For binary features, the Hamming distance is the same as the Euclidean and Manhattan distances.

Other more general measure is given by

$$d_{\chi^2}(a,b) = \sum_{j=1}^{m} \frac{(n_{a_j} + n_{b_j})}{n_{a_j} n_{b_j}} \cdot \delta(a_{j,} b_j)$$

6.4 Similarity Measures

Similarity measures are used to find similar pairs of object among X. Given two observations, i and j, two different rows of the data matrix. Let $s(i,j)$ be a similarity coefficient. If objects *i* and *j* are alike, then $s(i,j)$ becomes larger. Otherwise, $s(i,j)$ becomes smaller. For all objects *i* and *j*, a similarity measure needs to satisfy the following conditions:

- $0 \leq s(i,j) \leq 1$
- $s(i,i) = 1$
- $s(i,j) = s(j,i)$

The correlation coefficient is the most well known similarity measure between two instances containing only continuous attributes.

There are plenty of similarity measures for instances containing only nominal attributes, such as the Jaccard-Tanimoto neasure,

6.4.1 Dissimilarity matrix.

Given an information data set (U,A,C,D), we can obtain a matrix containing the dissimilarity between the observations. This is called the Similarity Matrix.

$$\text{Dissim(U, C,D)} = [Dis(a_i, a_j)]_{n\times n}$$

This matrix contains the dissimilarity between two different rows from a matrix that contains representative objects from every element set. Since all elements in an elementary set from a partition are similar to each other, it is not necessary to calculate the distance between all the elements. That is, the clustering process groups elementary sets, making the problem less complex than the original one.

In this thesis, we use a distance matrix, containing the distances among the representative of the elementary sets, as a dissimilarity matrix for the PAM clustering algorithm.

6.5 The PAM (Partitioning Around Medoids) algorithm

PAM is a partitioning clustering method introduced by Kauffman and Rousseeuw [41]. This algorithm is based on the finding of *k* representative objects called medoids. The best partition will be the one minimizing the average dissimilarity of objects to their closest representative object.

The PAM algorithm is divided in two phases: Build and Swap. The first phase consists in choosing *k* representative objects, and the second phase is an attempt to improve the set of representative objects that was selected in the first phase.

In this thesis, we combine the PAM Algorithm with a Rough set criteria to manage inconsistent data. We are proposing to build the clusters using only the representative from each equivalence class.

6.6 A Rough sets-based Clustering Algorithm

The Rough Cluster algorithm (see Fig. 6.1) considers an information system as input. According to the set of conditional features, a partition process is done. Then, some observations are extracted as representative objects for each elementary set. Using these objects, a distance matrix is formed to carry out the PAM algorithm, making this process computationally cheap. We have used the Manhattan and Hamming distances because they are more suitable for discrete value attributes. The output of the algorithm is the cluster membership of all the objects.

Input: An information system (U,A,C)

i. Consider only the set of conditional features C to make the partition $\{ E_1, E_2, ..., E_n \}$. Thus, the elementary sets are formed.

ii. Select an observation from every elementary set.

iii. Form a matrix D according to some distance measure D(i,j) is either the Manhattan distance from i-th elementary set to j-th elementary set or the Hamming distance.

iv. Carry out the PAM algorithm using the matrix D.

v. Finally the formed cluster are clustering of many elementary sets.

vi. Each instance is assigned to a cluster where its representative belongs to.

vii. An external evaluation criterion is used after the clustering process.

Output: List of cluster membership.

Figure 6.1 The Rough sets-based cluster algorithm

Since the time complexity of the partition is O($n^2 \times p$) in the worst case, and the PAM's computation is O($p(n\text{-}p)^2$) , then Rough sets-based cluster algorithm has complexity of order O($n^2 \times p$).

6.7 External Criteria Measures.

External validation measures are used to compare a clustering structure C produced by a clustering algorithm, with a partition P of the data set X drawn independently from the clustering structure C. These measures give the degree of agreement between a predetermined partition P and the proximity matrix (Distance matrix) of X .

In the sequel, we will define some parameters that appear in the cluster validation measures to be used in this thesis.

a : the number of pairs of vectors in X that belong to the same cluster in C and to the same group in partition P.

b: is the number of pairs of vector in X that belong to the same cluster in C and to different groups in P.

c: the number of pairs of vectors that belong to different clusters in C and to the same group in P and ,

d: the number of the pairs that belong to different clusters in C and to different groups in P.

The external validation measures most commonly used are the following:

6.7.1 Rand Index

This index measures the fraction of the total number of pairs that are either in the same cluster and in the same partition, or in different cluster and in different partitions.

$$R = \frac{a+d}{M}$$

where: $M = a+b+c+d$.

The value of this index lies between 0 and 1. Values close to 1 indicate high agreement between the generated clusters and the assumed partition.

6.7.2 Jaccard Coeficient

This measure calculates the proportion of pairs that are in the same cluster and in the same partition with respect to those that are either in the same cluster or in the same partition.

The Jaccard Coefficient is defined by:

$$J = \frac{a}{a+b+c}$$

This index ranges between 0 and 1. High values indicate high agreement between the generated clusters and the assumed partition.

6.7.3 Fowlkes and Mallow Index (FM)

This index is the geometrical mean of two probabilities: the probability that two random objects are in the same cluster given they are in the same group, and the probability that two random objects are in the same group given that they are in the same cluster.

The FM index is defined by:

$$FM = \sqrt{\left(\frac{a}{a+b}\right)\left(\frac{a}{a+c}\right)}$$

Once again a index value close to 1 indicates a high agreement between generated clusters and the assumed partition.

6.7.4 Hubert's Statistic

This index measures the correlation between the matrices, X and Y, of equal dimension, drawn independently of each other, where X(*i,j*) is equal to 1 if the pair of vector (x_i, x_j) belong to the same group in the partition C, and 0 otherwise. Y(i,j) is defined similarly but using the partition P instead of C. The statistic is defined by

$$\hat{\Gamma} = \frac{Ma - (a+b)(a+c)}{\sqrt{(a+b)(a+c)(M-(a+b))(M-(a+c))}}$$

This index lies between -1 and 1. Values near to 1 indicate high agreement between the generated clusters and the assumed partition.

6.8 Results and Discussion

The data used to prove the Rough set approach to construct the cluster are the same that are those used in other chapters on this thesis. In order to apply the proposed clustering algorithm, we have not considered the column containing the classes. However, for cluster validation, the class column has been considered as the ideal cluster membership for each observation.

The proposed algorithm considers only an observation of each elementary set to construct the clusters. The number of clusters that we have considered in our experiments is the same as the number of classes. We have not attack the problem on finding the optimal number of classes.

The main benefit of the proposed algorithm lies in the reduction of computation time. But it is necessary to validate the cluster formed. For this purpose, the four external criteria described in the previous section were used. Table 6.1 shows the results for the external criteria to evaluate the clusters created using Rough PAM clustering algorithm using Manhattan distance. This table also shows a column containing the Discretization method previously applied, given that the datasets contain some continuous features.

Table 6.1 External measures for Rough Cluster algorithm (PAM)

Dataset	Cluster PAM				
	Discretization Method	Rand	Jaccard	Fandm	Hubert
Iris	Entr	0.9341	0.8180	0.8999	0.8508
Iris	Chi 0.05	0.9656	0.9007	0.9478	0.9221
Iris	Rough	0.8568	0.6481	0.7866	0.6791
Sonar	Entr	0.6999	0.5528	0.7126	0.4013
Sonar	Chi 0.15	0.5006	0.3497	0.5185	0.0011
Sonar	Rough	0.5243	0.3898	0.5626	0.0494
Crx	Rough	0.6114	0.4841	0.6556	0.2271
Diabetes	Ent	0.6106	0.4696	0.6391	0.2164
Diabetes	Rough	0.6057	0.4663	0.6360	0.2060
Vehicle	Ent	0.6797	0.2213	0.3625	0.1486
Vehicle	Rough	0.6466	0.2079	0.3452	0.1047
German	No discret	0.5520	0.4694	0.6403	0.0565

Table 6.2 is similar to table 6.1 but using Hamming distance instead of Manhattan distance.

The Table 6.3 shows the different clustering validation measures using hierarchical clustering algorithm (HCLUST).

Table 6.2. External measures for Rough Cluster algorithm (PAM) Hamming distance

	Cluster PAM				
Dataset	Discretization Method	Rand	Jaccard	Fandm	Hubert
Iris	Entr	0.9341	0.8180	0.8999	0.8508
Iris	Chi 0.05	0.9575	0.8787	0.9355	0.9038
Iris	Rough	0.8154	0.5818	0.7369	0.5969
Sonar	Entr	0.7000	0.5528	0.7126	0.4013
Sonar	Chi 0.15	0.8332	0.7146	0.8335	0.6664
Sonar	Rough	0.5006	0.3689	0.5406	0.0012
Heartc	Entr	0.7327	0.5879	0.7407	0.4660
Crx	Rough	0.7014	0.5429	0.7037	0.4029
Diabetes	Ent	0.5000	0.3685	0.5385	-0.0083
Diabetes	Rough	0.5482	0.4128	0.5843	0.0896
Vehicle	Ent	0.6990	0.2725	0.4289	0.2253
Vehicle	Rough	0.6628	0.2101	0.3476	0.1205
German	No discret	0.5207	0.3966	0.5686	0.0325

Table 6.3 External measures for Rough Cluster algorithm (HCLUST)

	Cluster HCLUST				
Dataset	Discretization Method	Rand	Jaccard	Fandm	Hubert
Iris	Entr	0.8623	0.6646	0.7991	0.6953
Iris	Chi 0.05	0.9575	0.8787	0.9355	0.9038
Iris	Rough	0.8415	0.6120	0.7593	0.6412
Sonar	Entr	0.6819	0.5423	0.7048	0.3678
Sonar	Chi 0.15	0.4993	0.4175	0.5988	-0.0015
Sonar	Rough	0.4993	0.4970	0.7032	-0.0062
Crx	Rough	0.5042	0.5025	0.7079	0.0082
Diabetes	Ent	0.5491	0.4412	0.6135	0.0794
Diabetes	Rough	0.5482	0.5440	0.7357	0.0374

Vehicle	Ent	0.5903	0.2759	0.4547	0.1770
Vehicle	Rough	0.5024	0.2390	0.4179	0.0764
German	No discret	0.5836	0.5773	0.7569	0.0521

6.9 Conclusions

After applying the proposed clustering method and observing our experimental results we can conclude that:

i) Constructing clusters considering only one observation from each elementary set reduces the computation running time of the PAM algorithm, since the clustering process is applied to a smaller number of observations.

ii) The external cluster validation measures for the proposed algorithm on the datasets considered, yield good values (greater than .5), except sometimes for the Hubert measure, which seems to be a very strict measure.

iii) Our experimental study suggest that there is not much difference in applying a Rough set- based PAM algorithm and a Rough set-based Hierarchical clustering algorithm.

iv) Use a Hamming distance along with the Rough set- based PAM algorithm shows better results than Rough set- based PAM algorithm with Manhattan distance.

CHAPTER 7

ETHICS

Introduction

Knowledge discovery is an important area of the data analysis, Rough sets is a mathematical tools that helps the development of several Knowledge discovery methods. Uncertain information systems can be analyzed using algorithm based on Rough sets. Given the diversity of the applications of these methods in different areas such as Bio-engineering, business, financial, agriculture, chemistry and Biology, these methods could be used in a wrong way and their results can be missleading.

Ethics in science

Ethics is a subject studied in the literature of philosophy specially the philosophy of information technologies, like Kant's categorical imperative. He presented his view on how and why something may be considered moral, and he stated the following quote “Act only according to that maxim by which you can at the same time will that it would become a universal law." [25]. When the world is view or analyzed, we know *a priori* that morality is universal and necessary then, applied science according to ethical rules is a moral thing.

Ethics in the analysis of information process

Data gathering is very important in the Knowledge discovery process; therefore, much professionalism is required during the handling of the data. Thus, one must try not to wrong handling data, because such practice could lead to inappropriate results.

Thus is, the researcher has responsibilities about their data analysis, interpretation of results, and publications [6]. Hence, it is necessary to be careful on the research process until the task is finished and published.

Ethics in this thesis

Rough set philosophy is founded on the assumption that every object of the universe set is associated with some information Knowledge. Many applications of rough sets in different areas has been carried in the last twenty years, some of them has been in medicine [7], economy, genetic, biology, artificial intelligence and many more. Comparisons with other methods gave good results, and research in this area has continued.

In this work, many algorithms have been proposed according to each application. All the algorithms presented in this thesis have been tested on real data, Therefore, the results are interpretation of real problems coming from different fields. These data sets are commonly used in data mining for this kind of tasks.

If data sets were simulated then we could create an ideal situation for Rough sets theory but their interpretation would not be useful in the real world, in particular if the simulated data set is of lower dimension and has an small number of instance. However, simulation can be very useful to set up a worst case situation.

CHAPTER 8

CONCLUSIONS

This thesis covers mostly data preprocessing steps of the KDD process. Algorithms to carry out such tasks were done using mathematical tools from Rough set theory. These algorithms were applied to datasets that have been used extensively in the literature of data mining and Knowledge discovery.

The following conclusions are obtained:

i. Rough set is a good option to process the information for KDD methods
ii. Discretization based on Rough sets theory compares well with other discretization methods. The computational speed of the algorithm compares vey well with other discretization methods.
iii. Feature Selection using Rough sets theory is a way to identify relevant features. Only features having a large dependency with the decisional attribute are considered relevant.
iv. Case selection using Rough sets concepts shows good results. This is validated by the improvement on the performance of some classifiers.
v. Cluster methods using Rough set theory reduce the computational burden of the clustering algorithm because the elementary sets as treated as observations.

CHAPTER 9

FUTURE WORK

- Other kind of classifiers such as decision trees and neural networks can be combined with the feature selection methods purposed on this thesis
- Consider a more sophisticated sampling method instead of the random sampling used in the instance selection process. Also, we can study the effect, of choosing a given percentage of instances from the positive region, on the performance of a given classifier.
- Use a database software such as SQL or Oracle to perform the algorithms purposed on this thesis.
- Apply Rough sets methodology to multi-relational tables.
- Make a more extensive comparison of the methods purposed on this thesis with other Rough sets methods already existing.

REFERENCES

1. Acuña, E. (2003). A comparison of filters and wrappers for feature selection in supervised classification. Proceedings of the Interface 2003 Computing Science and Statistics. Vol 34.

2. Acuña, E., and Rodriguez, C. (2004). Dprep: Data preprocessing and visualization functions for classification. R package 1.0. http://math.uprm.edu/~edgar/dprep.html. accessed on February 17, 2006.

3. Aha D. W. and Wettschereck D. (1997) *Case-based learning: Beyond classification of feature vectors*. In Proceedings of the 9th European Conference on Machine Learning (ECML'97). Springer.

4. Aha. D. W. (1998) The omnipresence of case-based reasoning in science and applications. KnowledgeBased Systems, 11(5-6):261—273.

5. Ambroise, C. and McLachlan, G. (2002). Selection bias in gene extraction on the basis of microarray gene-expresion data. Proc Natl Acad Sci U S A, pp. 6562 - 6566.

6. American Statistical Association web page at:
http://www.amstat.org/profession/index.cfm?fuseaction=ethicalstatistics.
accessed on January 15, 2006.

7. Badeeh, A., Roudshdy, M. and Mahmoud, S. (2004). Mining patient data based on rough set theory to determine thrombosis disease. In The International Journal of Artificial Intelligence and Machine Learning. Pp. 23-27.

8. Bazan J., Szczuka M., and Wróblewski J., (2002). A new version of rough set exploration system (RSES). Rough Sets and Current Trends in Computing. Proc. of 3rd International Conference RSCTC 2002, Malvern, PA, USA. pp. 397 - 404.

9. Cano, J., Herrera F., and Lozano, M. (2003). Using evolutionary algorithm as instance selection for data reduction in KDD: An experimental study. In IEEE transactions on evolutionary computation. Vol. 7. No. 6. pp. 561-575.

10. Chen, D., Cui, D., Wang, C., and Wang, Z. (2006). A Rough Set-Based Hierarchical Clustering Algorithm for Categorical Data International Journal of Information Technology, Vol.12, No.3, 149-159.

11. Chi, N. (2003). A tolerance rough set approach to clustering web search result. Master thesis in computer science. Institute of Mathematics, Warsaw University.

12. Charikar, M. and Panigrahy, R. (2004) Clustering to minimize the sum of cluster diameter. J. Comput. Syst. Sci. 68(2): 417-441.

13. Chmielewski, M. R, and Grzymala-Busse, J. W. (1996) Global discretization of continuous attributes as preprocessing for Machine learning. Int. Journal of Approximate Reasoning, 15, pp 319-331.

14. Coaquira F., and Acuna, E. (2007). Applications of Rough sets theory to data preprocessing in Knowledge Discovery. To appear in Proceedings of the conference of Machine learning and data analysis to be held October 2007 at UC Berkeley, California.

15. Deogun, J., Choubey, S., Raghavan, V., and Sever, H. (1998). Feature selection and effective classifiers. Journal of ASIS 49, 5, 403-414.

16. Dougherty, J. Kohavi, R. and Sahami, M. et al (1995). Supervised and Unsupervised Discretization of Continuous Features. Proceeding of twelfth International Conference, Morgan Kaufmann Publishers. pp. 194 – 202.

17. Duntsch, I. and Gediga, G. (1998) Simple data filtering in rough set systems, International Journal of Approximate Reasoning pp. 93-106.

18. Duntsch, I. and Gediga, G. (2000). Rough set data analysis. In Encyclopedia of Computer Science and Technology, Marcel Dekker. 282-301.

19. Gediga, G. and Düntsch, I. (2003). Maximum consistency of incomplete data via non-invasive imputation. *Artificial Intelligence Review*, pp. 93-107.

20. Geng, L. and Hamilton, H.J. (2002). ESRS: A Case Selection Algorithm Using Extended Similarity-based Rough Sets. Second IEEE International Conference on Data Mining (ICDM'02)

21. Grochowski, M. and Jankowski, N. (2004) Comparison of the instance selection algorithms. II.Results and comments. In: ICAISC 2004, L. Rutkowski et al. (Eds.), LNAI 3070, pp. 580-585, 2004

22. Grzymala, J. and Ziarko, W. (2000). Data mining and rough set theory. Communications of the ACM.

23. Grzymala, J. and Siddhave, S. (2004). Rough set Approach to Rule Induction from Incomplete Data. Proceeding of the IPMU'2004, the10th International Conference on information Processing and Management of Uncertainty in Knowledge-Based System.

24. Gupta, D. (1988). Rough sets and information retrieval. Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 567 – 582.

25. Guthrie, S. (2001). Immanuel Kant and the Categorical Imperative. Published in The examined Life On-Line Philosophy Journal.

26. Guyon, I, and Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, pp 1157-1182.

27. Hall, M. (2000). Feature Selection for Discrete and Numeric Class Machine Learning. Proc. Seventeenth International conference on Machine Learning, San Francisco, CA, Morgan. Kaufmann, pp. 359-366.

28. Han,J. and Kamber,M. (2006) Data Mining: Concept and Techniques, 2nd ed. The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor Morgan Kaufmann Publishers, San Francisco, CA.

29. Han, J., Hu, X., Lin, T. Y.: Feature Subset Selection Based on Relative Dependency between Attributes. Rough Sets and Current Trends in Computing 2004: 176-185.

30. Hernández, J., Ramirez, M. and Ferri, C. (2004) "Introducción a la mineria de datos" Pearson Prentice Hall. Madrid – España.

31. Hoa S., Nguyen, H. Son Nguyen (1998). Pattern extraction from data. Fundamenta Informaticae, pp. 129-144.

32. Hu, X, and Cercone, N. (1994) Discovery of decision rules in international databases: a rough set approach. Proceedings of the third international conference on Information and Knowledge management.

33. Hu, K., Lu, Y and Shi C., (2003). Feature ranking in Rough sets. AI Communications, pp 41-50.

34. Jain.A.K, Duin, R.P.W., and Mao, J. (2000). Statistical Pattern Recognition: A Review. IEEE Transactions on pattern analysis and Machine Intelligence, pp. 4 - 37.

35. Jankowski N, Grochowski M (2004) *Comparison of Instances Selection Algorithms I. Algorithms Survey* In: ICAISC 2004, L. Rutkowski et al. (Eds.), LNAI 3070, pp. 598–603, 2004 and pp. 580-585 .

36. Jensen, R. and Shen, Q. (2003) Rough and Fuzzy sets for dimensionality reduction. *Proceedings of the 2001 UK Workshop on Computational Intelligence*, 69-74.

37. Jensen, R. and Shen, Q. (2003) Finding Rough Set Reducts with ant colony optimization. Proceeding of the 2003 UK Workshop on Computing Intelligence, pp. 15-22.

38. Jensen, R. and Shen, Q. (2004) Fuzzy-Rough Attribute Reduction with Application to Web Categorization. Fuzzy Sets and System, Vol. 141, No 3, pp. 469-485.

39. John, G. (1994). Irrelevant Feature and the subset selection problem. In machine Learning: Proceeding of the Eleventh International Conference. In Morgan Kaufmann Publisher, pp. 121-129.

40. Kantardzic Mehmed (2001) Data Mining Concepts, Models, Methods, and Algorithms.

41. Kaufman, L. and Rousseeuw, P (1990). Finding groups in data: An introduction to cluster analysis. John Wiley & Sons. New York.

42. Kohavi, R. (1994). A third dimension to rough sets. In proceeding of the Third international workshop on rough sets and soft computing. pp. 244-251.

43. Kohavi, R. and Frasca, B. (1994). Useful Feature Subsets and Rough Set Reducts. In proceeding of the Third International Workshop on rough sets and soft computing. San Jose, California, pp. 310-317.

44. Kononenko, I., Simec, E. and Robnik M. (1997). Overcoming the myopia of induction Learning algorithms with Relief. Aplied Intelligence. pp. 39-55.

45. Kusiak, A. (2001). Rough Set Theory: A Data Mining Tool for Semiconductor Manufacturing. IEEE Transactions on electronics Packaging Manufacturing.

46. Lesh N., Zaki M, and Ogihara M., *"Mining Features for Sequence Classification," 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, San Diego, CA, August 1999.

47. Lin, T. (1995) Introduction to the special issue on Rough Sets and Soft Computing. *Journal of Approximate Reasoning*, Vol 15, No 4, 1996, 395-414.

48. Lingras, P., (2002) Rough set clustering for web mining. In Proceedings of World Congress on Computational Intelligence, IEEE International Conference on Fuzzy Systems FUZZ-IEEE(02) Special Session on Computational Web Intelligence (CWI), Honolulu, Hawaii, USA pp. 1039-1044.

49. Lingras, P. and Yan, R., and West, C. (2003). Comparison of Conventional and Rough K-Means Clustering, Proceedings of the 9th. International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, Chongqing, P. R. China, Lecture Notes in Artificial Intelligence Series, 2639, Springer, pp.130-137.

50. Lingras, P., Yan, R., and Hogo, M. (2003). Rough Set Based Clustering: Evolutionary, Neural, and Statistical Approaches, Proceedings of the First Indian International Conference on Artificial Intelligence, pp. 1074-1087.

51. Lingras, P., Hogo, M., Snorek, M., and Leonard, B. (2003). Clustering Supermarket Customers using Rough Set Based Kohonen Networks, Proceedings of Fourteenth

International Symposium on Methodologies for Intelligent Systems, Maebashi City, Japan, October 28-31, 2003, Lecture Notes in Artificial Intelligence Series, 2871, Springer, pp.169-173.

52. Lingras P. (2007). Applications of Rough Sets based K-means, Kohonen, SOM, GA Clustering. Transaction on Rough Sets. pp 120-139.

53. Liu, H., Hussain, F., Lim, C. and Dash, M. (2002). Discretization: An Enabling Technique. Data Mining and Knowledge Discovery, 6(4), pp. 393 - 423.

54. Liu, H. and Motoda, H. (2002) On issues of instance selection. In Data Mining and Knowledge Discovery. pp.115-130.

55. Liu, H. and Setiono, R. (1996). A probabilistic approach to feature selection – a filter solution. Proc. International Conference of Machine Learning, pp. 319 -337.

56. Liu, H. and R. Setiono. (1997). Feature Selection via Discretization of Numeric Attributes, IEEE Trans. Knowledge and Data Engineering, 9 (4), pp.642-645.

57. Mitatha, S. (2003). Some experimental result of using rough sets for printed thai characters recognition. International Journal of Computational Cognition. Pp. 109-121.

58. Mitra,S., Pal, S., Mitra P. (2002). Data mining in soft computing framework: a survey. In IEEE Transactions on Neural Networks. pp. 3-14.

59. Nguyen, S., Nguyen, T., Skowron A. and P. Synak (1996). Knowledge discovery by rough set methods. In: Nagib C. Callaos (eds.), ISAS-96: Proc. of the International Conference on Information Systems Analysis and Synthesis. pp. 26-33.

60. Nguyen, S., Skowron A., Synak P., Wróblewski J., (1997). Knowledge Discovery in Databases: Rough Set Approach., Proceedings of the Seventh International Fuzzy Systems Association World Congress (IFSA'97), pp. 204 - 209.

61. Nguyen, H.S. and S.H. Nguyen. Discretization Methods for Data Mining. In: Rough Sets in Knowledge Discovery, Vol. 1, Chapter 22, ed by L. Polkowski and A. Skowron, pp.451-482. Physica-Verlag. 1998.

62. Nguyen, S., Skowron, A. and Stepaniuk, J. (2001) *Granular Computing: a rough set approach,* Computational Intelligence, pp.514-544.

63. Ohrn, A. (1999). Discernibility and Rough Sets in Medicine: Tools and applications. PhD thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway.

64. Pal, S. and Mitra, P. (2004) Pattern recognition algorithms for data mining: scalability, knowledgediscovery, and soft granular computing. Chapman & Hall / CRC. A CRC Press Company.

65. Pawlak, Z. (1982). Rough Sets, Theoretical Aspects of Reasoning about Data. Boston, MA: Kluwer Academic Publishers, Dordrecht.

66. Pawlak, Z., Grzymala, J., Slowinski, R., and Ziarko, W. (1995). Rough Sets. Communications of the ACM, pp 88-95.

67. Pawlak, Z. (1995). Rough sets and Fuzzy sets. Proceedings of the 1995 ACM 23rd annual conference on computer science. pp 252-254.

68. Peña, J. Létourneau, S., Famili, A..(1999). Application of Rough Sets algorithm to prediction of aircraft component failure. In proceeding of the third International Symposium on Intelligent Data Analysis, Amsterdam, The Netherlands.

69. Polkowski L. , Skowron (1995) A. Introducing Rough Mereological Controllers Rough Quality Control, Soft Computing, T. Y. Lin, A. M. Wildberger (eds.), Simulation Councils, San Diego pp. 240-243.

70. Pudil, P., Ferri, F.J., Novovicova, J., Kittler, J (1994). Floating search methods for feature selection with monotonic criterion function. International Conference on Pattern Recognition. pp. 279 - 283.

71. Salamó, M., Golobrdes, E. (2001) "Rough Sets reduction techniques for Case-Based Reasoning", Case-Based Reasoning Research and Development. Aha, D.W., Watson, I. & Yang, Q. (eds.), Proceedings of the 4th. International Conference on Case-Based Reasoning, ICCBR-01, Vancouver, Canada, 30 July - 2 August 2001, Springer, Lecture Notes in Artificial Intelligence, pp. 467-482, 2001

72. Salamó, M., Golobrdes, E. (2004) "Global, Local and Mixed Case Base Maintenance Techniques ". Sixth Catalan Conference on Artificial Intelligence. Barcelona, Spain, October 2004. Recent advances in Artificial Intelligence Research and Development, Frontiers in Artificial Intelligence and Applications volume 113, pp. 127-134.

73. Sever, H., (1998), "*The Status of Research on Rough Sets for Knowledge Discovery in Databases*", Proceedings of the Second International Conference on Nonlinear Problems in Aviation and Aerospace (ICNPAA98), Daytona Beach, Florida, USA, pp. 673-680.

74. Shen, l. (2002). Data mining techniques based on rough sets theory. Ph.D. thesis, Department of Mechanical Engineering, National University of Singapore. Singapore.

75. Shen, Q. and Chouchoulas, A. (2001) Rough set-based dimensionality reduction for supervised and unsupervised learning. Int. J. Applied Mathematics computational Science. Vol. 11, No. 3 pp. 583-601.

76. Shi, H. and Fu, J. (2005) A global discretization method based on rough sets. Proceeding of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, pp 18-21.

77. Stepaniuk, J. (1996). Rough Sets and Similarity Relations. Symulacja w badaniach rozwoju, Trzecie Warsztaty Naukowe PTSK, Wigry, 26-28 września, (red.) R. Bogacz, L. Bobrowski, Warszawa . pp. 405 - 413.

78. Stepaniuk, J. (1996). Rough sets, discretization of attributes and stock market data. Fourth European Congress on Intelligent Techniques and Soft Computing, Proceedings EUFIT'96 , Aachen, Germany, Verlag Mainz, pp.202-203.

79. Stepaniuk, J. (1998). *Rough set based data mining in diabetes mellitus data table.* Proceedings of the Sixth European Congress on Intelligent Techniques and Soft Computing. pp. 980 - 984.

80. Szczuka, M. (1999), Rules as attributes in classifier construction. Proceedings of RSFDGrC'99, Yamaguchi, Japan, LNAI 1711, Springer Verlag, Berlin, pp. 492 - 499.

81. Tay, F., and Shen, L. (2002).A Modified Chi2 Algorithm for Discretization. IEEE Transactions on Knowledge and Data Engineering. 14 (3), pp. 666 – 670.

82. Upadhyaya S., Arora A., Jain, R. (2006). Rough Set Theory: Approach for Similarity Measure in Cluster Analysis. Proceedings of the 2006 International Conference on Data Mining, DMIN 2006. pp 353-356.

83. Vinterbo, S. and Ohm, A. (1997). Rough sets approach to clustering. Proceedings of the Joint Conference on Information Sciences. Vol 3, pp 383-386.

84. Voges, K.E., and Pope, N.K.Ll, and Brown, M.R.. (2002). Cluster analysis of marketing data examining online shopping orientation: A comparison of *k*-means and rough clustering approaches', in *Heuristics and Optimization for Knowledge Discovery*, eds. H.A. Abbass, R.A. Sarker &C.S. Newton, Idea Group Publishing, Hershey, pp. 207-224.

85. Voges, K. E., Pope, N. K. Ll., and Brown, M. R. (2003). A rough cluster analysis of shopping orientation data. Paper presented at ANZMAC2003: Australian and New Zealand Marketing Academy Conference, December 1-3, 2003, University of South Australia, Adelaide, South Australia.pp. 1625-1630.

86. Wang, X., Yang J. Teng X., Xiang, W., Jensen, R. (2007) Feature selection based on Rough Sets and particle swarm optimization. Pattern recognition Letters 28, pp. 459-471.

87. Wilson, D.R., and Martinez, T.R. (2000). An integrated instance-based learning algorithm. Computational Intelligence, 16 (1), 1-28.

88. Witten, I. and Frank, E. (2000) Data Mining: Practical Machine Learning Tools and Techniques. 2nd ed.

89. Wu, C., Li, M., Han, Z., Zhang, Y., Yue, Y. (2004). Discretization Algorithms of Rough Sets Using Clustering. Proceedings International Conference on Robotics and Biomimetics August 22 - 26, 2004, Shenyang, China.

90. Yao, Y. (1998). Generalized Rough Set Models, In Rough Sets in Knowledge Discovery, Polkowski, L. and Skowron, A. Physica-Verlag, Heidelberg, pp. 386 - 318.

91. Yu, K., Xu, X., Tao, J., Ester, M., and Kriegel, P. (2002). Instance selection techniques for memory-based collaborative filtering. Proc. Second SIAM International conference on Data mining. http://www.siam.org/meetings/sdm02/proceedings/sdm02-04.pdf

92. Zaki, M. and Phoophakdee, B. (2003). Mirage: A framework for mining Exploring and Visualizing Minimal Association Rules. RPI CS Dept Technical Report.

93. Zhong, N. (2001). Using Rough Sets with Heuristics for Feature Selection. Journal of Intelligent Information Systems, pp. 199 - 214.

APPENDIX A. DATA DESCRIPTION

Table A: Data set Description

Data set	Instances	Classes	Features
Iris	**150**	**3**	**4**
Breast	**683**	**2**	**9**
Sonar	**208**	**2**	**60**
Glass	**214**	**6**	**9**
Bupa	**345**	**2**	**6**
Diabetes	**768**	**2**	**8**
Heartc	**297**	**2**	**13**
Ionosphere	**351**	**2**	**32**
Crx	**653**	**2**	**15**
Vehicle	**846**	**4**	**18**

Iris. This is the Iris database, created by R.A. Fisher. It is perhaps the best known database to be found in the pattern recognition literature. The data set contains 3 classes of 50 instances each, where each class refers to a type of the iris flower.

Breast. This Breast cancer databases was obtained from the university of Wisconsin Hospital, Madison. This data set contain 683 instances, 9 predictive features and two classes that represent the type of cancer (benign or malignant).

Sonar. This dataset contains 60 predictor features each one in the range 0.0 to 1.0. Each one of the 208 instances represents the energy within a particular frequency band, integrated over a certain period of time. This data set has two classes (Mines and Rocks).

Glass. This is a Glass identification database. It has originally 214 instances, 9 predictive attributes and 6 classes. Three of the features were eliminated in a cleaning step.

Bupa. It contains information on six attributes of 345 patients. Some of them have hepatitis and others are healthy.

Diabetes. This is the Pima Indians Diabetes Database. It was created by the National Institute of Diabetes and Digestive and Kidney Diseases. The dataset contains 8 features (all numeric-values) and two classes that represent the diagnostic if whether the patient shows signs of diabetes according to World Health Organization criteria. All the 768 patients are female.

Heartc. This dataset contains 297 instances, 13 predictive features and two classes, representing the absence or presence of heart disease. Only features 1,4,5,8,10 are continuous.

Ionosphere. This dataset comes from the classification of radar returns from the ionosphere. This dataset contains 351 instances, 34 predictive features and two classes.

Crx. It is also called the australian credit dataset. It consists of 653 instances and 14 predictors. There are two classes.

Vehicle. It contains information about four types of vehicle. The images were acquired by a camera looking downwards at the model vehicle from a fixed angle of elevation. This dataset contains 846 instances, 18 predictive features and 4 classes.

APPENDIX B: DESCRIPTION OF R FUNCTIONS CREATED IN THIS THESIS

1-numclass(S,data)
Input: S, data
S: is a list of instances
data: is the whole dataset
This function finds the number of classes to which belongs the instances listed on S.
Output: number of classes.

2-Depequals (x, xd)
Input: x,xd
x: is a value
xd: is the column where the value of x is contained
This function finds x-value in xd-column and save a list of target
Output: a list of target

3-Dependencyxy(x,y)
Input: x, y
x and y are two vectors with the same dimension
This function calculates the dependency coefficient between x and y
Output: dependency coefficient.
This function is equivalent to **dependency.** It requires the functions **numclass** and **dpequals.**

4-Discrough(data,varcon)
Input: data, varcon
data: The dataset.
varcon: vector of the continuous feature.
This function discretizes continuous features. For each continuous feature the number of intervals is upper bounded by the number given by the scott's formula.
It requires the function **dependencyxy**
Output: Discretized data set.

5-Dependency(data,col1,col2)
Input: data, col1, col2
data, the dataset
col1 and col2 are two columns of the dataset
This function calculates the dependency coefficient between col1 and col2
Output: dependency coefficient.

6-RoughSelVar(datadisc)
Input: datadisc
datadisc: The discretized dataset.
Output: List of features ordered according to its dependency measure.
It requires the **dependency** function.

7-Roughselvarf(topvar,data)
Input: data, topvar
data: data set.
topvar: vector of top features selected by the ranking method
This function performs the hybrid feature selection method. It considers the vector of top features according to the dependency coefficient and a forward selection method based on the performance of a given classifier.
Output:Feature selected.

8-Positivelist(data)
Input: data
This function finds the list of the instances that lie in the positive region.
Output: the positive list
It requires the **equals** function

9-RRoughSelCase(data)
Input: data
data: The dataset
This function performs instance selection. First, the data is discretized using the rough sets discretization method.
Output: A new dataset containing only the selected instances.
It requires the **positivelist** function.

10-Equals(x,data).
Input: x, data
x: is a vector, that represent an instance of the dataset.
data: is the whole dataset.
This function finds a list of the similar instances through the dataset.
Output:List of similar instances.

11-Positive(data)
Input: data
This function finds the positive coefficient of the data set
Output: a rate of the number of elements and data cardinality.

12-depnumclass(S,data)

Input: S, data
S: is a list of interest target
data: is a new data formed by two column, the second column act bye the decision feature.
This function calculates the number of values of the decision column.
Output: number of values.

13-Errorselcase(data,vcon,repet)
Input: data, vcon, repeat.
data: the dataset.
vcon : vector of continuous features,
repet: the number that the process was repeated.
This function calculates two errors rate: before and after the instance selection process.
The train dataset contains 70 percent of the data and the remaining 30 percent is considered as test dataset. A classifier like LDA or KNN is considered to obtain the error rate.
This process is repeated many times according to the repet option and the average error rate is reported.
Output: average error rate before and after instance selection process.

14-RoughClust(t,data)
Input: data, t
data: data set.
t: number of clusters
This function carries out the cluster process using one representative for each elementary set, a distance matrix is calculate to combine with the PAM cluster process.
Output: data with cluster assignment.

15-RoughSilclust(t,data)
Input: data, t.
data: data set.
t: number of clusters
This function considers the Hclust function to form the new clusters using a represent of each elementary sets.
Output: the sillhouette measure