Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

Rough Sets Reducts as a Filter Approach for Feature Subset Selection: An Empirical Comparison with Wrapper and Other Filters

Adriano Donizete Pila

Maria Carolina Monard/ILTC

Nº 134

RELATÓRIOS TÉCNICOS DO ICMC

USP – São Carlos
Março de 2001

Rough Sets Reducts as a Filter Approach for Feature Subset Selection: An Empirical Comparison with Wrapper and Other Filters*

Adriano Donizete Pila

Maria Carolina Monard/ILTC

University of São Paulo
Institute of Mathematics and Computer Sciences
Department of Computer Science and Statistics
Laboratory of Computational Intelligence
P.O. Box 668, 13560-970 - São Carlos, SP, Brazil
e-mail: {pila, mcmonard}@icmc.sc.usp.br

Abstract The Feature Subset Selection is an important problem within the Machine Learning area where the learning algorithm is faced with the problem of selecting relevant features while ignoring the rest. Rough Sets Theory is a mathematical tool to deal with vagueness and uncertainty information. This theory has been applied to rule induction in Machine Learning problems where the information is given in the form of a *decision table*. One of the main features of this approach are the *reducts*, which is a minimal feature set that preserves the ability to discern each object from the others. This work presents in detail several experiments, results and comparisons using Rough Sets, Wrapper and Filter approaches for the Feature Subset Selection problem. All the experiments where run on real word datasets, most of them obtained from the UCI Irvine Repository.

Keywords: Feature Selection; Rough Set; Data Mining; Machine Learning; Wrapper; Filter.

March 2001

*Work supported by FAPESP (98/16172-3) — www.fapesp.br

Contents

List of Figures

List of Tables

1 Introduction

With the technological evolution, the amount of information that can be gathered and stored increases very rapidly every day. As Artificial Intelligence Systems depend strongly on knowledge, which can be obtained from previous information sources, a problem that has to be faced is how to focus on the most relevant information.

In supervised Machine Learning — ML — an induction algorithm is typically presented with a set of training instances, where each instance is described by a vector of feature values and a class label. The task of the induction algorithm (inducer) is to induce a classifier that will be useful in classifying new cases. If x is an instance of the training set where x is described by a vector of characteristics named $(x_1, ..., x_n)$ and y is the class label, then the inducer must induce a classifier f that will predict the class label of new instances. So, the classifier is a function that maps every x in its correct class label y.

$$y = f(x)$$

One of the main problems in ML is the Feature Subset Selection — FSS — problem, *i.e.* the learning algorithm is faced with the problem of selecting some subset of features upon which to focus its attention, while ignoring the rest (Kohavi and John, 1997).

There are several reasons for doing FSS, such as improving the accuracy of the classifiers, improving the comprehensibility of rules generated by symbolic ML algorithms as well as reducing the cost of processing huge quantity of data.

Basically, there are three approaches in Machine Learning for FSS (Blum and Langley, 1997):

1. Embedded, where the FSS process is embedded within the basic induction algorithm
2. Filter, where the FSS is used to filter the features before the induction process occurs
3. Wrapper, where the induction algorithm is used as a black box, *i.e.* the FSS algorithm exists as a wrapper around the induction algorithm

In (Lee et al., 1999) a set of experiments using the filter and wrapper approaches for FSS are presented. Those experiments were run using nine datasets, most of them from UCI Irvine Repository (Blake et al., 1998).

In this work we included Rough Sets reducts as another filter approach for FSS. Rough Sets is a theory introduced by Zdzislaw Pawlak (Pawlak, 1982) in the early 1980s where the main feature is the *reduct*. A reduct is a minimal subset of features that preserves our ability to discern the examples from each other. In order to compare the results obtained using Rough Sets reduct as a filter approach for FSS, we selected the same datasets, inducers and tools used in (Lee et al., 1999), *i.e.* the inducers $\mathcal{C}4.5$, $\mathcal{C}4.5$-rules, $\mathcal{CN}2$, ID3 implemented in $\mathcal{MLC}$++ as well as the Column Importance facility provided by MineSet™. To find the Rough Sets reducts we selected Rosetta (Øhrn, 1999b) — Rough Set Toolkit for Analysis of Data. This tools presents all functionalities needed to perform several tasks using the Rough Sets approach.

In other words, the objective of this work is to describe and compare the results we obtained by using Rough Sets reducts as another filter approach for FSS with the results obtained by (Lee et al., 1999), which are also presented in this work. The description of those previous results as well as the new results obtained in this work using Rough Sets reducts, closely follows the one used by (Lee et al., 1999).

This work is organized as follows: Section 2 describes some important concepts about the Rough Sets Theory. Section 3 briefly describes each one of the induction algorithms used as black box to the wrapper approach for FSS as well as the algorithms used as filters. Section 4 gives a short description of the datasets used in the experiments. Section 5 shows the experimental setup used to run the experiments and Section 6 describes the results obtained from these experiments. Section 7 reports analysis and comparison of results. Finally, Section 8 gives some conclusions.

2 Rough Sets

This section deals with fundamental issues of the Rough Sets theory, which is a theory in the field of Machine Learning. The theory was introduced by Zdzislaw Pawlak in the early 1980's (Pawlak, 1982), and based on this theory one can propose a formal framework for the automated transformation of data into knowledge. Pawlak has shown that the principles for learning by examples can be formulated in the basis of his theory (Pawlak, 1995; Pawlak et al., 1995; Pawlak, 1996; Komorowski et al., 1999). An important result from the theory is that it simplifies the search for dominating attributes leading to specific properties.

The Rough Set theory is mathematically relatively simple. Despite of this, it has shown its fruitfulness in a variety of knowledge discovery areas. Among these are information retrieval, decision support, machine learning, and knowledge based systems. A wide range of applications utilize the ideas of the theory. Medical data analysis, aircraft pilot performance evaluation, image processing, and voice recognition are a few examples. In this work we present Rough Sets as a support for supervised machine learning problems.

Almost inevitably the database at used for ML will contain imperfection, such as noise, unknown values or errors due to inaccurate measuring equipment. The Rough Set theory comes handy for dealing with these types of problems, as it is a tool for handling vagueness and uncertainty inherent to decision situations as shown in (Komorowski and Øhrn, 1999; Øhrn, 1999a). An advantage of the Rough Sets methodology over the Bayesian approach is that no assumptions about the independence of the attributes are necessary nor is any background knowledge about the data. Because Rough Sets works with uncertainty it has been confused with other theory like the Dempster-Shafer Theory of Evidence and also with the Theory of Fuzzy Sets, but as attested in (Stein, 1993; Szladow and Ziarko, 1993; Yao, 1998) both theories works with uncertainty in different ways.

In this section, a set of definitions from the world of Rough Sets is given.

2.1 Information System

An *information system* consists of a set of *objects* where each object has a number of *attributes* with *attribute values* related to it. The attributes are the same for all objects, but the attribute values may differ. An information system is thus more or less the same as a relational database.

Definition 2.1 (Information System, Decision System) *An* Information System — *IS — is an ordered pair* $\mathcal{A} = (U,A)$ *where* U *is a nonempty finite set of objects — the* Universe*, and* A *is a nonempty, finite set of elements called* Attributes*. The elements of the Universe will in the following be referred to as Objects. Every attribute* $a \in A$ *is a total function* $a : U \rightarrow V_a$*, where* V_a *is the set of allowed values for the attribute (its range).*

A Decision System[1] *— DS — is an IS* $\mathcal{A} = (U,A)$ *for which the attributes in* A *are further classified into disjoint sets of* condition *attributes* C *and* decision *attributes* D*.* $(A = C \cup D, C \cap D = \emptyset)$*.*

An example of a decision system is shown in Table 2.1.1. As one could expect, it is a two dimensional data table. The rows represent objects, while the columns represent attribute values belonging to these objects.

	Attributes			**Decision**
Examples	*Studies*	*Education*	*Works*	*Income*
e_1	no	good	yes	high
e_2	no	good	yes	high
e_3	yes	good	yes	none
e_4	no	poor	no	low
e_5	no	poor	no	medium

Table 2.1.1: Decision System

In this DS there are 5 persons (objects) with attributes reflecting each person situation of life. Assume the intention is to discover rules predicting what degree of income a person gets, depending on attributes describing him or her. The attribute *Income* is therefore selected as a decision attribute (or dependent attribute). The rest of the attributes, *Studies*, *Education*, and *Works* are then the condition attributes (independent attributes). This situation with only one decision attribute is by far the most common, and will be the main focus of this report.

2.2 Discerning Objects

The next definition introduces the concept of an *indiscernibility relation*. If such a relation exists between two objects, it means that all their attribute values are identical with respect to the attributes under consideration, and thus cannot be discerned (distinguished) between when considering those attributes.

[1] In Machine Learning a decision system is called a dataset.

Definition 2.2 (Indiscernibility Relation) *With every subset of attributes $B \subseteq A$ in the IS $\mathcal{A} = (U, A)$, an equivalence relation IND(B) is associated, called an* Indiscernibility Relation*, which is defined as follows:*

$$IND(B) = \{(x, y) \in U^2 \mid \forall a \in B, a(x) = a(y)\} \tag{1}$$

By $U/IND(B)$ is meant the set of all equivalence classes in the relation $IND(B)$.

For the decision system given earlier, a calculation of $U/IND(C)$ gives the following result:

$$U = IND(\{Studies, Education, Works\}) = \{\{e_1, e_2\}, \{e_3\}, \{e_4, e_5\}\} \tag{2}$$

One can see that the objects are grouped together, and that the groups consist of objects that cannot be discerned between when using the selected set of attributes. The classes in tabular form are shown in Table 2.2.2. Class E_1 comes from objects e_1 and e_2, class E_2 from object e_3, while class E_3 comes from objects e_4 and e_5. Note that E_3 has two objects with different decision attribute values.

	Attributes		**Decision**
Examples	*Studies*	*Education*	*Works*
E_1	no	good	yes
E_2	yes	good	yes
E_3	no	poor	no

Table 2.2.2: Classes for B={*Studies, Education, Works*}

2.3 Discernibility Matrix

A *Discernibility Matrix* is a matrix in which the classes are indexes. In the matrix, the (condition) attributes which can be used to discern between the classes in the corresponding row and column are inserted.

Definition 2.3 (Discernibility Matrix) *For a set of attributes $B \subseteq A \in \mathcal{A} = (U, A)$, the* Discernibility Matrix $M_D(B) = \{m_D(i, j)\}_{n \times n}, 1 \leq i, j \leq n = |U/IND(B)|$*, where* $m_D(i, j) = \{a \in B | a(E_i) \neq a(E_j)\} for i, j = 1, 2, ..., n$

The entry $m_D(i, j)$ in the discernibility matrix is the set of attributes from B that discern object classes $E_i, E_j \in U/IND(B)$.

From the previous example, one can observe that the only attribute with a different value for classes E_1 and E_2 is *Studies*. This attribute is therefore placed in its corresponding places in the matrix. Naturally, the matrix will be symmetric due to the fact that the attributes that differ in value for objects *a* and *b*, differ "the other way around" in value for *b* and *a*. Completing the calculation of the discernibility matrix results in the matrix shown in Table 2.3.3[2].

[2]Thus the elements of the discernibility matrix are sets, the notation used in Rough Sets is that shown in Table 2.3.3

	E_1	E_2	E_3
E_1	—	*Studies*	*Education, Works*
E_2	*Studies*	—	*Studies, Education, Works*
E_3	*Education, Works*	*Studies, Education, Works*	—

Table 2.3.3: Discernibility Matrix

If some of the classes have the same decision value, one might decide not to discern between these classes. By doing so, attributes are not added to the matrix for classes with the same decision value. This can result in more simplistic rules if any classes have the same decision value. In the example presented earlier this is not an option, since all classes have different decision values.

2.4 Discernibility Functions

Definition 2.4 (Discernibility Function) *The* Discernibility Function *f(B) of a set of attributes $B \subseteq A$ is*

$$f(B) = \bigwedge_{i,j \in \{1,\ldots,n\}} \bigvee \overline{m}_D(E_i, E_j) \tag{3}$$

where $n = |U/IND(B)|$, and $\bigvee \overline{m}_D(E_i, E_j)$ is the disjunction taken over the set of boolean variables $\overline{m}_D(i,j)$ corresponding to the discernibility matrix element $m_D(i,j)$. The Relative *Discernibility Function f(E,B) of an object class E, attributes $B \subseteq A$ is*

$$f(E, B) = \bigwedge_{j \in \{1,\ldots,n\}} \bigvee \overline{m}(E, E_j) \tag{4}$$

where $n = |U/IND(B)|$.

This implies that the discernibility function $f(B)$ computes the minimal sets of attributes required to discern any equivalence class from all the others. Similarly, the relative discernibility function $f(E, B)$ computes the minimal sets of attributes required to discern a given class E from the others.

For the previous example, the following relative discernibility functions can be calculated:

$$f(E_1, B) = Studies \ \wedge \ (Education \vee Works)$$
$$f(E_2, B) = Studies \ \wedge \ (Studies \vee Education \vee Works)$$
$$f(E_3, B) = (Education \vee Works) \ \wedge \ (Studies \vee Education \vee Works)$$

Definition 2.5 (Dispensability) *An attribute a is said to be dispensable or superfluous in $B \subseteq A$ if IND(B) = IND(B-{a}), otherwise the attribute is* indispensable *in B. If all attributes $a \in B$ are indispensable in B, B is called orthogonal.*

From the example, over the set of classes the attributes values for attributes *Education* and *Works* go hand in hand. Whenever *Education* is *good*, *Works* is *yes*, and whenever *Education*

is *poor*, *Works* is *no*. Thus, $IND(C) = IND(C - \{Education\})$. The only indispensable attribute in our example is *Studies*.

2.5 Reducing Representation

The data in the information system can be used to discern classes only to a certain degree. Not all attributes may be required in order to be able to do so, however. This is why the next definition is helpful.

Definition 2.6 (Reduct, Relative Reduct) *A* Reduct *of B is a set of attributes $B' \subseteq B$ such that all attributes $a \in B - B'$ are dispensable, and IND(B')=IND(B). The term RED(B) is used to denote de family of reducts of B. The set of prime implicants of the discernibility function f(B) determines the reducts of B. The set of prime implicants of the relative discernibility function f(E,B) determines the relative reducts of B. The term RED(E,B) denotes the family of relative reducts of B for an object class E.*

What this implies is that a relative reduct contains enough information to discern objects in one class from all the other classes in the information system. To find the relative reducts for our example, the discernibility functions are employed. Each function is minimized to a sum of products form, as shown below.

$$\begin{aligned} f(E_1, C) &= Studies \ \wedge \ (Education \vee Works) \\ &= (Studies \wedge Education) \ \vee \ (Studies \ \wedge \ Works) \\ f(E_2, C) &= Studies \ \wedge \ (Studies \vee Education \vee Works) \\ &= Studies \\ f(E_3, C) &= (Education \vee Works) \ \wedge \ (Studies \vee Education \vee Works) \\ &= Education \wedge Works \end{aligned}$$

This gives the desired relative reducts. For instance, $RED(E_1, C) = \{\{Studies, Education\}, \{Studies, Works\}\}$.The relative reducts are minimal, because each discernibility function was minimized. A minimal (relative) reduct is thus a reduct in which none of the attributes may be removed without removing the reduct property.

2.6 Upper and Lower Approximation

The next definition is fundamental to the concept of rough sets, since it addresses the central point of the approach, the vague classes. These are the ones with more than one value for the decision attribute.

Definition 2.7 (Lower and Upper Approximation) *The Lower Approximation $\underline{B}X$ and the Upper Approximation $\overline{B}X$ of a set of objects $X \subseteq U$ with reference to a set of attributes $B \subseteq A$ (defining an equivalence relation on U) may be defined in terms of the classes in the equivalence relation, as follows:*

$$\underline{B}X = \bigcup\{E \in U/IND(B) \mid E \subseteq X\}$$
$$\overline{B}X = \bigcup\{E \in U/IND(B) \mid E \cap X \neq \emptyset\}$$

called the B-lower and the B-upper approximation of X, respectively. The region $BN_B(X) = \overline{B}X - \underline{B}X$ *is called the B-boundary (region) of* X[3].

The lower approximation of X is the collection of objects which can be classified with full certainty as members of the set X, using the attributes set B. Similarly, the upper approximation of X is the collection of objects that may possibly be classified as members of the set X . The boundary region comprises the objects that cannot be classified with certainty to be neither inside X, nor outside X, again using the attribute set B. Properties of these approximations are given in (Pawlak, 1996).

2.7 From Reducts to Rules

Rules represent dependencies in the dataset, and represent extracted knowledge which can be used when classifying new objects not in the original information system. After the reducts are found, the job of creating definite rules for the value of the decision attribute of the information system is practically done. To transform a reduct (relative or not) into a rule, one only has to bind the condition attribute values of the object class from which the reduct originated to the corresponding attributes of the reduct. Then, to complete the rule, a decision part comprising the resulting part of the rule is added. This is done in the same way as for the condition attributes. The rules in our example are as follows.

E_1 : $Studies = no \wedge Education = good \longrightarrow Income = high$
$Studies = no \wedge Works = yes \longrightarrow Income = high$
E_2 : $Studies = yes \longrightarrow Income = none$
E_3 : $Education = poor \longrightarrow Income = ?$
$Works = no \longrightarrow Income = ?$

The "rules" derived with basis in E_3 do not specify the resulting attribute value for *Income*, since it is not the same for all the objects in the class. It may therefore be called it a vague category. A better way of presenting this than through a question mark would be to say e.g. that if *Education* is *poor*, then there is a 50% chance that *Income* is *low*, and that there is a 50% chance that *Income* is *medium*.

If a new object is considered for classification, *i.e.* without decision value, one could attempt to determine this value by using the previously generated rules. If exactly one rule which fits is found, the classification is straightforward. This also implies that the object is in the lower approximation of the class to which it is classified to belong to. For objects contained in the boundary region of different classes, no such consistent decision can be made. Some results of rule induction using rough sets theory can be obtained in (Hu and Cercone, 1994; Hu, 1995).

[3]The B is related to the subset B of attributes of A. If another subset was chosen, for instance $F \subseteq A$, the corresponding names of this relation would be *F-lower approximation, F-upper approximation and F-boundary region.*

After this brief introduction on Rough Sets concepts, it follows a description of the inducers and tools used in this work.

3 Inducers and Tools

The following inducers, also found in the $\mathcal{MLC}$++ library (Kohavi et al., 1996), have been used in this work:

1. ID3
2. $\mathcal{C}4.5$ and $\mathcal{C}4.5$-rules
3. $\mathcal{CN}2$

These inducers are well known in the ML community and belong to the eager learning approach. Besides these inducers, it has also been used a tool named "Column Importance facility" — CI provided by MineSet™ from Silicon Graphics.

To find the Rough Sets reducts we used Rosetta (Øhrn, 1999b) — Rough Set Toolkit for Analysis of Data.

The next sections describe the data format used as input to the inducers and tools, a short description of each inducer[4], as well as the CI facility and the Rosetta Rough Set tool.

3.1 Data Format

Normally in the Machine Learning field the inducers use a set of training instances where each instance consists of a vector of feature values and a class label[5]. Generally this vector denoted by $(\mathbf{X}, Y)$ is in the attribute-value format.

Table 3.1.1 illustrates this organization where a row i refers to the i-th example or instance $\mathbf{X}_i$ and column entries x_{ij} refer to the individual value of the j-th feature f_j of instance i. The column labelled as *class* refers to the label or classification of that instance.

f_1	f_2	...	f_m	*class*
x_{11}	x_{12}	...	x_{1m}	y_1
x_{21}	x_{22}	...	x_{2m}	y_2
...	...	...	...	...
x_{n1}	x_{n2}	...	x_{nm}	y_n

Table 3.1.1: Feature-Value or Spreadsheet Format

[4]More details about the inducers can be found in (Lee et al., 1999).

[5]The class label value can be either discrete or continuous where the problem are called *classification* and *regression*, respectively.

By default each dataset recognized by $\mathcal{MLC}$++ needs three separated files with extensions *data*, *test* and *names* where the *data* and *test* files contain labelled instances of the training and test set respectively. The *names* file defines the scheme that allows parsing these two previous files besides the name and domain for each attribute and for the label. The accuracy of the classifier produced by the inducer is measured on unseen data *i.e.* the test set. More details can be found in (Kohavi et al., 1994; Felix et al., 1998).

3.2 ID3

ID3 (Quinlan, 1986) is member of a more general Machine Learning inducers family named Top Down Induction of Decision Trees – TDIDT — and it is a very basic decision tree algorithm with no pruning where a greedy search is conducted and the the algorithm never backtracks to reconsider earlier choices.

3.3 $\mathcal{C}4.5$

$\mathcal{C}4.5$ (Quinlan, 1993) is one of the ID3 successors. Many extensions to the basic ID3 algorithm were added, such as improving computational efficiency, handling continuous attributes, handling training data with missing attribute values, use of windowing — *i.e.* growing several trees — and the use of the gain ratio criterion, instead of the gain criterion used in the original version of ID3, to choose the attribute upon which the test will be applied.

3.4 $\mathcal{C}4.5$-rules

$\mathcal{C}4.5$-rules (Quinlan, 1993) examines the original decision tree produced by $\mathcal{C}4.5$ and derives from it a set of rules of the form $L \rightarrow R$. The left-hand side L is a conjunction of attribute-based tests and the right-hand side is a class. One of the classes is also designated as a default.

It is important to note that $\mathcal{C}4.5$-rules does not simply rewrite the tree to a collection of rules. In fact, it generalizes the rules by deleting superfluous conditions — *i.e.* irrelevant conditions that do not affect the conclusion — without affecting its accuracy, leaving the more appealing rules.

3.5 $\mathcal{CN}2$

The $\mathcal{CN}2$ (Clark and Niblett, 1987; Clark and Niblett, 1989; Clark and Boswell, 1991) is a Machine Learning algorithm that induces '*if* <complex> *then* <class>' rules in domains where there might be noise. Each <complex> is a disjunction of conjunctions.

For unknown nominal feature values, $\mathcal{CN}2$ uses the method of simply replacing unknown values with the most commonly occurring value. For continuous features, the mid-value of the most commonly occurring sub-range replaces the unknown value.

3.6 CI

CI is a "column importance facility" provided by MineSet™ from Silicon Graphics[6]. It is useful for determining how important various features are in making a particular classification.

Basically, CI uses a measure called "purity", which assigns a number from 0 to 100 that describes how important the columns (features) are in making a classification.

3.7 Rosetta

Rosetta (Ølhrn, 1999b) — Rough Set Toolkit for Analysis of Data — is a tool developed as a cooperative effort involving the Knowledge Systems Group at NTNU, Norway, and the Logic Group at Warsaw University, Poland. The kernel architecture, GUI front-end and computational kernel were designed and implemented at NTNU by Aleksander Ølhrn.

This tool presents all functionalities needed to perform some tasks using the Rough Sets approach. Methods for discretization, finding reducts, rules induction and cross-validation are provided.

Using this tool for a dataset it is possible to compute reducts relatively to the entire decision table or relatively to some example. In the first case the inducer computes a minimal attribute subset that preservers the ability to discern each object from the others. In the second case the inducer computes a set of reducts where each reduct is related to an example from the decision table. Note that some reducts have support for multiple examples.

When computing reducts in this two ways the number of rules are different because computing reducts relatively to some example produces a very high amount of rules with different number of antecedents, although there is a way to filter them out.

4 Datasets

Experiments were conducted on several real world domains. Most datasets are from the UCI Irvine Repository (Blake et al., 1998), except Smoke and TA datasets. This two datasets can be obtained respectively from

- http://lib.stat.cmu.edu/datasets/csb/ and
- http://www.stat.wisc.edu/p/stat/ftp/pub/loh/treeprogs/datasets/.

To assist comparisons, the datasets chosen also have different type of attributes. They involve continuous attributes, either alone or in combination with nominal attributes, as well as unknown values. Section 4.2 summarizes datasets characteristics. It follows a basic datasets description.

[6] http://www.sgi.com

4.1 General Description

As all datasets used in this work are described in detail in (Lee et al., 1999), a more simple description is presented here.

TA This dataset consists of evaluation of teaching performance over 3 regular semesters and 2 summer semesters of 151 teaching assistant assignments at the Statistics Department of the University of Wisconsin – Madison.

Bupa This dataset consists of predicting whether or not a male patient has liver disorders based on various blood tests and the amount of alcohol consumption.

Pima In this dataset all patients are females at least 21 years old of Pima Indian heritage living near Phoenix, Arizona, USA. The problem is to predict whether a patient would test positive for diabetes.

Breast-cancer2 This dataset is one of the breast cancer datasets at UCI, where the problem is to predict the recurrence or not of breast cancer.

CMC The examples in this dataset are married women who were either not pregnant or do not know if they were at the time of the interview. The problem is to predict the current contraceptive method choice (none, long-term methods or short-term methods) of a woman based on her demographic and socio-economic characteristics.

Breast-cancer In this dataset the problem is to predict whether a tissue sample taken from a patient's breast is malignant or benign.

Smoke This survey dataset is concerned with the problem of predicting attitude toward restrictions on smoking in the workplace (prohibited, restricted or unrestricted) based on by-law-related, smoking-related and sociodemographic covariates.

Hepatitis This dataset is for predicting life expectation of patients with hepatitis.

Hungaria This dataset is for diagnosing heart diseases.

4.2 Datasets Summary

Table 4.2.1 summarizes the datasets employed in this study. It shows, for each dataset, the number of instances (#Instances), number and percentage of duplicate (appearing more than once) or conflicting (same attribute-value but different class) instances, number of features (#Features) continuous and nominal, class distribution, the majority error and if the dataset have at least one missing value[7].

Datasets are presented in ascending order of the number of features, as will be in the remaining tables and graphs. Figure 4.2.1 shows datasets dimensionality, *i.e.* number of features and number of instances of each dataset. Observe that due to large variation, the number of instances in Figure 4.2.1 is represented as $\log_{10}$(#Instances).

[7]This information has been obtained using the $\mathcal{MLC}$++ *info* utility.

Dataset	# Instances	#Duplicate or conflicting (%)	# Features (cont.,nom.)	Class	Class %	Majority Error	Missing Values
ta	151	45 (39.13%)	5 (1,4)	1 2 3	32.45% 33.11% 34.44%	65.56% on value 3	N
bupa	345	4 (1.16%)	6 (6,0)	1 2	42.03% 57.97%	42.03% on value 2	N
pima	769	1 (0.13%)	8 (8,0)	0 1	65.02% 34.98%	34.98% on value 0	N
breast-cancer2	285	2 (0.7%)	9 (4,5)	recurrence no-recurrence	29.47% 70.53%	29.47% on value no-recurrence	Y
cmc	1473	115 (7.81%)	9 (2,7)	1 2 3	42.70% 22.61% 34.69%	57.30% on value 1	N
breast-cancer	699	8 (1.15%)	9 (9,0)	2 4	65.52% 34.48%	34.48% on value 2	Y
smoke	2855	29 (1.02%)	13 (2,11)	0 1 2	5.29% 25.18% 69.53%	30.47% on value 2	N
hungaria	294	1 (0.34%)	13 (13,0)	presence absence	36.05% 63.95%	36.05% on value absence	Y
hepatitis	155	0 (0%)	19 (6,13)	die live	20.65% 79.35%	20.65% on value live	Y

Table 4.2.1: Datasets Summary Descriptions

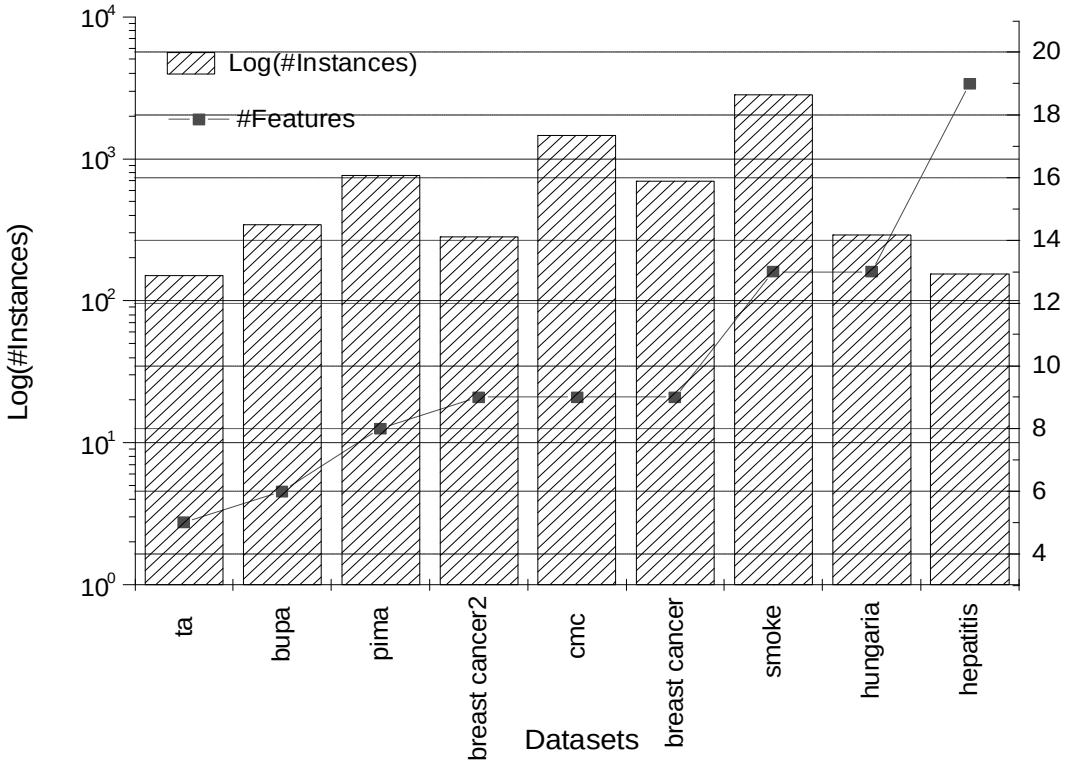

Figure 4.2.1: Datasets Dimensionality

5 Experimental Setup

A series of experiments were performed, using the algorithms and datasets described respectively in Sections 3 and 4. It is important to observe that all results obtained without the use of Rosetta, *i.e.* selected features using ID3, $\mathcal{C}4.5$ and CI as well as 10-fold cross-validation errors using those selected features in $\mathcal{C}4.5$, $\mathcal{C}4.5$-rules and $\mathcal{CN}2$ were extracted from (Lee et al., 1999).

It is also important to note that the original data has not been pre-processed in any way trying to remove or replace missing values or transform continuous attributes in categorical attributes. Furthermore, wrapper inducers as well as each individual inducer were run with default setting for all parameters, *i.e.* no attempt was made to tune any inducer.

For each approach, the performed experiments from (Lee et al., 1999) as well as the ones using Rough Sets reducts as filters, can be divided into two main steps — Figure 5.1:

1. The first step runs the wrapper approach using $\mathcal{C}4.5$, $\mathcal{C}4.5$-rules and $\mathcal{CN}2$ as black box; also in this step $\mathcal{C}4.5$, ID3, CI and Rosetta are used as filters
2. The second step uses features selected by the wrapper in step 1 to compute the accuracy for each one of the inducers used as black box; also filter selected features in step 1 are used to compute the accuracy for $\mathcal{C}4.5$, $\mathcal{C}4.5$-rules and $\mathcal{CN}2$ inducers

A more detailed description of the experiments using wrapper as filter can be found in (Lee et al., 1999).

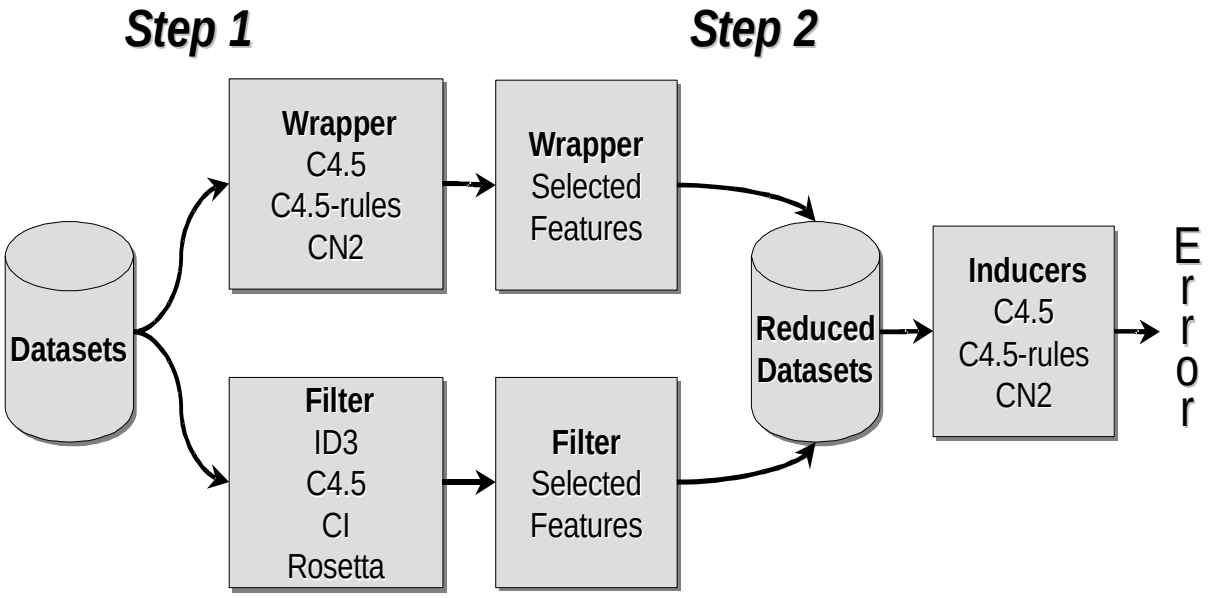

Figure 5.1: Experiments Steps

The filter process was conducted as follows: ID3, $\mathcal{C}4.5$, CI and Rosetta were applied as filters for all the datasets described earlier.

It is important to note that when using Rosetta as a filter the result is a set of subsets where each subset is a set of selected features (reducts) and there can be several reducts[8]. Rosetta has a default setting to compute a set of reducts where all resulting reducts have the same ability to discern the examples from each other. So each reduct is a subset of selected features where the number of selected features may be different.

In this work we decided to select as filter the reduct with the smallest number of features, *i.e.* if Rosetta brought up five different reducts that preserve the same indiscerbility relation of the entire set of features in the dataset, we selected the reduct with less number of features — thus introducing some bias in the experiments. Our choice is based on **Occam's razor** (Mitchell, 1997) that says to *"prefer the simplest hypothesis that fits the data"*. We expect that selecting the smallest reduct as filter, *i.e.* smallest number of relevant features for the Rough Set approach, will allow the inducers to find more simple rules.

After selecting the smallest reduct, the subset of features of the reduct (similarly to the subset of features found by (Lee et al., 1999) using ID3, $\mathcal{C}4.5$ and CI) were used to compute the accuracy for $\mathcal{C}4.5$, $\mathcal{C}4.5$-rules and $\mathcal{CN}2$ inducers.

6 Experimental Results

The next sections present the results obtained through these experiments. Note that all the experimental results related to the wrapper approach as well as the ones related to the use of ID3, $\mathcal{C}4.5$ and CI as filters, were extracted from (Lee et al., 1999).

6.1 Summary Tables Description

For each dataset four tables are presented:

1. The first table describes each feature in the dataset: feature number (features numbering starts at zero), feature name and type (continuous or nominal). For nominal features, the maximum possible number of values (as described in the *names* file) and the actual number of values (the one really found in the dataset through the $\mathcal{MLC}$++ *info* utility) are shown. It should be observed that a number of actual nominal values greater than the possible number of values indicates that there are missing values for that specific attribute. The reverse is not true.

2. The second table describes wrapper and filter selected features. To specify the experiment, it is used the notation FSS(*method,inducer*) where:

   - *method* $\in$ {wf, wb, f} indicating if wrapper forward (wf), backward (wb) or filter (f) selection of features has been used;

[8] More on reducts in Section 2.

- *inducer* $\in \{\mathcal{C}4.5, \mathcal{C}4.5\text{-rules}, \mathcal{CN}2, \text{ID3}, \text{CI}, \text{RS}\}$ indicating the algorithm or tool that has been used as wrapper or filter.

This table shows, for each FSS(*method,inducer*), the features subset selected, the number of features in the selected subset (#F), proportion of selected features (%F) as well as the time taken by the wrapper or filter method to obtain the selected features. Time (in seconds) is related to a standard *Indigo 2* Silicon Graphics workstation, except for Rosetta that was run in a Standard Pentium III 500MHz PC. It should be observed that *Indigo 2* is a little bit faster.

3. The third table shows similar information than the second one, but in a different way such that it is easy to visualize common features found by every FSS(*method,inducer*) tested.

4. The fourth table shows the error of each inducer (mean and standard deviation) using 10-fold cross-validation[9] (10-cv) using all features as well as the features subset selected by each FSS(*method,inducer*) considered. Each column represents the inducer used for accuracy estimation and each row represents the feature subset used. For instance, the first column indicates errors using $\mathcal{C}4.5$ as inducer; the first row of this column indicates error of $\mathcal{C}4.5$ using all features in the dataset, the second row indicates error using the feature subset selected by FSS(wf,$\mathcal{C}4.5$) and so on.

Note that in the second table of each dataset, any entry indicated as MC means that the majority class error is smaller than the error obtained by the subset of features being selected by the wrapper, *i.e.* the halting criterion is reached and the smaller error is given by the empty set of features. In the corresponding fourth table, errors marked with:

- † indicates that these errors are related with the majority class, *i.e.* the same entries marked with MC in the second table
- $\bullet$ indicates that these errors are grater than the majority class error, considering only the mean error
- $\triangle$ indicates that these errors are significantly higher at 95% confidence level.

6.2 TA

Feature Number	Feature Name	#Distinct Values possible	actual	type
#0	Eng-speaker	-	2	Nominal
#1	Course-inst	-	25	Nominal
#2	Course	-	26	Nominal
#3	Sem	-	2	Nominal
#4	Class-size	-	46	Continuous

continued on next page

[9] A 10-fold cross-validation (cv) is performed by dividing the data into 10 mutually exclusive subsets (folds) of cases of approximately equal size. The inducer is trained and tested 10 times, each time tested on a fold and trained on the dataset minus the fold. The cv estimate of accuracy is the average of the estimated accuracies from the 10 folds.

continued from previous page				
Feature Number	Feature Name	#Distinct Values		
		possible	actual	type

Table 6.2.1: TA – Feature Description

Inducer	Selected Features	#F	%F	Time (s)
FSS(wf,C4.5)	0 1 2 3	4	80.00%	11.60
FSS(wb,C4.5)	0 1 2 3	4	80.00%	8.90
FSS(wf,CN2)	0 1 2 4	4	80.00%	66.7
FSS(wb,CN2)	0 1 2 4	4	80.00%	63.1
FSS(wf,C4.5-rules)	MC	0	0.00%	13.20
FSS(wb,C4.5-rules)	MC	0	0.00%	30.00
FSS(f,CI)	0 1 2 3	4	80.00%	0.10
FSS(f,C4.5)	0 1 2 3 4	5	100.00%	0.00
FSS(f,ID3)	0 1 2 3 4	5	100.00%	0.70
FSS(f,RS)	1 2 4	3	60.00%	0.00

Table 6.2.2: TA – Time for Selecting Features

Feature Number	FSS									
	(wf,$\mathcal{C}$4.5)	(wb,$\mathcal{C}$4.5)	(wf,$\mathcal{CN}$2)	(wb,$\mathcal{CN}$2)	(wf,$\mathcal{C}$4.5-rules)	(wb,$\mathcal{C}$4.5-rules)	(f,CI)	(f,$\mathcal{C}$4.5)	(f,ID3)	(f,RS)
#0	◇	◇	◇	◇			◇	◇	◇	
#1	◇	◇	◇	◇			◇	◇	◇	◇
#2	◇	◇	◇	◇			◇	◇	◇	◇
#3	◇	◇					◇	◇	◇	
#4			◇	◇				◇	◇	◇
Total 5	4	4	4	4	0	0	4	5	5	3
100%	80.00%	80.00%	80.00%	80.00%	0.00%	0.00%	80.00%	100.00%	100.00%	60.00%

Table 6.2.3: TA – Wrapper and Filter Selected Features

ta 10-cv	$\mathcal{C}$4.5	$\mathcal{CN}$2	$\mathcal{C}$4.5-rules
all features	52.92±6.36	51.67±3.42	53.58±6.00
FSS(wf,$\mathcal{C}$4.5)	51.58±5.41		
FSS(wb,$\mathcal{C}$4.5)	51.58±5.41		
FSS(wf,$\mathcal{CN}$2)		48.34±3.11	
FSS(wb,$\mathcal{CN}$2)		48.34±3.11	
FSS(wf,$\mathcal{C}$4.5-rules)			65.56±3.88†
FSS(wb,$\mathcal{C}$4.5-rules)			65.56±3.88†
FSS(f,CI)	51.58±5.41	50.28±3.92	50.25±5.25
FSS(f,$\mathcal{C}$4.5)	52.92±6.36	51.67±3.42	53.58±6.00
FSS(f,ID3)	52.92±6.36	51.67±3.42	53.58±6.00
FSS(f,RS)	54.25±6.19	51.06±4.03	48.33±5.86

Table 6.2.4: TA – Errors

6.3 Bupa

Feature Number	Feature Name	#Distinct Values		
		possible	actual	type
#0	mcv	-	26	continuous
#1	alkphos	-	78	continuous
#2	sgpt	-	67	continuous
#3	sgot	-	47	continuous
#4	gammagt	-	94	continuous
#5	drinks	-	16	continuous
continued on next page				

continued from previous page

Feature Number	Feature Name	#Distinct Values possible	actual	type

Table 6.3.1: Bupa – Feature Description

Inducer	Selected Features	#F	%F	Time (s)
FSS(wf,C4.5)	0 1 2 4 5	5	83.33%	28.70
FSS(wb,C4.5)	0 1 2 4 5	5	83.33%	23.70
FSS(wf,CN2)	0 2 3 4 5	5	83.33%	189.70
FSS(wb,CN2)	0 2 3 4 5	5	83.33%	164.10
FSS(wf,C4.5-rules)	1 3	2	33.33%	28.30
FSS(wb,C4.5-rules)	1 3	2	33.33%	53.90
FSS(f,CI)	4	1	16.67%	0.10
FSS(f,C4.5)	0 1 2 3 4 5	6	100.00%	0.00
FSS(f,ID3)	0 1 2 3 4 5	6	100.00%	0.90
FSS(f,RS)	0 1 2	3	50.00%	0.00

Table 6.3.2: Bupa – Time for Selecting Features

Feature Number	FSS (wf,$\mathcal{C}$4.5)	(wb,$\mathcal{C}$4.5)	(wf,$\mathcal{CN}$2)	(wb,$\mathcal{CN}$2)	(wf,$\mathcal{C}$4.5-rules)	(wb,$\mathcal{C}$4.5-rules)	(f,CI)	(f,$\mathcal{C}$4.5)	(f,ID3)	(f,RS)
#0	◇	◇	◇	◇				◇	◇	◇
#1	◇	◇			◇	◇		◇	◇	◇
#2	◇	◇	◇	◇				◇	◇	◇
#3			◇	◇	◇	◇		◇	◇	
#4	◇	◇	◇	◇			◇	◇	◇	
#5	◇	◇	◇	◇				◇	◇	
Total 6	5	5	5	5	2	2	1	6	6	3
100%	83.33%	83.33%	83.33%	83.33%	33.33%	33.33%	16.67%	100.00%	100.00%	50.00%

Table 6.3.3: Bupa – Wrapper and Filter Selected Features

bupa 10-cv	$\mathcal{C}$4.5	$\mathcal{CN}$2	$\mathcal{C}$4.5-rules
all features	32.70±2.79	35.35±2.01	34.13±2.85
FSS(wf,$\mathcal{C}$4.5)	30.99±3.29		
FSS(wb,$\mathcal{C}$4.5)	30.99±3.29		
FSS(wf,$\mathcal{CN}$2)		32.17±2.96	
FSS(wb,$\mathcal{CN}$2)		32.17±2.96	
FSS(wf,$\mathcal{C}$4.5-rules)			46.66±2.07•△
FSS(wb,$\mathcal{C}$4.5-rules)			46.66±2.07•△
FSS(f,CI)	41.42±2.85△	45.21±1.98•△	41.42±2.85
FSS(f,$\mathcal{C}$4.5)	32.70±2.79	35.35±2.01	34.13±2.85
FSS(f,ID3)	32.70±2.79	35.35±2.01	34.13±2.85
FSS(f,RS)	43.19±2.18•△	38.53±2.94	42.62±2.49•△

Table 6.3.4: Bupa – Errors

6.4 Pima

Feature Number	Feature Name	#Distinct Values possible	actual	type
#0	Number	-	17	continuous
#1	Plasma	-	136	continuous
#2	Diastolic	-	47	continuous
#3	Triceps	-	51	continuous
#4	Two	-	186	continuous
#5	Body	-	248	continuous

continued on next page

continued from previous page

Feature Number	Feature Name	#Distinct Values		
		possible	actual	type
#6	Diabetes	-	517	continuous
#7	Age	-	52	continuous

Table 6.4.1: Pima – Feature Description

Inducer	Selected Features	#F	%F	Time (s)
FSS(wf,C4.5)	0 1 4 5 6	5	62.50%	81.90
FSS(wb,C4.5)	1 2 3 5 7	5	62.50%	89.20
FSS(wf,CN2)	0 1 2 4 5 6 7	7	87.50%	1292.10
FSS(wb,CN2)	0 1 2 4 5 6 7	7	87.50%	790.70
FSS(wf,C4.5-rules)	2 6 7	3	37.50%	172.50
FSS(wb,C4.5-rules)	2 6 7	3	37.50%	234.70
FSS(f,CI)	0 1 4 5 6 7	6	75.00%	0.40
FSS(f,C4.5)	0 1 2 4 5 6 7	7	87.50%	0.10
FSS(f,ID3)	0 1 2 3 4 5 6 7	8	100.00%	2.10
FSS(f,RS)	1 2 6	3	37.50%	1.00

Table 6.4.2: Pima – Time for Selecting Features

Feature Number	FSS									
	(wf,$\mathcal{C}$4.5)	(wb,$\mathcal{C}$4.5)	(wf,$\mathcal{CN}$2)	(wb,$\mathcal{CN}$2)	(wf,$\mathcal{C}$4.5-rules)	(wb,$\mathcal{C}$4.5-rules)	(f,CI)	(f,$\mathcal{C}$4.5)	(f,ID3)	(f,RS)
#0	◇		◇	◇			◇	◇	◇	
#1	◇	◇	◇	◇			◇	◇	◇	◇
#2		◇	◇	◇	◇	◇		◇	◇	◇
#3		◇							◇	
#4	◇		◇	◇			◇	◇	◇	
#5	◇	◇	◇	◇			◇	◇	◇	
#6	◇		◇	◇	◇	◇	◇	◇	◇	◇
#7		◇	◇	◇	◇	◇	◇	◇	◇	
Total 8	5	5	7	7	3	3	6	7	8	3
100%	62.50%	62.50%	87.50%	87.50%	37.50%	37.50%	75.00%	87.50%	100.00%	37.50%

Table 6.4.3: Pima – Wrapper and Filter Selected Features

pima 10-cv	$\mathcal{C}$4.5	$\mathcal{CN}$2	$\mathcal{C}$4.5-rules
all features	25.87±1.28	25.12±1.97	25.87±1.07
FSS(wf,$\mathcal{C}$4.5)	24.84±1.01		
FSS(wb,$\mathcal{C}$4.5)	23.01±1.07		
FSS(wf,$\mathcal{CN}$2)		23.69±1.22	
FSS(wb,$\mathcal{CN}$2)		23.69±1.22	
FSS(wf,$\mathcal{C}$4.5-rules)			37.83±1.66•△
FSS(wb,$\mathcal{C}$4.5-rules)			37.83±1.66•△
FSS(f,CI)	26.53±0.73	25.13±1.49	26.53±0.78
FSS(f,$\mathcal{C}$4.5)	25.88±0.99	23.69±1.22	26.39±1.13
FSS(f,ID3)	25.87±1.28	25.12±1.97	25.87±1.07
FSS(f,RS)	27.45±1.57	29.15±1.31△	27.71±1.49

Table 6.4.4: Pima – Errors

6.5 Breast Cancer2

Feature Number	Feature Name	#Distinct Values		
		possible	actual	type
#0	Age	-	44	continuous
#1	Age-at-meno	-	3	nominal
#2	Tumor-size	-	23	continuous

continued on next page

continued from previous page				
Feature Number	Feature Name	#Distinct Values		
		possible	actual	type
#3	Involved-nodes	-	18	continuous
#4	Node-capsule	3	3	nominal
#5	Degree-of-malig	-	3	continuous
#6	Breast	-	2	nominal
#7	Breast-Quadrant	6	6	nominal
#8	Irradiation	-	2	nominal

Table 6.5.1: Breast Cancer2 – Feature Description

Inducer	Selected Features	# F	%F	Time (s)
FSS(wf,C4.5)	1 3 5 6 8	5	55.56%	69.70
FSS(wb,C4.5)	1 3 5 6 8	5	55.56%	51.70
FSS(wf,CN2)	0 2 5 6	4	44.44%	312.50
FSS(wb,CN2)	0 1 4 5 6 7	7	77.78%	283.20
FSS(wf,C4.5-rules)	3 4 5 7	4	44.44%	49.80
FSS(wb,C4.5-rules)	0 1 2 3 4 6 7	6	66.67%	139.90
FSS(f,CI)	1 2 3 4 5 6 7 8	8	88.89%	0.20
FSS(f,C4.5)	0 1 3 4 5 6 7 8	8	88.89%	0.00
FSS(f,ID3)	0 1 2 3 4 5 6 7 8	9	100.00%	1.10
FSS(f,RS)	0 2 3 5 7	5	55.56%	1.00

Table 6.5.2: Breast Cancer2 – Time for Selecting Features

Feature Number	FSS									
	(wf,$\mathcal{C}$4.5)	(wb,$\mathcal{C}$4.5)	(wf,$\mathcal{CN}$2)	(wb,$\mathcal{CN}$2)	(wf,$\mathcal{C}$4.5-rules)	(wb,$\mathcal{C}$4.5-rules)	(f,CI)	(f,$\mathcal{C}$4.5)	(f,ID3)	(f,RS)
#0				◇	◇	◇		◇	◇	◇
#1	◇	◇		◇		◇	◇	◇	◇	
#2				◇	◇		◇		◇	◇
#3	◇	◇	◇	◇			◇	◇	◇	◇
#4			◇	◇		◇	◇	◇	◇	
#5	◇	◇	◇		◇	◇	◇	◇	◇	◇
#6	◇	◇		◇	◇	◇	◇	◇	◇	
#7			◇	◇		◇	◇	◇	◇	◇
#8	◇	◇					◇	◇	◇	
Total 9	5	5	4	7	4	6	8	8	9	5
100%	55.56%	55.56%	44.44%	77.78%	44.44%	66.67%	88.89%	88.89%	100.00%	55.56%

Table 6.5.3: Breast Cancer2 – Wrapper and Filter Selected Features

breast-cancer2 10-cv	$\mathcal{C}$4.5	$\mathcal{CN}$2	$\mathcal{C}$4.5-rules
all features	26.66±2.89	27.03±2.29	27.71±1.73
FSS(wf,$\mathcal{C}$4.5)	21.06±2.27		
FSS(wb,$\mathcal{C}$4.5)	21.06±2.27		
FSS(wf,$\mathcal{CN}$2)		21.41±1.82△	
FSS(wb,$\mathcal{CN}$2)		24.61±2.75△	
FSS(wf,$\mathcal{C}$4.5-rules)			35.44±2.61•△
FSS(wb,$\mathcal{C}$4.5-rules)			34.75±2.65•
FSS(f,CI)	25.63±2.59	27.71±1.68	29.46±2.48
FSS(f,$\mathcal{C}$4.5)	22.81±2.92	29.16±2.75	24.19±2.37
FSS(f,ID3)	26.66±2.89	27.03±2.29	27.71±1.73
FSS(f,RS)	24.95±1.89	27.75±2.79	25.70±2.37

Table 6.5.4: Breast Cancer2 – Errors

6.6 Cmc

Feature	Feature	#Distinct Values
continued on next page		

continued from previous page

Feature Number	Feature Name	#Distinct Values		
		possible	actual	type
Number	Name	possible	actual	type
#0	Wage	-	34	continuous
#1	Wedu	-	4	nominal
#2	Hedu	-	4	nominal
#3	Nchi	-	15	continuous
#4	Wrel	-	2	nominal
#5	Work	-	2	nominal
#6	Hocu	-	4	nominal
#7	Stdliv	-	4	nominal
#8	Medexp	-	2	nominal

Table 6.6.1: Cmc – Feature Description

Inducer	Selected Features	#F	%F	Time (s)
FSS(wf,C4.5)	0 1 3 8	4	44.44%	170.10
FSS(wb,C4.5)	0 1 3 8	4	44.44%	289.70
FSS(wf,CN2)	0 1 2 3 8	5	55.56%	4801.30
FSS(wb,CN2)	0 1 2 3 8	5	55.56%	4907.70
FSS(wf,C4.5-rules)	6 8	2	22.22%	270.20
FSS(wb,C4.5-rules)	6 8	2	22.22%	1985.30
FSS(f,CI)	0 1 2 3 4 5 6 7 8	9	100.00%	0.60
FSS(f,C4.5)	0 1 2 3 4 5 6 7 8	9	100.00%	0.20
FSS(f,ID3)	0 1 2 3 4 5 6 7 8	9	100.00%	5.50
FSS(f,RS)	0 1 2 3 4 5 6 7 8	9	100.00%	5.00

Table 6.6.2: Cmc – Time for Selecting Features

Feature Number	FSS									
	(wf,$\mathcal{C}$4.5)	(wb,$\mathcal{C}$4.5)	(wf,$\mathcal{CN}$2)	(wb,$\mathcal{CN}$2)	(wf,$\mathcal{C}$4.5-rules)	(wb,$\mathcal{C}$4.5-rules)	(f,CI)	(f,$\mathcal{C}$4.5)	(f,ID3)	(f,RS)
#0	◇	◇	◇	◇			◇	◇	◇	◇
#1	◇	◇	◇	◇			◇	◇	◇	◇
#2			◇	◇			◇	◇	◇	◇
#3	◇	◇	◇	◇			◇	◇	◇	◇
#4							◇	◇	◇	◇
#5							◇	◇	◇	◇
#6					◇	◇	◇	◇	◇	◇
#7							◇	◇	◇	◇
#8	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇
Total 9	4	4	5	5	2	2	9	9	9	9
100%	44.44%	44.44%	55.56%	55.56%	22.22%	22.22%	100%	100%	100%	100%

Table 6.6.3: Cmc – Wrapper and Filter Selected Features

cmc 10-cv	$\mathcal{C}$4.5	$\mathcal{CN}$2	$\mathcal{C}$4.5-rules
all features	47.94±1.49	49.64±1.01	45.90±1.38
FSS(wf,$\mathcal{C}$4.5)	43.93±0.78△		
FSS(wb,$\mathcal{C}$4.5)	43.93±0.78△		
FSS(wf,$\mathcal{CN}$2)		46.38±1.27△	
FSS(wb,$\mathcal{CN}$2)		46.38±1.27△	
FSS(wf,$\mathcal{C}$4.5-rules)			61.31±1.08•△
FSS(wb,$\mathcal{C}$4.5-rules)			61.31±1.08•△
FSS(f,CI)	47.94±1.49	49.64±1.01	45.90±1.38
FSS(f,$\mathcal{C}$4.5)	47.94±1.49	49.64±1.01	45.90±1.38
FSS(f,ID3)	47.94±1.49	49.64±1.01	45.90±1.38
FSS(f,RS)	47.94±1.49	49.22±1.05	45.90±1.38

Table 6.6.4: Cmc – Errors

6.7 Breast Cancer

Feature Number	Feature Name	#Distinct Values		
		possible	actual	type
#0	Clump Thickness	-	10	continuous
#1	Uniformity of Cell Size	-	10	continuous
#2	Uniformity of Cell Shape	-	10	continuous
#3	Marginal Adhesion	-	10	continuous
#4	Single Epithelial Cell Size	-	10	continuous
#5	Bare Nuclei	-	10	continuous
#6	Bland Chromatin	-	10	continuous
#7	Normal Nucleoli	-	10	continuous
#8	Mitoses	-	9	continuous

Table 6.7.1: Breast Cancer – Feature Description

Inducer	Selected Features	#F	%F	Time (s)
FSS(wf,C4.5)	0 1 3 4 5 6 8	7	77.78%	116.40
FSS(wb,C4.5)	0 1 3 4 5 6 8	7	77.78%	85.90
FSS(wf,CN2)	0 1 5 7 8	5	55.56%	606.60
FSS(wb,CN2)	0 1 5 7 8	5	55.56%	723.30
FSS(wf,C4.5-rules)	MC	0	0.00%	55.00
FSS(wb,C4.5-rules)	MC	0	0.00%	227.00
FSS(f,CI)	0 1 2 3 4 5 6 7 8	9	100.00%	0.40
FSS(f,C4.5)	0 1 2 3 4 5 6 8	8	88.89%	1.20
FSS(f,ID3)	0 1 2 3 4 5 6 7	8	88.89%	1.60
FSS(f,RS)	0 3 5 6	4	44.44%	1.00

Table 6.7.2: Breast Cancer – Time for Selecting Features

Feature Number	FSS									
	(wf,$\mathcal{C}$4.5)	(wb,$\mathcal{C}$4.5)	(wf,$\mathcal{CN}$2)	(wb,$\mathcal{CN}$2)	(wf,$\mathcal{C}$4.5-rules)	(wb,$\mathcal{C}$4.5-rules)	(f,CI)	(f,$\mathcal{C}$4.5)	(f,ID3)	(f,RS)
#0	◇	◇	◇	◇			◇	◇	◇	◇
#1	◇	◇	◇	◇			◇	◇	◇	
#2							◇	◇	◇	
#3	◇	◇					◇	◇	◇	◇
#4	◇	◇					◇	◇	◇	
#5	◇	◇	◇	◇			◇	◇	◇	◇
#6	◇	◇					◇	◇	◇	◇
#7			◇	◇			◇		◇	
#8	◇	◇	◇	◇			◇	◇		
Total 11	7	7	5	9	0	0	9	8	8	4
100%	77.78%	77.78%	55.56%	100.00%	0.00%	0.00%	100.00%	88.89%	88.89%	44.44%

Table 6.7.3: Breast Cancer – Wrapper and Filter Selected Features

breast-cancer 10-cv	C4.5	CN2	C4.5-rules
all features	5.86±0.84	4.87±0.77	4.29±0.60
FSS(wf,C4.5)	4.00±0.55		
FSS(wb,C4.5)	4.00±0.55		
FSS(wf,CN2)		3.57±0.67△	
FSS(wb,CN2)		3.57±0.67△	
FSS(wf,C4.5-rules)			34.48±1.80†
FSS(wb,C4.5-rules)			34.48±1.80†
FSS(f,CI)	5.86±0.84	4.87±0.77	4.29±0.60
FSS(f,C4.5)	6.01±0.76	4.44±0.61	4.29±0.60
FSS(f,ID3)	5.72±0.74	5.16±0.86	4.86±0.80
FSS(f,RS)	4.86±0.71	6.72±0.79△	4.29±0.67

Table 6.7.4: Breast Cancer – Errors

6.8 Smoke

Feature Number	Feature Name	#Distinct Values		
		possible	actual	type
#0	Weight	-	128	continuous
#1	Time	-	2	nominal
#2	Work1	-	2	nominal
#3	Work2	-	2	nominal
#4	Residence	-	2	nominal
#5	Smoking1	-	2	nominal
#6	Smoking2	-	2	nominal
#7	Smoking3	-	2	nominal
#8	Smoking4	-	2	nominal
#9	Knowledge	-	13	nominal
#10	Sex	-	2	nominal
#11	Age	-	73	continuous
#12	Education	-	5	nominal

Table 6.8.1: Smoke – Feature Description

Inducer	Selected Features	#F	%F	Time (s)
FSS(wf,C4.5)	MC	0	0.00%	671.90
FSS(wb,C4.5)	0 1 4 5 8 11	6	46.15%	1016.00
FSS(wf,CN2)	MC	0	0.00%	1084.10
FSS(wb,CN2)	0 1 2 4 5 9 11	7	53.85%	35408.40
FSS(wf,C4.5-rules)	0 2 6 7 8 9 10 12	8	61.54%	17082.90
FSS(wb,C4.5-rules)	0 1 3 4 8 9 11 12	8	61.54%	2975.00
FSS(f,CI)	1 2 3 4 5 6 7 8 9 10 12	11	84.62%	1.80
FSS(f,C4.5)	0 1 2 3 4 5 6 7 8 9 10 11 12	13	100.00%	0.50
FSS(f,ID3)	0 1 2 3 4 5 6 7 8 9 10 11 12	13	100.00%	11.50
FSS(f,RS)	0 2 3 4 5 6 7 8 9 11 12	11	84.62%	24.00

Table 6.8.2: Smoke – Time for Selecting Features

Feature Number	FSS									
	(wf,$\mathcal{C}$4.5)	(wb,$\mathcal{C}$4.5)	(wf,$\mathcal{CN}$2)	(wb,$\mathcal{CN}$2)	(wf,$\mathcal{C}$4.5-rules)	(wb,$\mathcal{C}$4.5-rules)	(f,CI)	(f,$\mathcal{C}$4.5)	(f,ID3)	(f,RS)
#0		◇		◇	◇	◇		◇	◇	◇
#1		◇		◇		◇	◇	◇	◇	
#2				◇	◇		◇	◇	◇	◇
#3						◇	◇	◇	◇	◇
#4		◇		◇		◇	◇	◇	◇	◇
#5		◇		◇			◇	◇	◇	◇
#6					◇		◇	◇	◇	◇
#7					◇		◇	◇	◇	◇
#8		◇			◇	◇	◇	◇	◇	◇
#9				◇	◇	◇	◇	◇	◇	◇
#10							◇	◇	◇	
#11		◇		◇	◇	◇		◇	◇	◇
#12					◇	◇	◇	◇	◇	◇
Total 13	0	6	0	7	8	8	11	13	13	11
100%	0.00%	46.15%	0.00%	53.85%	61.54%	61.54%	84.62%	100.00%	100.00%	84.62%

Table 6.8.3: Smoke – Wrapper and Filter Selected Features

smoke 10-cv	$\mathcal{C}$4.5	$\mathcal{CN}$2	$\mathcal{C}$4.5-rules
all features	31.45±0.93●	32.18±0.64●△	32.54±0.68●△
FSS(wf,$\mathcal{C}$4.5)	30.47±0.86†		
FSS(wb,$\mathcal{C}$4.5)	30.40±0.92△		
FSS(wf,$\mathcal{CN}$2)		30.47±0.86†	
FSS(wb,$\mathcal{CN}$2)		31.51±0.81●	
FSS(wf,$\mathcal{C}$4.5-rules)			35.13±1.10●
FSS(wb,$\mathcal{C}$4.5-rules)			34.92±1.06●△
FSS(f,CI)	30.47±0.95△	35.02±0.71●△	33.21±0.82●

continued on next page

continued from previous page

	$\mathcal{C}4.5$	$\mathcal{CN}2$	$\mathcal{C}4.5$-rules
FSS(f,$\mathcal{C}4.5$)	31.45±0.93•	32.18±0.64•△	32.54±0.68•△
FSS(f,ID3)	31.45±0.93•	32.18±0.64•△	32.54±0.68•△
FSS(f,RS)	31.42±0.84•	32.01±0.82•△	33.10±1.01•△

Table 6.8.4: Smoke – Errors

6.9 Hungaria

Feature Number	Feature Name	#Distinct Values		
		possible	actual	type
#0	age	-	38	continuous
#1	sex	-	2	continuous
#2	cp	-	4	continuous
#3	trestbps	-	31	continuous
#4	chol	-	153	continuous
#5	fbs	-	2	continuous
#6	restecg	-	3	continuous
#7	thalach	-	71	continuous
#8	exang	-	2	continuous
#9	oldpeak	-	10	continuous
#10	slope	-	3	continuous
#11	ca	-	2	continuous
#12	thal	-	3	continuous

Table 6.9.1: Hungaria – Feature Description

Inducer	Selected Features	#F	%F	Time (s)
FSS(wf,C4.5)	0 9 10 11 12	5	38.46%	83.60
FSS(wb,C4.5)	0 4 5 6 9 10 11 12	8	61.54%	104.80
FSS(wf,CN2)	8 10 11 12	4	30.77%	314.20
FSS(wb,CN2)	1 2 3 7 10 11 12	7	53.85%	1242.90
FSS(wf,C4.5-rules)	0 3 6 11	4	30.77%	118.50
FSS(wb,C4.5-rules)	0 2 4 6 8 12	6	46.15%	392.60
FSS(f,CI)	1 2 4 5 6 7 8 9 11 12	10	76.92%	0.40
FSS(f,C4.5)	0 1 2 3 4 5 6 7 8 9 10	11	84.62%	0.00
FSS(f,ID3)	0 1 2 3 4 5 7 8 9 10 12	11	84.62%	0.90
FSS(f,RS-b)	4 7 9	3	23.07%	0.00

Table 6.9.2: Hungaria – Time for Selecting Features

Feature Number	FSS									
	(wf,$\mathcal{C}4.5$)	(wb,$\mathcal{C}4.5$)	(wf,$\mathcal{CN}2$)	(wb,$\mathcal{CN}2$)	(wf,$\mathcal{C}4.5$-rules)	(wb, $\mathcal{C}4.5$-rules)	(f,CI)	(f,$\mathcal{C}4.5$)	(f,ID3)	(f,RS)
#0	◇	◇			◇	◇		◇	◇	
#1				◇			◇	◇	◇	
#2				◇		◇	◇	◇	◇	
#3				◇	◇			◇	◇	
#4		◇				◇	◇	◇	◇	◇
#5		◇					◇	◇	◇	
#6		◇			◇	◇	◇	◇		
#7				◇			◇	◇	◇	◇
#8			◇			◇	◇	◇	◇	
#9	◇	◇					◇	◇	◇	◇
#10	◇	◇	◇	◇				◇	◇	
#11	◇	◇	◇	◇	◇		◇			
#12	◇	◇	◇	◇		◇	◇		◇	
Total 13	5	8	4	7	4	6	10	11	11	3
100%	38.46%	61.54%	30.77%	53.85%	30.77%	46.15%	76.92%	84.62%	84.62%	23.07%

Table 6.9.3: Hungaria – Wrapper and Filter Selected Features

hungaria 10-cv	$\mathcal{C}4.5$	$\mathcal{CN}2$	$\mathcal{C}4.5$-rules
all features	20.08±2.69	21.44±2.19	20.05±2.90
FSS(wf,$\mathcal{C}4.5$)	17.03±2.71		
FSS(wb,$\mathcal{C}4.5$)	17.03±2.71		
FSS(wf,$\mathcal{CN}2$)		16.01±2.00△	
FSS(wb,$\mathcal{CN}2$)		15.97±2.59	
FSS(wf,$\mathcal{C}4.5$-rules)			44.60±2.97•△
FSS(wb,$\mathcal{C}4.5$-rules)			24.47±2.81
FSS(f,CI)	19.74±2.50	21.79±2.22	20.41±2.18
FSS(f,$\mathcal{C}4.5$)	20.09±2.59	20.02±2.62	19.40±2.66
FSS(f,ID3)	20.75±2.68	21.09±2.23	18.03±2.21
FSS(f,RS)	21.41±3.45	26.17±3.11	20.75±3.61

Table 6.9.4: Hungaria – Errors

6.10 Hepatitis

Feature Number	Feature Name	#Distinct Values		
		possible	actual	type
#0	age	-	49	continuous
#1	female	2	2	nominal
#2	steroid	2	3	nominal
#3	antivirals	2	2	nominal
#4	fatigue	2	3	nominal
#5	malaise	2	3	nominal
#6	anorexia	2	3	nominal
#7	liver-big	2	3	nominal
#8	liver-firm	2	3	nominal
#9	spleen-palpable	2	3	nominal
#10	spiders	2	3	nominal
#11	ascites	2	3	nominal
#12	varices	2	3	nominal
#13	bilirubin	-	34	continuous
#14	alk-phosphate	-	83	continuous
#15	sgot	-	84	continuous
#16	albumin	-	29	continuous
#17	protime	-	44	continuous
#18	histology	2	2	nominal

Table 6.10.1: Hepatitis – Feature Description

Inducer	Selected Features	#F	%F	Time (s)
FSS(wf,C4.5)	11 12 13 16 18	5	26.32%	77.20
FSS(wb,C4.5)	0 1 2 5 8 10 17	7	36.84%	149.60
FSS(wf,CN2)	1 3 4 6 9 11 16	7	36.84%	700.40
FSS(wb,CN2)	0 1 2 3 4 6 7 8 10 11 12 14 15 16 17 18	16	84.21%	583.00
FSS(wf,C4.5-rules)	0 6 8 9 13	5	26.32%	138.30
FSS(wb,C4.5-rules)	0 1 2 5 6 9 10 12 13 14 15 16	12	63.16%	310.70
FSS(f,CI)	2 3 5 8 10 11 13 16 17 18	10	52.63%	0.70
FSS(f,C4.5)	0 1 3 4 5 7 8 10 11 15 16 17	12	63.16%	0.00
FSS(f,ID3)	0 3 7 10 11 13 14 16 17	9	47.37%	0.60
FSS(f,RS)	0 10 16	3	15.79%	1.00

Table 6.10.2: Hepatitis – Time for Selecting Features

Feature Number	FSS									
	(wf,$\mathcal{C}4.5$)	(wb,$\mathcal{C}4.5$)	(wf,$\mathcal{CN}2$)	(wb,$\mathcal{CN}2$)	(wf,$\mathcal{C}4.5$-rules)	(wb, $\mathcal{C}4.5$-rules)	(f,CI)	(f,$\mathcal{C}4.5$)	(f,ID3)	(f,RS)
#0		◇		◇	◇	◇		◇	◇	◇

continued on next page

continued from previous page

Feature Number	FSS (wf,$\mathcal{C}4.5$)	(wb,$\mathcal{C}4.5$)	(wf,$\mathcal{CN}2$)	(wb,$\mathcal{CN}2$)	(wf,$\mathcal{C}4.5$-rules)	(wb,$\mathcal{C}4.5$-rules)	(f,CI)	(f,$\mathcal{C}4.5$)	(f,ID3)	(f,RS)
#1		◇	◇	◇		◇		◇		
#2		◇		◇		◇	◇			
#3			◇	◇			◇	◇	◇	
#4			◇	◇				◇		
#5		◇				◇	◇	◇		
#6			◇	◇	◇	◇				
#7				◇				◇	◇	
#8		◇		◇	◇		◇	◇		
#9			◇		◇	◇				
#10		◇		◇		◇	◇	◇	◇	◇
#11	◇		◇	◇			◇	◇	◇	
#12	◇			◇		◇				
#13	◇				◇	◇	◇		◇	
#14				◇		◇			◇	
#15				◇		◇		◇		
#16	◇		◇	◇		◇	◇	◇	◇	◇
#17		◇		◇			◇	◇	◇	
#18	◇			◇			◇			
Total 19 100%	5 26.32%	7 36.84%	7 36.84%	16 84.21%	5 26.32%	12 63.16%	10 52.63%	11 57.89%	9 47.37%	3 15.79%

Table 6.10.3: Hepatitis – Wrapper and Filter Selected Features

hepatitis 10-cv	$\mathcal{C}4.5$	$\mathcal{CN}2$	$\mathcal{C}4.5$-rules
all features	21.92±3.20●	16.18±1.80	20.54±3.02
FSS(wf,$\mathcal{C}4.5$)	14.17±2.67△		
FSS(wb,$\mathcal{C}4.5$)	12.25±1.77△		
FSS(wf,$\mathcal{CN}2$)		8.41±2.18	
FSS(wb,$\mathcal{CN}2$)		12.99±2.57	
FSS(wf,$\mathcal{C}4.5$-rules)			29.21±4.74●△
FSS(wb,$\mathcal{C}4.5$-rules)			29.79±3.98●
FSS(f,CI)	20.75±3.54●	20.09±3.42	18.71±3.36
FSS(f,$\mathcal{C}4.5$)	17.42±1.64	14.86±2.53	18.75±2.03
FSS(f,ID3)	19.46±2.93	18.17±2.21	19.46±2.44
FSS(f,RS)	19.33±3.42	20.66±3.01●△	18.71±3.86

Table 6.10.4: Hepatitis – Errors

7 Results Comparison

The following two subsections show tables which present a summary of the number of selected features by each method as well as the time for selecting those features for each dataset considered in this work. The third subsection presents tables and graphs which are useful to compare the obtained results.

7.1 Number of Selected Features

Table 7.1.1 shows, for each dataset, the number of selected features using the wrapper and filter approaches. It is also shown in this table the percentage of the total number of features selected by each FSS approach considering all datasets. Similar information is given in Table 7.1.2 considering the proportion and average of selected features.

Note that, in these tables, a zero value indicates that no feature has been selected, in such case the error is given by the the majority class.

As stated in (Lee et al., 1999), for the wrapper approach it can be observed that the number

of features selected by forward selection is always smaller or equal to the number of features selected by backward selection, *i.e.*

$$\#\text{FSS(wf,inducer)} \leq \#\text{FSS(wb,inducer)}$$

Another result from that previous work is that for the filter approach, the number of features selected by CI is always smaller or equal than the number of features selected by $\mathcal{C}4.5$ and ID3, *i.e.*

$$\#\text{FSS(f,CI)} \leq \#\text{FSS(f,}\mathcal{C}4.5\text{)} \text{ and } \#\text{FSS(f,CI)} \leq \#\text{FSS(f,ID3)}$$

Considering the new results obtained using the Rough Sets approach as filter it can be observed that the number of features selected by RS is always smaller or equal than the number of features selected by $\mathcal{C}4.5$ and ID3, *i.e.*

$$\#\text{FSS(f,RS)} \leq \#\text{FSS(f,}\mathcal{C}4.5\text{)} \text{ and } \#\text{FSS(f,RS)} \leq \#\text{FSS(f,ID3)}$$

Furthermore, the number of features selected by RS is smaller or equal to the number of features selected by CI, except for bupa dataset. It can be observed that the overall percentage of selected features using RS is less than 50% while the overall percentage of the selected features using CI is not less than 70%.

As expected, since $\mathcal{C}4.5$ and ID3 induce decision tree, the number of features selected by both algorithms is more or less the same, not considering hepatitis dataset. Furthermore, the overall percentage of selected features is around 85%.

From these results and only considering the number of features selected by each one of the four filters, *i.e.* CI, $\mathcal{C}4.5$, ID3 and RS, it is possible to conclude that RS is the overall winner.

The second step of the experiments — Figure 5.1 page 13 — is described in Section 7.3.

Dataset	#F	FSS									
		(wf,$\mathcal{C}4.5$)	(wb,$\mathcal{C}4.5$)	(wf,$\mathcal{CN}2$)	(wb,$\mathcal{CN}2$)	(wf,$\mathcal{C}4.5$-rules)	(wb,$\mathcal{C}4.5$-rules)	(f,CI)	(f,$\mathcal{C}4.5$)	(f,ID3)	(f,RS)
ta	5	4	4	4	4	0	0	4	5	5	3
bupa	6	5	5	5	5	2	0	1	6	6	3
pima	8	5	5	7	7	3	3	6	7	8	3
breast cancer2	9	5	5	4	7	4	6	8	8	9	5
cmc	9	4	4	5	5	2	2	9	9	9	9
breast cancer	9	7	7	5	9	0	0	9	8	8	4
smoke	13	0	6	0	7	8	8	11	13	13	11
hungaria	13	5	8	4	7	4	6	10	11	11	3
hepatitis	19	5	7	7	16	5	12	10	12	9	3
Total	100%	43.96%	56.04%	45.05%	73.63%	30.77%	40.66%	74.73%	86.81%	85.71%	48.35%

Table 7.1.1: Number of Selected Features

Dataset	#F	FSS									
		(wf,$\mathcal{C}4.5$)	(wb,$\mathcal{C}4.5$)	(wf,$\mathcal{CN}2$)	(wb,$\mathcal{CN}2$)	(wf,$\mathcal{C}4.5$-rules)	(wb,$\mathcal{C}4.5$-rules)	(f,CI)	(f,$\mathcal{C}4.5$)	(f,ID3)	(f,RS)
ta	5	80.00%	80.00%	80.00%	80.00%	0.00%	0.00%	80.00%	100.00%	100.00%	60.00%
bupa	6	83.33%	83.33%	83.33%	83.33%	33.33%	0.00%	16.67%	100.00%	100.00%	50.00%
pima	8	62.50%	62.50%	87.50%	87.50%	37.50%	37.50%	75.00%	87.50%	100.00%	37.50%
breast cancer2	9	55.56%	55.56%	44.44%	77.78%	44.44%	66.67%	88.89%	88.89%	100.00%	55.56%
cmc	9	44.44%	44.44%	55.56%	55.56%	22.22%	22.22%	100.00%	100.00%	100.00%	100.00%
breast cancer	9	77.78%	77.78%	55.56%	100.00%	0.00%	0.00%	100.00%	88.89%	88.89%	44.44%
smoke	13	0.00%	46.15%	0.00%	53.85%	61.54%	61.54%	84.62%	100.00%	100.00%	84.62%

continued on next page

continued from previous page

Dataset	#F	FSS									
		(wf,$\mathcal{C}$4.5)	(wb,$\mathcal{C}$4.5)	(wf,$\mathcal{CN}$2)	(wb,$\mathcal{CN}$2)	(wf,$\mathcal{C}$4.5-rules)	(wb,$\mathcal{C}$4.5-rules)	(f,CI)	(f,$\mathcal{C}$4.5)	(f,ID3)	(f,RS)
hungaria	13	38.46%	61.54%	30.77%	53.85%	30.77%	46.15%	76.92%	84.62%	84.62%	23.07%
hepatitis	19	26.32%	36.84%	36.84%	84.21%	26.32%	63.16%	52.63%	63.16%	47.37%	15.79%
Average	10.11	52.04%	60.91%	52.67%	75.12%	28.46%	33.03%	74.97%	90.34%	91.21%	52.33%

Table 7.1.2: Proportion of Selected Features

7.2 Time for Selecting Features

All experiments were run in a standard *Indigo 2* Silicon Graphics workstation, except for Rosetta that was run in a Standard Pentium III 500MHz PC. It should be observed that *Indigo 2* is a little bit faster. Table 7.2.1 shows the time taken, in seconds, to run the methods for selecting features.

Dataset	#F	FSS									
		(wf,$\mathcal{C}$4.5)	(wb,$\mathcal{C}$4.5)	(wf,$\mathcal{CN}$2)	(wb,$\mathcal{CN}$2)	(wf,$\mathcal{C}$4.5-rules)	(wb,$\mathcal{C}$4.5-rules)	(f,CI)	(f,$\mathcal{C}$4.5)	(f,ID3)	(f,RS)
ta	5	11.6	8.9	66.7	63.1	13.2†	30.0†	0.1	$\leq$ 0.01	0.7	0.0
bupa	6	28.7	23.7	189.7	164.1	28.3	53.9	0.1	$\leq$ 0.01	0.9	0.0
pima	8	81.9	89.2	1292.1	790.7	172.5	234.7	0.4	0.1	2.1	1.0
breast cancer2	9	69.7	51.7	312.5	283.2	49.8	139.9	0.2	$\leq$ 0.01	1.1	1.0
cmc	9	170.1	289.7	4801.3	4907.7	270.2	1985.3	0.6	0.2	5.5	5.0
breast cancer	10	116.4	85.9	606.6	723.3	55.0†	227.0†	0.4	1.2	1.6	1.0
smoke	13	671.9†	1016.0	1084.1†	35408.4	17082.9	2975.0	1.8	2.0	11.5	24.00
hungaria	13	83.6	104.8	314.2	1242.9	118.5	392.6	0.4	$\leq$ 0.01	0.9	0.0
hepatitis	19	77.2	149.6	700.4	583.0	138.3	310.7	0.7	$\leq$ 0.01	0.6	1.0
Total Time		1311.1	1819.5	9367.6	44166.4	17928.7	6349.1	4.7	3.5	24.9	32.00

Table 7.2.1: Time (in seconds) for Selecting Features

As before, any entry marked with †means that the value is related with the majority class error, *i.e.* this error is smaller than the error obtained by the subset of features being selected by the wrapper, in other words the halting criterion is reached and the smaller error is given by the empty set of features. Note also that the experiments that were run in a time smaller than 0.01s are indicated as $\leq$0.01.

As expected, the wrapper approach is quite slow in most cases when compared with the filter approach. For example, the maximum time taken for the filter approach is 24.0s for FSS(f,RS) using dataset smoke, while for the wrapper approach the maximum time is 35408.4s for FSS(wf,$\mathcal{CN}$2) using dataset hungarian, more than 1400 times slower.

Table 7.2.2 shows the time taken by the three algorithms used in this work for running ten-fold cross-validation using all features in the dataset.

Dataset	$\mathcal{C}$4.5	$\mathcal{CN}$2	$\mathcal{C}$4.5-rules	RS
	10-cv			
ta	0.5	6.9	2.1	1.0
bupa	1.6	8.1	2.7	3.0
pima	4.2	26.0	7.3	63.0
breast cancer2	1.3	8.0	2.6	2.0
cmc	5.6	133.5	100.8	12.0
breast cancer	3.2	13.8	7.2	5.0
smoke	13.5	443.9	533.1	106.0
hungaria	2.0	12.2	3.6	4.0
hepatitis	1.1	5.0	2.2	1.0
Average	3.7	73.0	73.5	21.89

continued on next page

continued from previous page				
Dataset	$\mathcal{C}4.5$	$\mathcal{CN}2$	$\mathcal{C}4.5$-rules	RS

Table 7.2.2: Time Taken by $\mathcal{C}4.5$, $\mathcal{C}4.5$-rules, $\mathcal{CN}2$ and Rosetta for Running Ten-Fold Cross-Validation Using all Features

As can be observed, $\mathcal{CN}2$ and $\mathcal{C}4.5$-rules inducers take more time than the others.

7.3 Comparing No FSS, Filter FSS, Forward and Backward Wrapper FSS

To determine whether the difference between two algorithms — say A_1 and A_2 — is significant or not, several graphs are presented in this section, each one showing six bars.

Each bar corresponds to the mean error divided by the standard deviation where ten-fold cross-validation has been used. When the length of the bars are greater than two, the results are significant at 95% confidence level.

The comparisons are made such that A_2 represents the inducer using the wrapper or filter selected features and A_1 is the inducer itself using all features. When the bar is bellow zero it means that A_2 outperforms A_1 — meaning that using only the wrapper or filter selected features did improve the accuracy of the standard algorithm.

For each dataset, the combined mean $\mathrm{m}(A_2 - A_1)$ and standard deviation $\mathrm{sd}(A_2 - A_1)$ are calculated, respectively, according to Equations 5 and 6. The difference in standard deviations is given by Equation 7.

$$\mathrm{m}(A_2 - A_1) = \mathrm{m}(A_2) - \mathrm{m}(A_1) \tag{5}$$

$$\mathrm{sd}(A_2 - A_1) = \sqrt{\frac{\mathrm{sd}(A_2)^2 + \mathrm{sd}(A_1)^2}{2}} \tag{6}$$

$$\mathrm{ad}(A_2 - A_1) = \frac{\mathrm{m}(A_2 - A_1)}{\mathrm{sd}(A_2 - A_1)} \tag{7}$$

Table 7.3.1 shows the results obtained by Equation 7, for each inducer error using no feature selection (*inducer*), forward (FSS(wf,*inducer*)) and backward (FSS(wb,*inducer*)) wrapper selected features for the same inducer (black box wrapper inducer equals accuracy estimator inducer). It is also presented in this table the results for ID3, $\mathcal{C}4.5$, Column Importance and RS used as filters FSS (FSS(f,*inducer*)).

Dataset	FSS(wf,$\mathcal{C}4.5$) $-\mathcal{C}4.5$	FSS(wb,$\mathcal{C}4.5$) $-\mathcal{C}4.5$	FSS(f,CI) $-\mathcal{C}4.5$	FSS(f,$\mathcal{C}4.5$) $-\mathcal{C}4.5$	FSS(f,ID3) $-\mathcal{C}4.5$	FSS(f,RS) $-\mathcal{C}4.5$
ta	-0.52	-0.52	-0.52	0.00	0.00	0.21
bupa	0.76	0.76	4.05	0.00	0.00	4.19
pima	-0.47	-1.60	1.51	0.26	0.00	1.10
breast cancer2	-1.79	-1.79	-0.19	-1.18	0.00	-0.70
cmc	-6.55	-6.55	0.00	0.00	0.00	0.00
breast cancer	-0.56	-0.56	-0.46	0.00	-1.40	-1.29

continued on next page

continued from previous page

Dataset	FSS(wf,$\mathcal{C}$4.5) $-\mathcal{C}$4.5	FSS(wb,$\mathcal{C}$4.5) $-\mathcal{C}$4.5	FSS(f,CI) $-\mathcal{C}$4.5	FSS(f,$\mathcal{C}$4.5) $-\mathcal{C}$4.5	FSS(f,ID3) $-\mathcal{C}$4.5	FSS(f,RS) $-\mathcal{C}$4.5
smoke	-1.33	-2.14	-2.08	0.00	0.00	-0.03
hungaria	-1.45	-1.45	-0.69	0.08	0.09	0.43
hepatitis	-2.39	-4.56	-0.51	-1.79	-0.19	-0.78

Dataset	FSS(wf,$\mathcal{CN}$2) $-\mathcal{CN}$2	FSS(wb,$\mathcal{CN}$2) $-\mathcal{CN}$2	FSS(f,CI) $-\mathcal{CN}$2	FSS(f,$\mathcal{C}$4.5) $-\mathcal{CN}$2	FSS(f,ID3) $-\mathcal{CN}$2	FSS(f,RS) $-\mathcal{CN}$2
ta	-0.54	-0.54	1.97	0.00	0.00	-0.16
bupa	1.02	1.02	5.39	0.00	0.00	1.26
pima	-0.09	-0.09	0.44	-0.11	0.00	2.41
breast cancer2	-5.08	-2.18	-0.44	-1.18	0.00	0.29
cmc	-2.88	-2.88	0.00	0.00	0.00	-0.40
breast cancer	-2.94	-2.94	0.00	-1.01	-0.38	2.37
smoke	-2.12	0.00	6.33	0.00	0.00	-0.23
hungaria	-2.02	-0.77	-0.45	0.19	-0.70	1.76
hepatitis	-3.13	-1.60	1.59	-1.11	-0.33	1.81

Dataset	FSS(wf,$\mathcal{C}$4.5-rules) $-\mathcal{C}$4.5-rules	FSS(wb,$\mathcal{C}$4.5-rules) $-\mathcal{C}$4.5-rules	FSS(f,CI) $-\mathcal{C}$4.5-rules	FSS(f,$\mathcal{C}$4.5) $-\mathcal{C}$4.5-rules	FSS(f,ID3) $-\mathcal{C}$4.5-rules	FSS(f,RS) $-\mathcal{C}$4.5-rules
ta	-4.52	-4.52	-0.70	0.00	0.00	-1.04
bupa	6.25	6.25	3.35	0.00	0.00	3.17
pima	8.47	8.47	1.62	0.00	0.00	1.42
breast cancer2	2.75	1.10	0.65	0.83	0.00	-0.97
cmc	13.32	13.32	0.00	0.00	0.00	0.00
breast cancer	42.55	42.55	0.00	0.00	0.00	0.64
smoke	1.76	2.00	-0.47	0.00	0.00	0.65
hungaria	6.61	0.57	-0.17	-0.17	0.11	0.21
hepatitis	1.34	1.54	-0.01	-1.51	0.07	-0.53

Table 7.3.1: Difference in Standard Deviations of Errors

Figures 7.3.1, 7.3.2 and 7.3.3 show the corresponding graphs from Table 7.3.1.

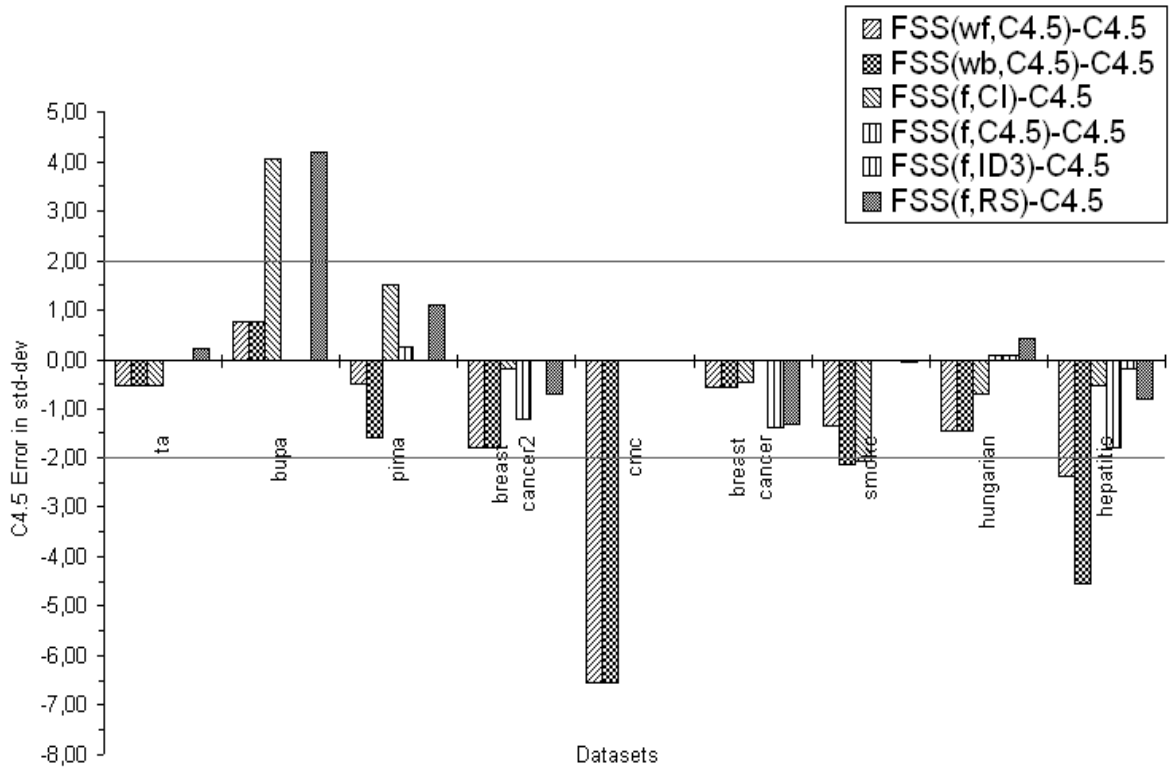

Figure 7.3.1: $\mathcal{C}$4.5 Difference in Standard Deviations of Errors

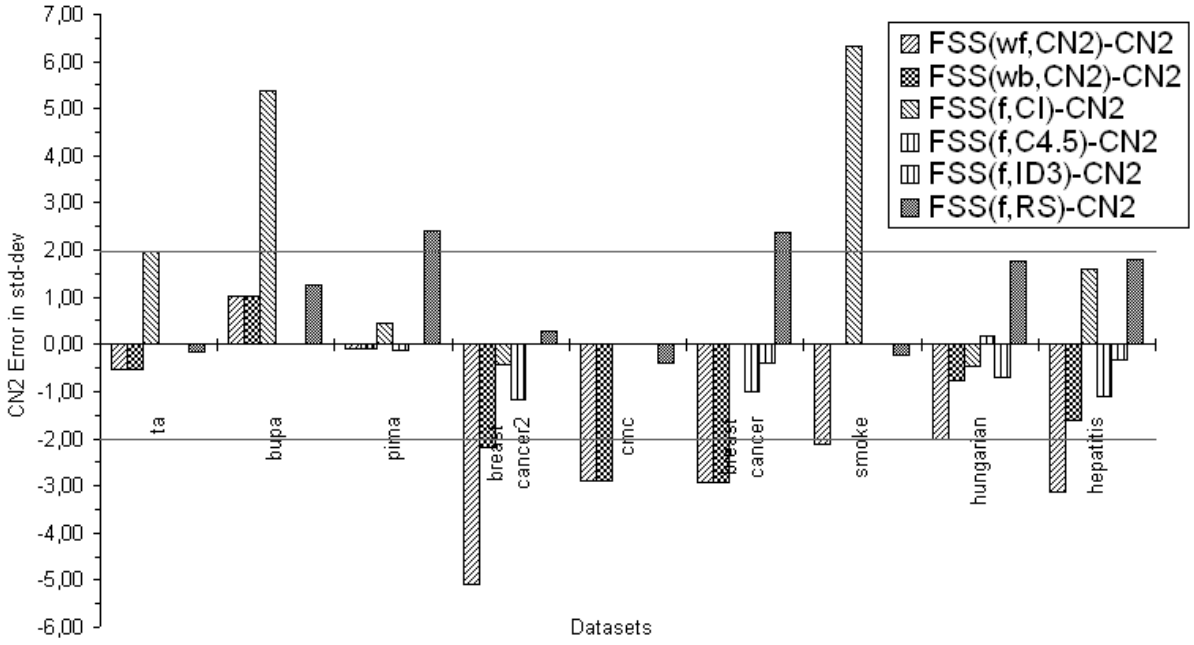

Figure 7.3.2: $\mathcal{CN}2$ Difference in Standard Deviations of Errors

For each dataset, the first bar in the graph corresponds to the comparison of wrapper forward feature selection against no feature selection. The second one corresponds to the comparison of wrapper backward feature selection against no feature selection.

The last four bars correspond to the algorithms used as filters against no feature selection where now is included RS approach as filter — main object of our study.

Considering graphs from Figures 7.3.1 and 7.3.2, it can be observed that the wrapper approach outperforms the standard inducer — $\mathcal{C}4.5$, $\mathcal{CN}2$ and $\mathcal{C}4.5$-rules respectively — in most cases, although not necessarily at the 95% confidence level.

Considering only the cases where the wrapper or filter approach outperforms the standard inducer at the 95% confidence level, or the other way round where the standard inducer outperforms the wrapper or filter approach at the 95% level, we have for $\mathcal{C}4.5$ — see Figure 7.3.1:

- For the cmc, smoke and hepatitis datasets, five cases where the wrapper approach showed to be better than the standard inducer
- For the bupa dataset, two cases where the standard inducer outperformed CI and RS used as filter, respectively
- For the smoke dataset, the CI used as filter outperformed the standard inducer once

Similarly for $\mathcal{CN}2$, we have —see Figure 7.3.2:

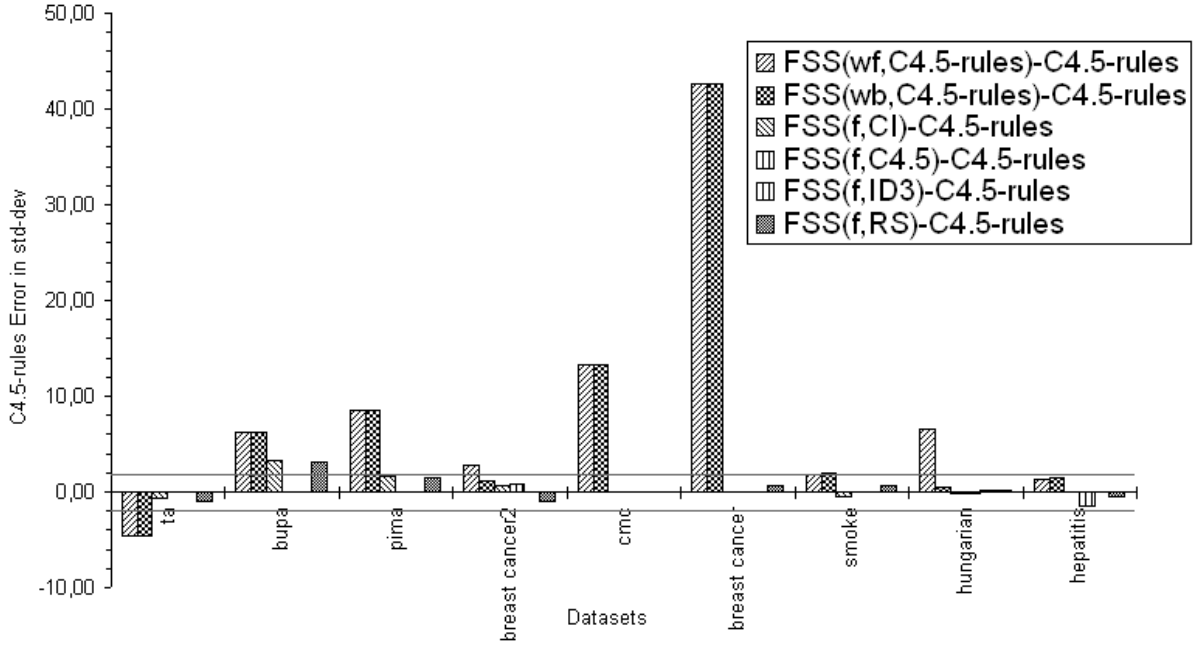

Figure 7.3.3: $\mathcal{C}$4.5-rules Difference in Standard Deviations of Errors

- For datasets bupa and smoke, two cases where the standard inducer outperformed the filter approach
- For datasets pima and breast cancer, two cases where the standard inducer outperformed RS used as filter
- For breast cancer2, cmc, breast-cancer, smoke, hungaria and hepatitis, nine cases where the wrapper approach outperformed the standard inducer

However, for $\mathcal{C}$4.5-rules — see Figure 7.3.3, it can be noted that the standard inducer outperforms the wrapper and filter approach in 12 cases and only for the dataset ta the wrapper showed to be better, at the 95% confidence level, than the standard inducer.

Table 7.3.2 shows improved accuracies at the significance level (95% confidence) for wrapper forward and backward selection, as well as filter selection, compared with the standard inducers: $\mathcal{C}$4.5, $\mathcal{C}$4.5-rules and $\mathcal{CN}$2.

Dataset	FSS												# △	# ▽
	(wf, $\mathcal{C}$4.5)	(wf, CN2)	(wf, $\mathcal{C}$4.5-rules)	(wb, $\mathcal{C}$4.5)	(wb, CN2)	(wb, $\mathcal{C}$4.5-rules)	(f,CI) $\mathcal{C}$4.5	(f,CI) CN2	(f,CI) $\mathcal{C}$4.5-rules	(f,RS) $\mathcal{C}$4.5	(f,RS) CN2	(f,RS) $\mathcal{C}$4.5-rules		
ta			△			△							2	0
bupa			▽			▽	▽	▽	▽	▽		▽	0	7
pima			▽			▽					▽		0	3
breast cancer2		△	▽		△								2	1
cmc	△	△	▽	△	△	▽							4	2
breast cancer		△	▽		△	▽					▽		2	3
smoke		△		△		▽	△	▽					3	2
hungaria		△	▽										1	1
hepatiti	△	△		△									3	0
# △	2	6	1	3	3	1	1	0	0				17	
# ▽	0	0	6	0	0	5	1	2	1					19

Table 7.3.2: Improved Accuracies at the Significance Level

Observe that for the filter approach, Table 7.3.2 only shows CI and RS filter selection compared with the standard inducers, since no improved accuracy at the 95% confidence level was obtained by using $\mathcal{C}$4.5 and ID3 as filters.

Improvements bellow 2 standard deviations are reported with △, *i.e.* the wrapper or filter approach outperforms the standard inducer at the 95% confidence level, and those bellow, where the standard inducer outperforms the wrapper or filter approach, with ▽.

Through Table 7.3.2, it can be seen that the wrapper approach outperforms the standard inducer in 16 of the 54 presented comparisons while the standard inducer outperforms the wrapper approach in 11 of the 54 comparisons.

Considering only this general result, it seems that the wrapper approach is not as good as expected. However, it should be observed that the standard inducer outperforms 11 times the wrapper approach but only for the $\mathcal{C}$4.5-rules inducer, confirming once more the good performance of $\mathcal{C}$4.5-rules on its own.

Another result is that FSS using the filter approach outperformed the standard inducer in only one case at the 95% confidence level. While the standard inducer outperformed eight times the filter approach at the 95% confidence level. Specifically, when using RS as filter, there is not a single case that improved the accuracy at the 95% confidence level and in four cases the standard inducer outperformed RS filter approach at the 95% confidence level.

Considering the total number of comparison using this approach — 81 comparisons — the results at 95% confidence level are not good.

Although there is only one case where the filter approach outperforms at the 95% confidence level the error rate of the standard inducer — dataset smoke using FSS(f,CI) and $\mathcal{C}$4.5 as inducer as shown in Table 7.3.2 — we decided to investigate these results further.

One of the reasons is that the filter approach is a very feast method, in contrast with the wrapper approach. Furthermore, in some cases, for example high cost in measuring features, it may be worth to consider the possibility of allowing a slight increase in classification error if some costly features can be discarded.

7.4 Other Results for Filter FSS

In this section we shall only focus on filter methods.

Some figures are presented showing, for each dataset and inducer used as filter, the difference of error in standard deviation as well as a coefficient that represents the proportion of discarded features after filter FSS. This coefficient is calculated as shown in Equation 8.

$$Dec(f, D) = 1 - \frac{|Features_f|}{|Features_D|} \tag{8}$$

where $|Features_D|$ is the total number of features present in dataset D and $|Features_f|$ is the number of features selected using the filter method f. Thus, $Dec(f, D)$ gives the percentage of discarded features after FSS.

In the following figures $Dec(f, D)$ is represented on the left vertical axis and the correspondent filter on the right. Thus, the filter which appears at the top right hand corner is the one that discarded more features. Note that it is possible to have a draw.

The difference of error in standard deviation refers for the inducer using only the features selected by the correspondent filter against the inducer using all features. Then, bars to the left indicate advantage of the filter method and to the right disadvantage.

Taking only into account the percentage of discarded features after FSS, it can be observed that Rough Sets is similar or outperforms the other filters, except for dataset bupa, where CI discarded more features — Figure 7.4.5.

However, the classification error should be taken into account for choosing a convenient pair *(Filter,Inducer)* such that the increase in classification error is reasonable considering the decrease in the number of features.

For dataset *Ta* — Figure 7.4.4 — FSS(f,RS) is appropriated for the three inducers.

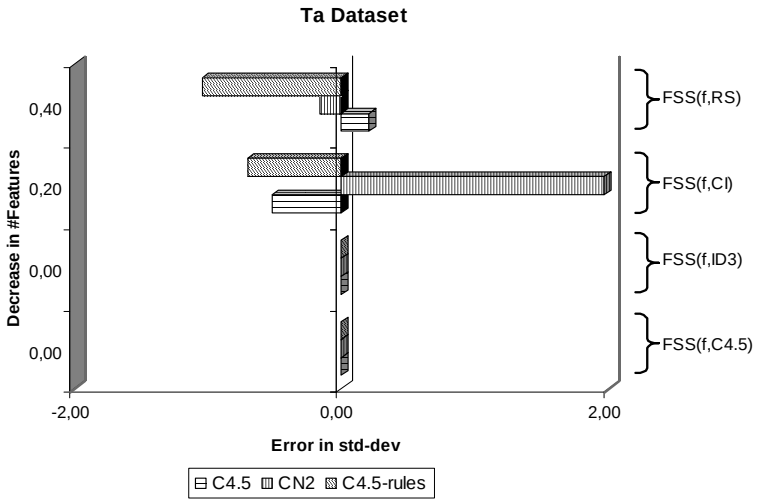

Figure 7.4.4: Difference in Std-Dev of Errors and Decrease in #F for dataset Ta

For dataset *Bupa* — Figure 7.4.5 — FSS(f,RS) is the best option but only for $\mathcal{CN}2$. In fact this dataset shows the worst results for the filter approach.

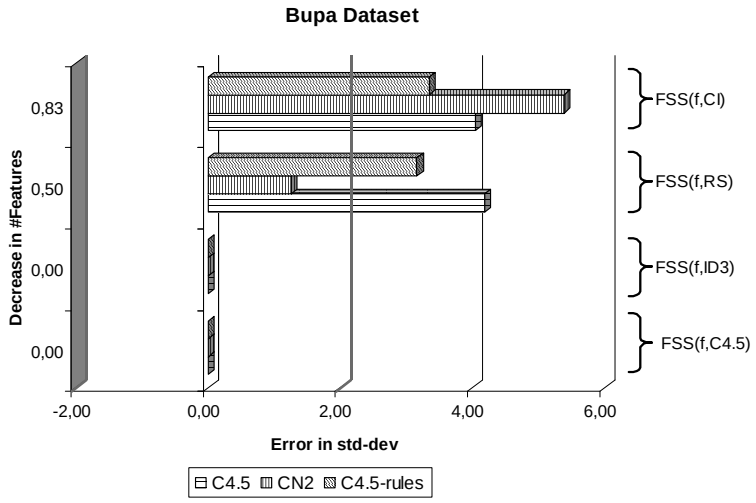

Figure 7.4.5: Difference in Std-Dev of Errors and Decrease in #F for dataset Bupa

For dataset *Pima* — Figure 7.4.6 — FSS(f,RS) is appropriated but only for $\mathcal{C}4.5$ and $\mathcal{C}4.5$-rules, and FSS(f,CI) for $\mathcal{CN}2$. However, if the classification error is the main concern, then FSS(f,$\mathcal{C}4.5$) should be selected for the three inducers.

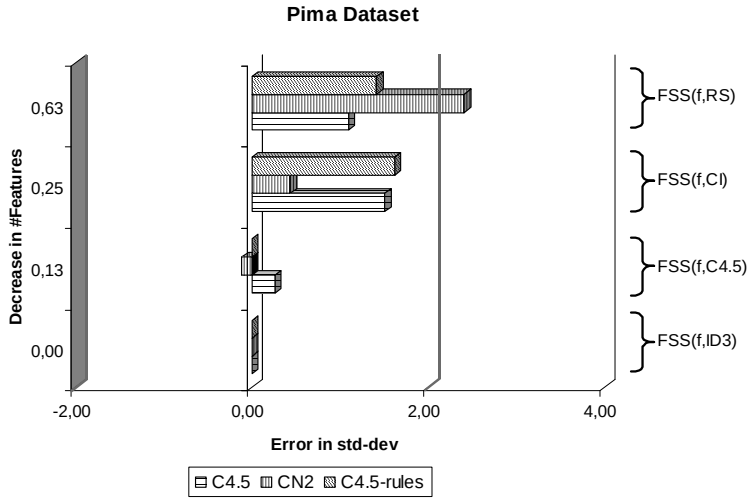

Figure 7.4.6: Difference in Std-Dev of Errors and Decrease in #F for dataset Pima

For dataset *Breast Cancer2* — Figure 7.4.7 — FSS(f,RS) is more appropriated for $\mathcal{C}4.5$ and $\mathcal{C}4.5$-rules, while FSS(f,CI) should be used with $\mathcal{CN}2$.

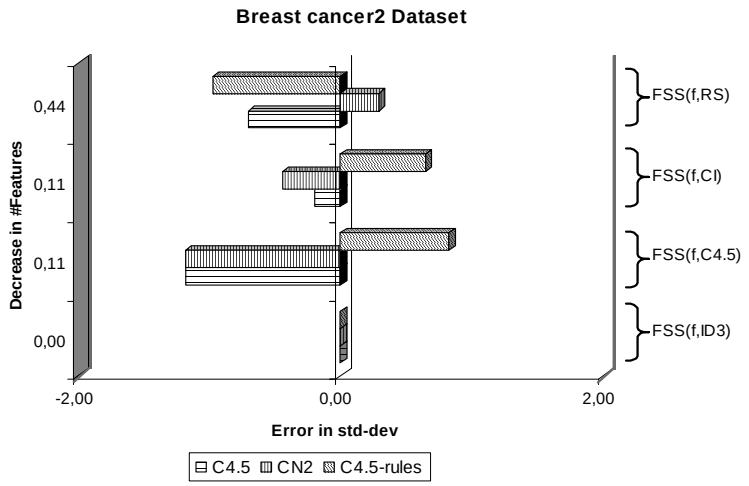

Figure 7.4.7: Difference in Std-Dev of Errors and Decrease in #F for dataset Breast Cancer2

For dataset *Cmc* — Figure 7.4.8 — all features seem to be relevant since none of the filters was able to discard any feature.

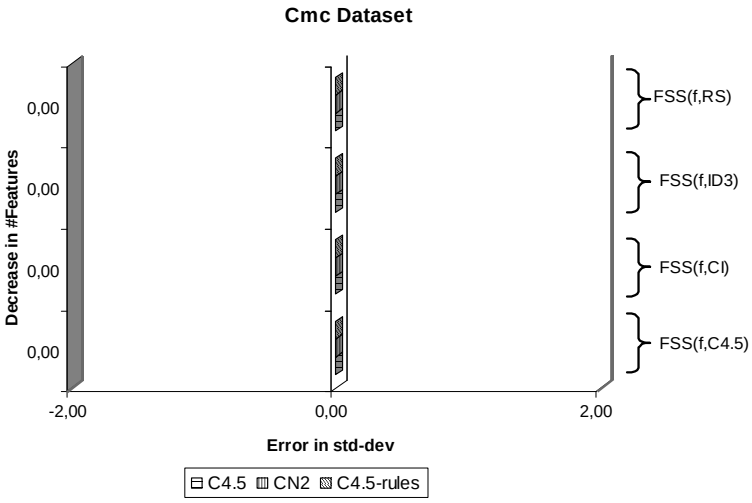

Figure 7.4.8: Difference in Std-Dev of Errors and Decrease in #F for dataset Cmc

For dataset *Breast Cancer* — Figure 7.4.9 — FSS(f,RS) is appropriated for $\mathcal{C}4.5$ and $\mathcal{C}4.5$-rules but not for $\mathcal{CN}2$, since the standard inducer outperforms it at the 95% confidence level. For $\mathcal{CN}2$, FSS(f,$\mathcal{C}4.5$) is more appropriated.

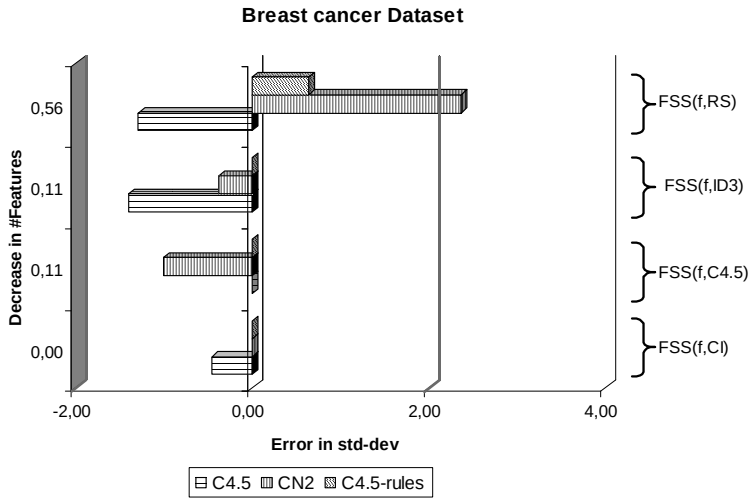

Figure 7.4.9: Difference in Std-Dev of Errors and Decrease in #F for dataset Breast Cancer

For dataset *Smoke* — Figure 7.4.10 — FSS(f,RS) is appropriated for $\mathcal{CN}2$ and FSS(f,CI) for $\mathcal{C}4.5$ and $\mathcal{C}4.5$-rules.

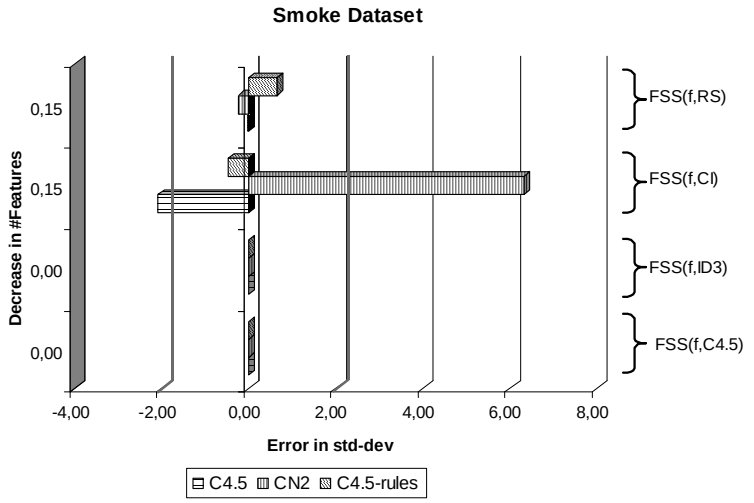

Figure 7.4.10: Difference in Std-Dev of Errors and Decrease in #F for dataset Smoke

For dataset *Hungarian* — Figure 7.4.11 — FSS(f,RS) is appropriated for $\mathcal{C}4.5$ and $\mathcal{C}4.5$-rules and FSS(f,CI) is appropriated for $\mathcal{CN}2$. Again, if the classification error is the main concern, then FSS(f,CI) should be selected for the three inducers.

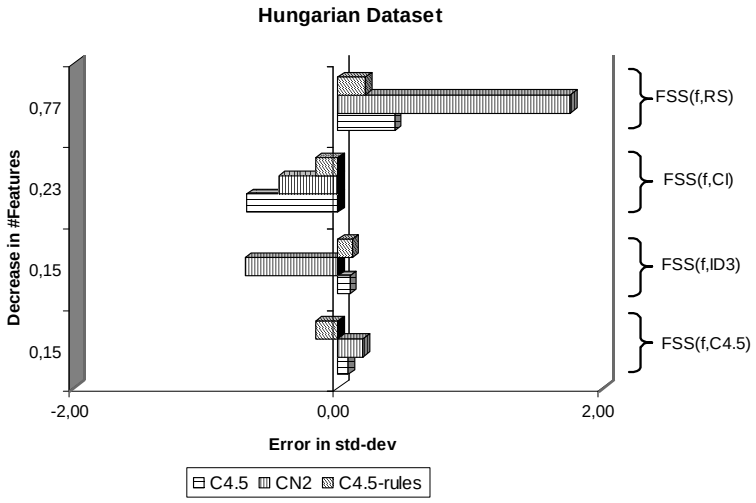

Figure 7.4.11: Difference in Std-Dev of Errors and Decrease in #F for dataset Hungarian

For dataset *Hepatitis* — Figure 7.4.12 — FSS(f,RS) is appropriated for $\mathcal{C}4.5$ and $\mathcal{C}4.5$-rules and FSS(f,ID3) for $\mathcal{CN}2$. However, if classification error is the main concern, the FSS(f,$\mathcal{C}4.5$) is a good option for the three inducers.

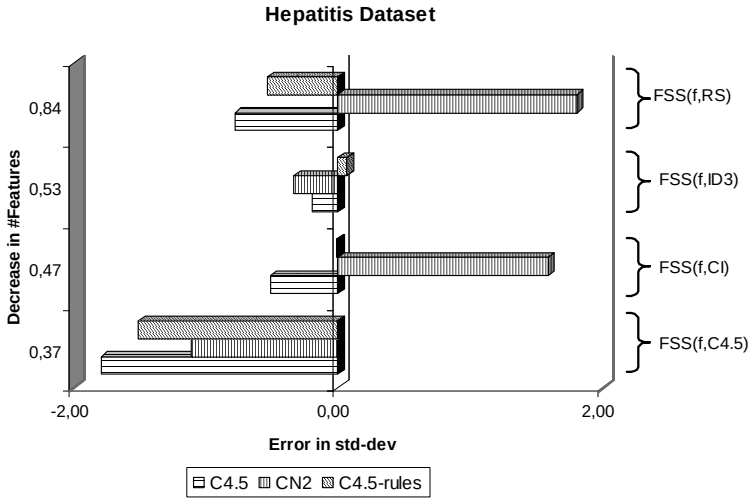

Figure 7.4.12: Difference in Std-Dev of Errors and Decrease in #F for dataset Hepatitis

8 Conclusions

At a conceptual level, the problem of Feature Subset Selection is that of finding a subset of the original features of a dataset, such that given this subset to an induction algorithm, it generates a classifier with the lowest possible error. It is important to notice that FSS chooses a set of features from the existing features and does not construct new ones, *i.e.* the description space is not increased.

In practice, it is desirable that the FSS process removes features which are not essential for learning since ML algorithms do not work well in the presence of many features. Furthermore, FSS can improve comprehensibility and can reduce the cost of processing huge quantities of data.

In this work we propose Rough Sets reducts as a filter method for FSS comparing its performance with previous empirical results extracted from (Lee et al., 1999) using the wrapper and other filter approaches for FSS on nine real world datasets.

Lee reports several results, which are also included in this work, using $\mathcal{C}4.5$, $\mathcal{C}4.5$-rules and $\mathcal{CN}2$ for wrapper approach as well as results using $\mathcal{C}4.5$, ID3 and CI MineSet™ facility as filters. In this work we show new results using RS reducts as filter.

The reducts, *i.e.* the filtered features using Rough Sets, were found using the software Rosetta (Øhrn, 1999b). Afterwards, similarly to Lee's work, the inducers $\mathcal{C}4.5$, $\mathcal{CN}2$ and $\mathcal{C}4.5$-rules were run using the $\mathcal{MLC}$++ library with its default option setting. The scripts used to run the experiments are listed in Appendix A.

Some extracted results from (Lee et al., 1999) and also reported in this work, show that for 7 of the 9 datasets $\mathcal{C}4.5$-rules has a very bad performance when used as a wrapper. Bad performance is also obtained using $\mathcal{C}4.5$ and $\mathcal{CN}2$. This bad performance is related to errors higher than the majority class error, *i.e.* given a new example it is better to classify it with the majority class than using returned classifier. A possible explanation for this is that features used to describe the dataset, *i.e.* the description language, are not adequate. One of the possible ways to try to improve the description language is the use of Constructive Induction (Bloedorn and Michalski, 1998; Lee and Monard, 2000a; Lee and Monard, 2000b).

Other explanation is that all the features used to describe the datasets are relevant for the learning bias of those inducers.

Related to the wrapper approach it can be observed that the time taken to select features is much greater than the time taken by the filter approach. When the number of features increases, the running time for this sort of datasets would make the wrapper approach infeasible. This can be observed in the results reported by (Baranauskas et al., 1999a; Baranauskas et al., 1999b; Baranauskas and Monard, 1999) where some experiments were done on datasets having a larger number of features.

Related to the filter approach, results show that this is a very fast method although except for one case, it does not outperforms the standard inducer at the 95% confidence level. Furthermore, in a few cases the standard inducer outperforms the filter approach at the 95% level.

Still, not considering bupa dataset, in most cases the increase in classification error is reasonable

considering the decrease in the number of selected features.

Results using Rough Sets reducts as filter show that for almost all datasets used in this work, this method selects the smallest subset of features although not necessary with the smallest increase in classification error.

We consider that a general procedure to follow in the filter approach is to test several methods. Afterwards, based on the allowed classification error *versus* the decrease in the number of features, it is possible to choose the more appropriated method for the specific problem.

It should be observed that we have considered all the errors as having equal importance not paying attention to unbalanced number of examples (Batista et al., 1999; Batista et al., 2000). However, for many applications, distinctions among different types of errors turn out to be important. A natural alternative is to assign different misclassification costs to each type of error, *i.e.* a penalty for making a mistake (Weiss and Kulikowski, 1990).

In Symbolic Machine Learning it is also important to consider the number and the kind of rules induced. We are currently investigating the impact of filters on the induced rules. Work in this direction is important for Datamining as shown in (Lin and Cercone, 1997).

References

Baranauskas, J. A. and Monard, M. C. (1999). The MLC^{++} wrapper for feature subset selection using decision tree, production rule, instance based and statistical inducers: Some experimental results. Technical Report 87, ICMC-USP. ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/Rt_87.ps.zip.

Baranauskas, J. A., Monard, M. C., and Horst, P. S. (1999b). Evaluation of CN2 induced rules using feature selection. *Argentine Symposium on Artificial Intelligence (ASAI/JAIIO/SADIO)*, pages 141–154. http://www.fmrp.usp.br/∼augusto/ps/ASAI99.web.ps.zip.

Baranauskas, J. A., Monard, M. C., and Horst, P. S. (1999a). Evaluation of feature selection by wrapping around the CN2 inducer. *Encontro Nacional de Inteligência Artificial (ENIA/SBC)*, pages 315–326. http://www.fmrp.usp.br/∼augusto/ps/ENIA99.web.ps.zip.

Batista, G. E., Carvalho, A., and Monard, M. C. (1999). Aplicando seleção unilateral em conjuntos de exemplos desbalanceados: Resultados iniciais. In *Anais II Encontro Nacional de Inteligência Artificial - ENIA 99*, pages 327–340.

Batista, G. E. A. P. A., Carvalho, A. C. P. L., and Monard, M. C. (2000). Applying one-sided selection to unbalanced datasets. In *Proceedings of the Mexican Congress on Artificial Intelligence (MICAI-2000), Lecture Notes in Artificial Intelligence*, pages 315–325.

Blake, C., Keogh, E., and Merz, C. (1998). Uci irvine repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Bloedorn, E. and Michalski, R. S. (1998). Data-Driven Construtive Induction. *IEEE Intelligent Systems*, 13(2):30–37. March/April 1998.

Blum, A. L. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, pages 245–271.

Clark, P. and Boswell, R. (1991). Rule induction with cn2: Some recent improvements. In Kodratoff, Y., editor, *Proceedings of the 5th European Conference EWSL 91*, pages 151–163. Springer-Verlag.

Clark, P. and Niblett, T. (1987). Induction in noise domains. In Bratko, I. and Lavrač, N., editors, *Proceedings of the 2nd European Working Session on Learning*, pages 11–30, Wilmslow, UK. Sigma.

Clark, P. and Niblett, T. (1989). The cn2 induction algorithm. *Machine Learning*, 3(4):261–283.

Ølrn, A. (1999a). *Discernibility and Rough Sets in Medicine: Tools and Applications*. PhD thesis, Norwegian University on Science and Technology.

Ølrn, A. (1999b). Rosetta: Technical reference manual. Technical report, Knowledge System Group.

Felix, L. C. M., Rezende, S. O., Doi, C. Y., de Paula, M. F., and Romanato, M. J. (1998). MLC^{++} biblioteca de aprendizado de máquina em C^{++}. Technical Report 72, ICMC-USP. ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/Rt_72.ps.zip.

Hu, X. (1995). *Knowledge Discovery in Databases: An Attibute-Oriented Rough Set Approach*. PhD thesis, University of Regina.

Hu, X. and Cercone, N. (1994). Discovery of decision rules in relational databases: A rough set approach. *CIKM'94*, page 9.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, pages 273–324.

Kohavi, R., Sommerfield, D., and Dougherty, J. (1994). *$\mathcal{MLC}$++: A Machine Learning Library in C++*. IEEE Computer Society Press.

Kohavi, R., Sommerfield, D., and Dougherty, J. (1996). Data mining using $\mathcal{MLC}$++: A amchine learning library in $\mathcal{C}$++. *Tools with IA*, pages 234–245.

Komorowski, J. and Øhrn, A. (1999). Modelling prognostic power of cardiac tests using rough sets. *Artificial Intelligence in Medicine*, pages 167–191.

Komorowski, J., Pawlak, Z., Polkowski, L., and Skowron, A. (1999). Rough sets: A tutorial. Technical report, Warsaw University.

Lee, H. D. and Monard, M. C. (2000a). Indução construtiva guiada pelo conhecimento: Um estudo de caso do processamento de sêmen diagnóstico. In *Proceedings IBERAMIA-SBIA 2000 Open Discussion Track, ISBN 85-87837-03-6*, pages 157–166.

Lee, H. D. and Monard, M. C. (2000b). A practical approach for knowledge-driven constructive induction. In *Procedings Argentine Symposium on Artificial Intelligence, ASAI'00, 29th International Conference SADIO*, pages 71–86.

Lee, H. D., Monard, M. C., and Baranauskas, J. A. (1999). Empirical comparison of wrapper and filter approaches for feature subset selection. Technical Report 94, ICMC-USP. ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/Rt_94.ps.zip.

Lin, T. Y. and Cercone, N. (1997). *Rough Sets and Data Mining: Analysis of Imprecise Data*. Kluwer Academic Publishers.

Mitchell, T. M. (1997). *Machine Learning*. WCB/McGraw-Hill.

Pawlak, Z. (1982). Rough sets. *International Jornal of Computer and Information Sciences*, pages 341–356.

Pawlak, Z. (1995). Rough set approach to knowledge-based decision support. *14th European Conference on Operational Research*, page 12.

Pawlak, Z. (1996). Rough sets, rough relations and rough functions. *Fundamenta Informaticae, 27*, pages 103–108.

Pawlak, Z., Grzymala-Busse, J., Slowinski, R., and Ziarko, W. (1995). Rough sets. *Comunications of the ACM*, pages 89–95.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

Quinlan, J. R. (1993). *$\mathcal{C}$4.5: Programs for Machine Learning*. Morgan Kaufmann. San Francisco, CA.

Stein, R. (1993). The dempster-shafer theory of evidential reasoning. *AI Expert*, August:26–31.

Szladow, A. and Ziarko, W. (1993). Rough sets: Working with imperfect data. *AI Expert*, July:36–41.

Weiss, S. M. and Kulikowski, C. A. (1990). *Computer Sysytems that Learn*. Morgan Kaufmann Publishers, Inc.

Yao, Y. Y. (1998). A comparative study of fuzzy sets and rough sets. *Journal of Information Sciences*, pages 227–242.

A Scripts used to Run the Experiments

The scripts used to run the experiments described in this work are listed in this Appendix.

A.1 K-fold Cross-Validation

```
fss-perfest <loglevel> <number-of-folds>

#!/bin/csh
#
# Author: Adriano Donizete Pila (pila@icmc.sc.usp.br)
#         LABIC-ICMC-USP --- Modified from a previous script from
          Jos Augusto Baranauskas (jaugusto@icmc.sc.usp.br)
#
# Summary: This script runs the MLC++ accuracy estimation PerfEst in
# several datasets with several inducers. Accuracies are estimated
# using cross-validation (cv). For each dataset, a file named dataset.fss
# contains features to be used for accuracy estimation.
# Results are kept in files for later user evaluation.
#
# arguments:
#     a) MLC++ loglevel (optional)
#     b) Number of folds (optional)
#
# pre:
#     a) file "datasets" containing in each line one dataset name,
#        without extension (.names, .data and .test assumed)
#     b) file "inducers" containing in each line one
#        MLC++ inducer to be used as accuracy estimator.
#     c) file "$dataset.fss" where $dataset must be one of the
#        datasets present in the datasets file.
#
# pos:
#     a) files $dataset.10CV.$inducer.result, for each $dataset in the
#        "dataset" file and for each $inducer in the "inducers" file. Each
#        output file contains the MLC++ accuracy estimation for cv evaluation
#        for each feature set present in the
#        $dataset file
#
# NOTE: There is no value checking for datasets and inducers to be used.
#       The user must check them for valid values before running this script.

# Search path for MLC++ libraries
unalias rm
alias libinfo 'setenv LD_LIBRARY_PATH /lib:/usr/mlclib/mlc'
alias libAccEst 'setenv LD_LIBRARY_PATH /usr/lib:/lib:/usr/mlclib/mlc'
```

```
alias libproject libinfo

# Define default MLC++ loglevel as 1 if it was not user supplied
set loglevel = 1
if ($1 != "") then        # has been supplied by the user?
  set loglevel = $1       # yes, set it up
endif
setenv LOGLEVEL $loglevel

# Define no. of folds. 10 is the default if it was not user supplied
set folds = 10
if ($2 != "") then        # has been supplied by the user?
  set folds = $2          # yes, set it up
endif

# Change this if your dataset has too many classes
setenv MAX_LABEL_VALS 30

if (! (-e inducers.accest)) then
  echo "There is no inducers.accest file"
  exit 1
endif

foreach dataset (`cat datasets`)
  foreach inducer (`cat inducers`)
    echo "================================================================="
    echo "Working on $dataset with Inducer $inducer ..."
    echo "================================================================="
    set outfile = $dataset.10CV.$inducer.result
    set stime=`date`
    echo "Start time ..: $stime" > $outfile
    echo "Inducer .....: $inducer" >> $outfile
    echo "Dataset  ....: $dataset" >> $outfile
    echo "Working dir .: `pwd`" >> $outfile
    echo "Output file .: $outfile" >> $outfile
    setenv INDUCER $inducer
    setenv DATAFILE $dataset.all
    setenv NAMESFILE $dataset.names
    set et  = `time PerfEst >> & $outfile`
    echo "Start time ......: $stime " >> $outfile
    echo "Stop time .......: `date`" >> $outfile
    echo "Execution time ..: $et ">> $outfile
  end
end
```