Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

Empirical Comparison of Wrapper and Filter Approaches for Feature Subset Selection

Huei Diana Lee

Maria Carolina Monard/ILTC

José Augusto Baranauskas

Nº 94

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos

Out./1999

Empirical Comparison of Wrapper and Filter Approaches for Feature Subset Selection *

Huei Diana Lee
Maria Carolina Monard/ILTC
José Augusto Baranauskas

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: {huei, mcmonard, jaugusto}@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. Some methods have been proposed to approach the Feature Subset Selection problem which can be grouped as: embeded, filter and wrapper methods. This work presents in details several experimental results and comparisons using the wrapper and filter approaches. $\mathcal{C}4.5$, $\mathcal{C}4.5$-rules and $\mathcal{CN}2$ were used as black box for the wrapper approach and ID3, $\mathcal{C}4.5$ and MineSet™ Column Importance facility for the filter approach. All the experiments were run on real world datasets, most of them obtained from the UCI Irvine Repository.

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

1999

*Work partially supported by State University of the West of Paraná — UNIOESTE, National Research Council — FINEP and Faculty of Medicine at Ribeirão Preto — USP.

Contents

List of Figures

List of Tables

1 Introduction

With the technological evolution, the amount of information that can be collected 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.

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 a variety of reasons that justify doing FSS. The first reason that can be pointed out is that most of the ML algorithms, that are computationally feasible, do not work well in the presence of very large number of features. This means that FSS can improve the accuracy of the classifiers generated by these algorithms. Another reason to use FSS is that it can improve comprehensibility, *i.e.* the human ability of understanding the data and the rules generated by symbolic ML algorithms. A third reason for doing FSS is the high cost in some domains for collecting data. Finally, FSS can reduce the cost of processing huge quantities of data.

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

- Embedded, where the FSS process is embedded within the basic induction algorithm
- Filter, where the FSS is used to filter the features before the induction process occurs
- 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 this work, we focus on the filter and wrapper approaches. To run the experiments, we selected nine datasets, most of them form UCI Irvine Repository (Blake et al., 1998). We also selected the inducers: $\mathcal{C}4.5$, $\mathcal{C}4.5$-rules, $\mathcal{CN}2$ and ID3 implemented in $\mathcal{MLC}$++ and the Column Importance facility provided by MineSet™.

The organization as well as the description of the results obtained in this work closely follows the one used by (Baranauskas and Monard, 1999).

This work is organized as follows: Section 2 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 3 gives a short description of the datasets used in the experiments. Section 4 shows the experimental setup used to run the experiments and Section 5 describes the results obtained from these experiments. Section 6 reports analysis and comparison of results. Finally, Section 7 gives some conclusions. The Appendix contains the scripts used to run the experiments.

2 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. In this approach, the algorithms greedily compile the training data into an intentional concept description, such as a rule set or decision tree, discarding the data after this process (Aha, 1997). Only the learned concept is used to classify new cases.

Besides these inducers, it has also been used a tool named "Column Importance facility" — CI provided by MineSet™ from Silicon Graphics.

The next sections describe the data format used as input to the inducers, a short description of each of inducer, as well as the CI facility.

2.1 Data Format

In supervised Machine Learning, it is generally presented to an inducer a set of training instances. Each instance is described typically by a vector of feature values and a class label which value can be either discrete or continuous. This vector is denoted by $(\mathbf{X}, Y)$ and is known as the feature-value (either attribute-value or spreadsheet) format. Table 2.1.1 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 rotulated 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 2.1.1: Feature-Value or Spreadsheet Format

The datasets file formats that $\mathcal{MLC}$++ recognizes by default are the *data*, *test* and *names* files. The *data* and *test* files contain labeled instances, one per line, of the training and test set respectively. The *names* file defines the scheme that allows parsing these two previous files. It describes 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).

2.2 ID3

ID3 (Quinlan, 1986) is member of a more general Machine Learning inducers family named Top Down Induction of Decision Trees – TDIDT. ID3 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.

A node in a decision tree represents a test on a particular attribute. Building a decision tree proceeds as follows (Quinlan, 1986): using the training set, an attribute is chosen to split it according to attribute's value. For each subset, another attribute is chosen to split each one according to

some criterion. This continues as long as each subset contains mix of instances belonging to different classes. Once a uniform subset — *i.e.* all instances in that subset belongs to the same class — has been obtained, a leaf node is created and labeled with the same name of the respective class.

When a new instance should be classified, beginning from the root of the induced tree, ID3 test-and-branch each node with the respective feature until it reaches one leaf. The class prediction of this instance is assigned as the class of that leaf. If no rule is satisfied, the default rule assigns the most common (majority) class to the new example.

The original version of ID3 uses as test in the decision nodes the gain criterion which is calculated based on a quantity known as entropy. The criterion used in the $\mathcal{MLC}$++ ID3 with default settings is called Normalized-Mutual-Info which is very similar to the gain criterion. It is also based on entropy and is given by:

$$\frac{Entropy}{\log_2(NumberChildNodes)} \tag{1}$$

This ID3 version of $\mathcal{MLC}$++ with its default settings also handles unknown values, although the original version (Quinlan, 1986) of this algorithm did not.

The next inducer to be described has a better mechanism of handling unknown values.

2.3 $\mathcal{C}4.5$

$\mathcal{C}4.5$ (Quinlan, 1993) is one of the ID3 successors. Many extensions to the basic ID3 algorithm were added, such as improving computacional 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. The use of the gain ratio criterion can avoid a serious deficiency of the gain criterion: it has a strong bias in favor of tests with many outcomes.

2.4 $\mathcal{C}4.5$-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.

To classify a case using a production rule model, the ordered list of rules is examined to find the first whose left-hand side is satisfied by the case. The predicted class is then the one nominated by the right-hand side of this rule. If no rule's left-hand side is satisfied, the case is predicted as belonging to the default class.

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.

2.5 $\mathcal{CN}2$

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 midvalue of the most commonly occurring sub-range replaces the unknown value.

To classify a new instance using induced unordered rules (default $\mathcal{CN}2$ rule generation), all rules are tried and those which fire are collected. If more than one class is predicted by fired rules, the method used is to tag each rule with the distribution of covered examples among classes and then to sum these distributions to find the most probable class. For instance, consider the three rules:

if	head=square	**and**	hold=gun	**then**	class=enemy	covers	[15,1]
if	size=tall	**and**	flies=no	**then**	class=friend	covers	[1,10]
if	look=angry			**then**	class=enemy	covers	[20,0]

Here the two classes are [enemy,friend] and [15,1] denotes that the rule covers 15 training instances of enemy and 1 of friend. Given a new instance of a robot which has square head, carries a gun, tall, non-flying and angry, all three rules are fired. $\mathcal{CN}2$ resolve this clash by summing the covered instances [36,11] and then predicting the most common class in the sum — enemy.

2.6 CI

CI is a "column importance facility" provided by MineSet™ from Silicon Graphics[1]. 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 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 3.2 summarizes datasets characteristics. It follows a basic datasets description.

3.1 General Description

TA This dataset was first reported by (Loh and Shih, 1997). It consists of evaluation of teaching performance over 3 regular semesters and 2 summer semesters of 151 teaching assistant assignments

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

at the Statistics Department of the University of Wisconsin – Madison. The scores are grouped into 3 roughly equal-sized categories to form the class attribute: low, medium and high. There are 5 attributes and 151 instances.

Bupa This dataset was contributed by R. S. Forsyth to the UCI repository. The problem is to predict whether or not a male patient has liver disorders based on various blood tests and the amount of alcohol consumption.

Pima This dataset was donated by V. Sigillito, Applied Physics Laboratory, Johns Hopkins University to the UCI repository. This dataset is also a subset of a larger database maintained by the National Institute of Diabetes and Digestive and Kidney Diseases.

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 according to World Health Organization (WHO) criteria — *i.e.*, if the 2-hour post-load plasma glucose is at least 200 mg/dl at any survey examination or if found during routine medical care — given a number of physiological measurements and medical test results.

Breast-cancer2 This dataset is one of the breast cancer datasets at UCI, donated by Ljubljana Oncology Institute. There are 285 instances, 2 classes and 10 attributes, including the class attribute. The problem is to predict the recurrence or not of breast cancer.

CMC This dataset is composed by a subset of the 1987 National Indonesia Contraceptive Prevalence Survey and was donated by Tjen-Sien Lim. The samples 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 (no use, long-term methods or short-term methods) of a woman based on her demographic and socio-economic characteristics. There are 1473 instances, 3 classes and 9 attributes.

Breast-cancer This dataset was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg (Mangasarian and Wolberg, 1990). The problem is to predict whether a tissue sample taken from a patient's breast is malignant or benign. Tissue samples consist of visually assessed nuclear features of fine needle aspirates taken from patient's breast. Each sample was assigned a 9-dimensional vector. Each component is in the range 1 to 10, with 1 referring to a normal state and 10 to a most abnormal one. Malignancy is determined by taking a tissue sample from patient's breast and performing a biopsy on it. A benign diagnosis is confirmed by biopsy or by periodic examination, depending on the patient's choice.

Smoke This survey dataset (Bull, 1994) 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. It is composed by 3 classes, 13 attributes and 2855 instances.

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

Hungaria This dataset is for diagnosing heart diseases.

3.2 Datasets Summary

Table 3.2.1 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[2].

Datasets are presented in ascending order of the number of features, as will be in the remaining tables and graphs. Figure 3.2.1 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 3.2.1 is represented as $\log_{10}$(#Instances).

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

Table 3.2.1: Datasets Summary Descriptions

4 Experimental Setup

A series of experiments were performed, using the algorithms and datasets described respectively in Sections 2 and 3. It is important to observe that the original data has not been pre-processed in any way, for example by removing or replacing missing values or transforming nominal to numerical attributes.

Futhermore, 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.

As stated earlier, we used the wrapper and filter approach for FSS. For each approach, the performed experiments can be divided into two independent steps – Figure 4.0.1:

- 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 and CI are used as filters
- 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; 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

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

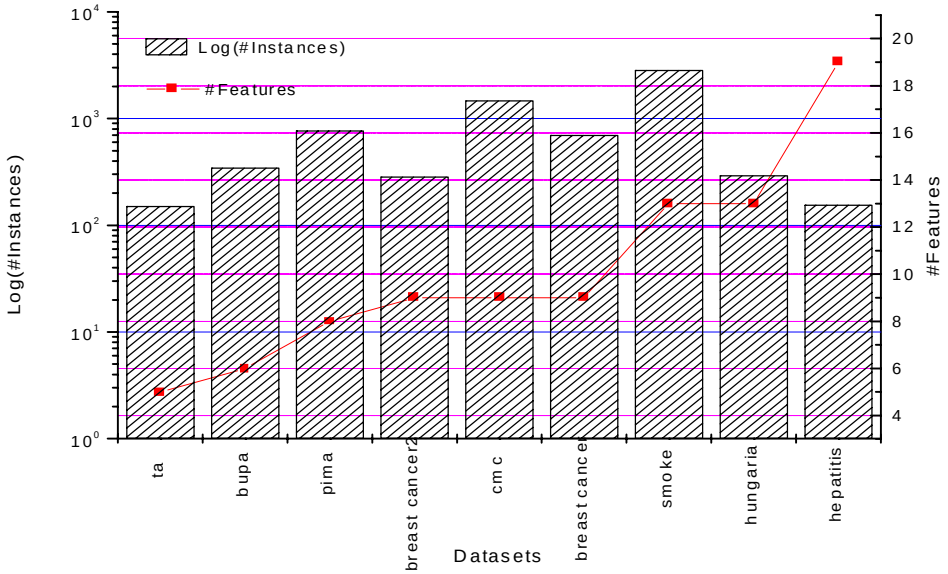

Figure 3.2.1: Datasets Dimensionality

It follows a more detailed description of the experiments.

The $\mathcal{MLC}$++ wrapper was run using $\mathcal{C}4.5$, $\mathcal{C}4.5$-rules and $\mathcal{CN}2$ over each dataset considering both forward and backward selection. In foward selection the initial state is the empty set of features and features are added step by step till the halting criterion is reached. The backward selection approach begins with the full set of features and on each step features are removed until the halting criterion is reached. This process produced as outcome a set of features which could be the original set of features, a subset of features or an empty set of features. The last case happens when the error predicted by the majority class is smaller than the one predicted using the subsets of features selected by the wrapper.

After this, the error of each inducer, using all features and the features selected by the wrapper, was measured using ten-fold cross-validation and ten-fold stratified cross-validation.

The filter process was conducted as follows: ID3, $\mathcal{C}4.5$ and CI were applied as filters for all the datasets described earlier. It should be observed that the obtained outcome is a subset of features or the original set of features, depending on the bias of the inducer and the dataset itself. Again, the result of this process was applied to $\mathcal{C}4.5$, $\mathcal{C}4.5$-rules and $\mathcal{CN}2$ and the errors computed.

The results obtained by wrapping around the $\mathcal{C}4.5$ and $\mathcal{CN}2$ inducers for bupa, hungaria, hepatiti and pima datasets were extracted from a previous work developed by (Baranauskas and Monard, 1999).

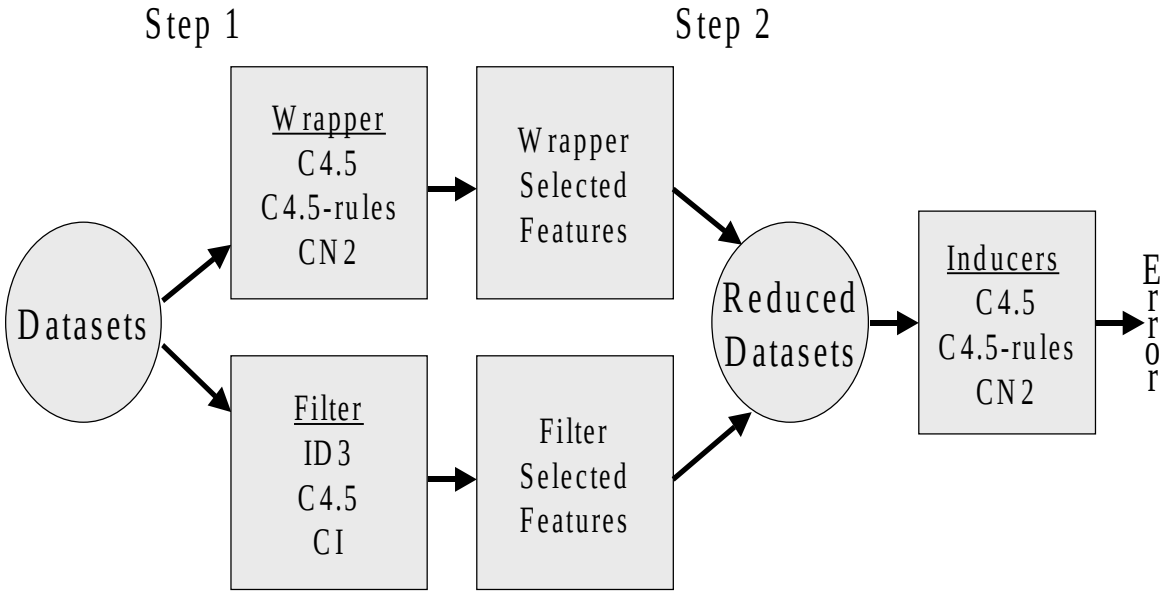

Figure 4.0.1: Experiments Steps

5 Experimental Results

The next sections present the results obtained through these experiments.

5.1 Summary Tables Description

Four tables are presented for each dataset:

- 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.
- 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;
  - *inducer* $\in$ {$\mathcal{C}$4.5, $\mathcal{C}$4.5-rules, $\mathcal{CN}$2, ID3, CI} indicating the algorithm 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.

- 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.

- The fourth table shows the error of each inducer (mean and standard deviation) using 10-fold cross-validation[3] (10-cv) and 10-fold stratified cross-validation[4] (10-strat-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. Also, in the corresponding fourth table, these errors related with the majority class are marked with †.

5.2 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

Table 5.2.1: 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

Table 5.2.2: 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)
#0	◇	◇	◇	◇			◇	◇	◇
#1	◇	◇	◇	◇			◇	◇	◇
#2	◇	◇	◇	◇			◇	◇	◇
#3	◇	◇					◇	◇	◇
#4			◇	◇				◇	◇

continued on next page

[3] 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.

[4] Similar to 10-fold cross-validation but the folds are stratified so that they contain approximatelly the same proportion of labels as the original dataset.

continued from previous page									
Total 5 100%	4 80.00%	4 80.00%	4 80.00%	4 80.00%	0 0.00%	0 0.00%	4 80.00%	5 100.00%	5 100.00%

Table 5.2.4: 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)			34.44±3.88†
FSS(wb,$\mathcal{C}4.5$-rules)			34.44±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
ta 10-strat-cv	$\mathcal{C}4.5$	$\mathcal{CN}2$	$\mathcal{C}4.5$-rules
all features	51.67±3.82	42.43±3.64	51.00±3.44
FSS(wf,$\mathcal{C}4.5$)	49.67±3.86		
FSS(wb,$\mathcal{C}4.5$)	49.67±3.86		
FSS(wf,$\mathcal{CN}2$)		40.42±3.80	
FSS(wb,$\mathcal{CN}2$)		40.42±3.80	
FSS(wf,$\mathcal{C}4.5$-rules)			34.44±3.88†
FSS(wb,$\mathcal{C}4.5$-rules)			34.44±3.88†
FSS(f,CI)	49.67±3.86	48.99±2.99	48.37±4.01
FSS(f,$\mathcal{C}4.5$)	51.67±3.82	42.43±3.64	51.00±3.44
FSS(f,ID3)	51.67±3.82	42.43±3.64	51.00±3.44

Table 5.2.5: TA – Errors

5.3 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

Table 5.3.1: 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

Table 5.3.2: 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)
#0	◇	◇	◇	◇				◇	◇
#1	◇	◇			◇	◇		◇	◇
#2	◇	◇	◇	◇				◇	◇
#3			◇	◇	◇	◇		◇	◇
#4	◇	◇	◇	◇			◇	◇	◇
#5	◇	◇	◇	◇				◇	◇
Total 6	5	5	5	5	2	2	1	6	6
100%	83.33%	83.33%	83.33%	83.33%	33.33%	33.33%	16.67%	100.00%	100.00%

Table 5.3.3: 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
bupa 10-strat-cv	$\mathcal{C}$4.5	$\mathcal{CN}$2	$\mathcal{C}$4.5-rules
all features	31.29±1.73	32.18±2.11	31.87±2.20
FSS(wf,$\mathcal{C}$4.5)	33.03±2.76		
FSS(wb,$\mathcal{C}$4.5)	33.03±2.76		
FSS(wf,$\mathcal{CN}$2)		34.19±1.83	
FSS(wb,$\mathcal{CN}$2)		34.19±1.83	
FSS(wf,$\mathcal{C}$4.5-rules)			44.57±1.85
FSS(wb,$\mathcal{C}$4.5-rules)			44.57±1.85
FSS(f,CI)	39.08±2.10	44.37±2.40	39.08±2.10
FSS(f,$\mathcal{C}$4.5)	31.29±1.73	32.18±2.11	31.87±2.20
FSS(f,ID3)	31.29±1.73	32.18±2.11	31.87±2.20

Table 5.3.4: Bupa – Errors

5.4 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
#6	Diabetes	-	517	continuous
#7	Age	-	52	continuous

Table 5.4.1: 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

continued on next page

continued from previous page				
Inducer	Selected Features	#F	%F	Time (s)
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

Table 5.4.2: 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)
#0	◇		◇	◇			◇	◇	◇
#1	◇	◇	◇	◇			◇	◇	◇
#2		◇	◇	◇	◇	◇		◇	◇
#3		◇							◇
#4	◇		◇	◇			◇	◇	◇
#5	◇	◇	◇	◇			◇	◇	◇
#6	◇		◇	◇	◇	◇	◇	◇	◇
#7		◇	◇	◇	◇	◇	◇	◇	◇
Total 8	5	5	7	7	3	3	6	7	8
100%	62.50%	62.50%	87.50%	87.50%	37.50%	37.50%	75.00%	87.50%	100.00%

Table 5.4.3: 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
pima 10-strat-cv	$\mathcal{C}$4.5	$\mathcal{CN}$2	$\mathcal{C}$4.5-rules
all features	25.74±1.13	25.38±1.38	26.00±1.03
FSS(wf,$\mathcal{C}$4.5)	25.23±1.04		
FSS(wb,$\mathcal{C}$4.5)	24.05±0.98		
FSS(wf,$\mathcal{CN}$2)		25.25±1.43	
FSS(wb,$\mathcal{CN}$2)		25.25±1.43	
FSS(wf,$\mathcal{C}$4.5-rules)			37.05±1.53
FSS(wb,$\mathcal{C}$4.5-rules)			37.05±1.53
FSS(f,CI)	27.18±0.73	25.91±1.02	27.70±1.07
FSS(f,$\mathcal{C}$4.5)	26.01±0.94	25.25±1.43	26.00±0.99
FSS(f,ID3)	25.74±1.13	25.38±1.38	26.00±1.03

Table 5.4.4: Pima – Errors

5.5 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
#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 5.5.1: 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

Table 5.5.2: 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)
#0				◇	◇	◇		◇	◇
#1	◇	◇		◇		◇	◇	◇	◇
#2				◇	◇		◇		◇
#3	◇	◇	◇	◇			◇	◇	◇
#4			◇	◇		◇	◇	◇	◇
#5	◇	◇	◇		◇	◇	◇	◇	◇
#6	◇	◇		◇	◇	◇	◇	◇	◇
#7			◇	◇		◇	◇	◇	◇
#8	◇	◇					◇	◇	◇
Total 9	5	5	4	7	4	6	8	8	9
100%	55.56%	55.56%	44.44%	77.78%	44.44%	66.67%	88.89%	88.89%	100.00%

Table 5.5.3: 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
breast-cancer2 10-strat-cv	$\mathcal{C}$4.5	$\mathcal{CN}$2	$\mathcal{C}$4.5-rules
all features	25.92±1.80	29.81±1.86	26.59±2.47
FSS(wf,$\mathcal{C}$4.5)	22.76±1.74		
FSS(wb,$\mathcal{C}$4.5)	22.76±1.74		
FSS(wf,$\mathcal{CN}$2)		21.41±1.42	
FSS(wb,$\mathcal{CN}$2)		25.63±1.97	
FSS(wf,$\mathcal{C}$4.5-rules)			32.59±1.85
FSS(wb,$\mathcal{C}$4.5-rules)			29.50±2.83
FSS(f,CI)	25.55±2.06	29.08±1.40	28.33±2.86
FSS(f,$\mathcal{C}$4.5)	23.83±1.74	27.69±1.84	28.74±2.68
FSS(f,ID3)	25.92±1.80	29.81±1.86	26.59±2.47

Table 5.5.4: Breast Cancer2 – Errors

5.6 Cmc

Feature Number	Feature Name	#Distinct Values		
		possible	actual	type
#0	Wage	-	34	continuous
#1	Wedu	-	4	nominal
#2	Hedu	-	4	nominal
#3	Nchi	-	15	continuous

continued on next page

continued from previous page

Feature Number	Feature Name	#Distinct Values possible	actual	type
#4	Wrel	-	2	nominal
#5	Work	-	2	nominal
#6	Hocu	-	4	nominal
#7	Stdliv	-	4	nominal
#8	Medexp	-	2	nominal

Table 5.6.1: 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

Table 5.6.2: 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)
#0	◇	◇	◇	◇			◇	◇	◇
#1	◇	◇	◇	◇			◇	◇	◇
#2			◇	◇			◇	◇	◇
#3	◇	◇	◇	◇			◇	◇	◇
#4							◇	◇	◇
#5							◇	◇	◇
#6					◇	◇	◇	◇	◇
#7							◇	◇	◇
#8	◇	◇	◇	◇	◇	◇	◇	◇	◇
Total 9	4	4	5	5	2	2	9	9	9
100%	44.44%	44.44%	55.56%	55.56%	22.22%	22.22%	100%	100%	100%

Table 5.6.3: 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
cmc 10-strat-cv	$\mathcal{C}$4.5	$\mathcal{CN}$2	$\mathcal{C}$4.5-rules
all features	49.02±0.89	50.22±1.07	46.44±1.03
FSS(wf,$\mathcal{C}$4.5)	43.66±0.74		
FSS(wb,$\mathcal{C}$4.5)	43.66±0.74		
FSS(wf,$\mathcal{CN}$2)		47.47±0.82	
FSS(wb,$\mathcal{CN}$2)		47.47±0.82	
FSS(wf,$\mathcal{C}$4.5-rules)			60.56±1.09

continued on next page

continued from previous page

	$\mathcal{C}4.5$	$\mathcal{CN}2$	$\mathcal{C}4.5$-rules
FSS(wb,$\mathcal{C}4.5$-rules)			60.56±1.09
FSS(f,CI)	49.02±0.89	50.22±1.07	46.44±1.03
FSS(f,$\mathcal{C}4.5$)	49.02±0.89	50.22±1.07	46.44±1.03
FSS(f,ID3)	49.02±0.89	50.22±1.07	46.44±1.03

Table 5.6.5: Cmc – Errors

5.7 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 5.7.1: 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	9	100.00%	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

Table 5.7.2: 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)
#0	◇	◇	◇	◇			◇	◇	◇
#1	◇	◇	◇	◇			◇	◇	◇
#2							◇	◇	◇
#3	◇	◇					◇	◇	◇
#4	◇	◇					◇	◇	◇
#5	◇	◇	◇	◇			◇	◇	◇
#6	◇	◇					◇	◇	◇
#7			◇	◇			◇		◇
#8	◇	◇	◇	◇			◇	◇	
Total 11	7	7	5	9	0	0	9	8	8
100%	77.78%	77.78%	55.56%	100.00%	0.00%	0.00%	100.00%	88.89%	88.89%

Table 5.7.3: 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	

continued on next page

continued from previous page

	$\mathcal{C}4.5$	$\mathcal{CN}2$	$\mathcal{C}4.5$-rules
FSS(wb,CN2)		3.57±0.67	
FSS(wf,C4.5-rules)			65.52±1.80†
FSS(wb,C4.5-rules)			65.52±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
breast-cancer 10-scv	C4.5	CN2	C4.5-rules
all features	5.43±0.70	5.72±1.08	4.86±0.91
FSS(wf,C4.5)	5.00±0.83		
FSS(wb,C4.5)	5.00±0.83		
FSS(wf,CN2)		3.15±0.60	
FSS(wb,CN2)		3.15±0.60	
FSS(wf,C4.5-rules)			65.52±1.80†
FSS(wb,C4.5-rules)			65.52±1.80†
FSS(f,CI)	5.43±0.70	5.72±1.08	4.86±0.91
FSS(f,C4.5)	5.72±0.56	4.85±0.86	4.86±0.83
FSS(f,ID3)	5.43±0.70	5.28±1.24	4.86±0.91

Table 5.7.4: Breast Cancer – Errors

5.8 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 5.8.1: 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)	0 1 2 3 4 5 6 7 8 9 10 11 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

Table 5.8.2: 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)
#0		◇		◇	◇	◇		◇	◇
#1		◇		◇		◇	◇	◇	◇
#2				◇	◇		◇	◇	◇
#3						◇	◇	◇	◇

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)
#4		◇		◇		◇	◇	◇	◇
#5		◇		◇			◇	◇	◇
#6					◇		◇	◇	◇
#7					◇		◇	◇	◇
#8		◇			◇	◇	◇	◇	◇
#9				◇	◇	◇	◇	◇	◇
#10							◇	◇	◇
#11		◇		◇	◇	◇		◇	◇
#12					◇	◇	◇	◇	◇
Total 13	0	6	0	7	8	8	11	13	13
100%	0.00%	46.15%	0.00%	53.85%	61.54%	61.54%	84.62%	100.00%	100.00%

Table 5.8.3: 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}$4.5-rules)			34.92±1.06
FSS(f,CI)	30.47±0.95	35.02±0.71	33.21±0.82
FSS(f,$\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
smoke 10-strat-cv	$\mathcal{C}$4.5	$\mathcal{CN}$2	$\mathcal{C}$4.5-rules
all features	31.52±0.71	31.87±0.35	32.71±0.65
FSS(wf,$\mathcal{C}$4.5)	30.47±0.86†		
FSS(wb,$\mathcal{C}$4.5)	30.44±0.06		
FSS(wf,$\mathcal{CN}$2)		30.47±0.86†	
FSS(wb,$\mathcal{CN}$2)		31.87±0.41	
FSS(wf,$\mathcal{C}$4.5-rules)			34.08±0.89
FSS(wb,$\mathcal{C}$4.5-rules)			34.22±0.85
FSS(f,CI)	30.47±0.06	35.90±0.83	32.43±0.54
FSS(f,$\mathcal{C}$4.5)	31.52±0.71	31.87±0.35	32.71±0.65
FSS(f,ID3)	31.52±0.71	31.87±0.35	32.71±0.65

Table 5.8.4: Smoke – Errors

5.9 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 5.9.1: 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

Table 5.9.2: 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)
#0	◇	◇			◇	◇		◇	◇
#1				◇			◇	◇	◇
#2				◇		◇	◇	◇	◇
#3				◇	◇			◇	◇
#4		◇				◇	◇	◇	◇
#5		◇					◇	◇	◇
#6		◇			◇	◇	◇	◇	
#7				◇			◇	◇	◇
#8			◇			◇	◇	◇	◇
#9	◇	◇					◇	◇	◇
#10	◇	◇	◇	◇				◇	◇
#11	◇	◇	◇	◇	◇		◇		
#12	◇	◇	◇	◇		◇	◇		◇
Total 13	5	8	4	7	4	6	10	11	11
100%	38.46%	61.54%	30.77%	53.85%	30.77%	46.15%	76.92%	84.62%	84.62%

Table 5.9.3: 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
hungaria 10-strat-cv	$\mathcal{C}$4.5	$\mathcal{CN}$2	$\mathcal{C}$4.5-rules
all features	22.48±4.20	22.07±3.06	22.09±3.63
FSS(wf,$\mathcal{C}$4.5)	17.03±3.27		
FSS(wb,$\mathcal{C}$4.5)	17.03±3.27		
FSS(wf,$\mathcal{CN}$2)		16.34±2.60	
FSS(wb,$\mathcal{CN}$2)		19.35±3.94	
FSS(wf,$\mathcal{C}$4.5-rules)			42.18±2.30
FSS(wb,$\mathcal{C}$4.5-rules)			24.14±3.50
FSS(f,CI)	19.76±3.61	20.75±2.80	21.46±3.74
FSS(f,$\mathcal{C}$4.5)	22.83±4.08	22.75±3.43	21.43±3.99
FSS(f,ID3)	22.84±3.56	20.02±2.79	22.46±3.13

Table 5.9.4: Hungarian – Errors

5.10 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 5.10.1: 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

Table 5.10.2: Hepatitis – Time for Selecting Features

Feature Number	FSS (wf,C4.5)	(wb,C4.5)	(wf,CN2)	(wb,CN2)	(wf,C4.5-rules)	(wb, C4.5-rules)	(f,CI)	(f,C4.5)	(f,ID3)
#0		◇		◇	◇	◇		◇	◇
#1		◇	◇	◇		◇		◇	
#2		◇		◇		◇	◇		
#3			◇	◇			◇	◇	◇
#4			◇	◇				◇	
#5		◇				◇	◇	◇	
#6			◇	◇	◇	◇			
#7				◇				◇	◇
#8		◇		◇	◇		◇	◇	
#9			◇		◇	◇			
#10		◇		◇		◇	◇	◇	◇
#11	◇		◇	◇			◇	◇	◇
#12	◇			◇		◇			
#13	◇				◇	◇	◇		◇
#14				◇		◇			◇
#15				◇		◇		◇	
#16	◇		◇	◇		◇	◇	◇	◇
#17		◇		◇			◇	◇	◇
#18	◇			◇			◇		
Total 19	5	7	7	16	5	12	10	11	9
100%	26.32%	36.84%	36.84%	84.21%	26.32%	63.16%	52.63%	57.89%	47.37%

Table 5.10.3: 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
hepatiti 10-strat-cv	$\mathcal{C}4.5$	$\mathcal{CN}2$	$\mathcal{C}4.5$-rules
all features	20.62±2.27	18.25±3.83	21.29±2.99
FSS(wf,$\mathcal{C}4.5$)	15.50±2.00		
FSS(wb,$\mathcal{C}4.5$)	11.62±1.62		
FSS(wf,$\mathcal{CN}2$)		9.01±1.66	
FSS(wb,$\mathcal{CN}2$)		12.87±2.81	
FSS(wf,$\mathcal{C}4.5$-rules)			26.04±4.01
FSS(wb,$\mathcal{C}4.5$-rules)			25.96±3.06
FSS(f,CI)	19.42±2.39	23.31±2.39	21.25±2.55
FSS(f,$\mathcal{C}4.5$)	16.79±2.00	14.87±1.95	16.81±2.93
FSS(f,ID3)	20.12±2.89	17.05±3.49	21.50±3.07

Table 5.10.4: Hepatitis – Errors

6 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.

6.1 Number of Selected Features

Table 6.1.1 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. Similar information is given in Table 6.1.2 considering the proportion and average of selected features.

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

For the wrapper approach, and not considering the zero value FSS cases, 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.*

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

Similar results were obtained in (Baranauskas and Monard, 1999), where datasets with a larger number of features were used, confirming the idea that going backwards from the full set of features would favor to capture interactive features.

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.*

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

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)
ta	5	4	4	4	4	0	0	4	5	5
bupa	6	5	5	5	5	2	0	1	6	6
pima	8	5	5	7	7	3	3	6	7	8
breast cancer2	9	5	5	4	7	4	6	8	8	9
cmc	9	4	4	5	5	2	2	9	9	9
breast cancer	10	7	7	5	9	0	0	9	8	8
smoke	13	0	6	0	7	8	8	11	13	13
hungaria	13	5	8	4	7	4	6	10	11	11
hepatitis	19	5	7	7	16	5	12	10	12	9
Total	100%	43.48%	55.43%	44.57%	72.83%	30.43%	40.22%	73.91%	85.87%	84.78%

Table 6.1.1: 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)
ta	5	80.00%	80.00%	80.00%	80.00%	0.00%	0.00%	80.00%	100.00%	100.00%
bupa	6	83.33%	83.33%	83.33%	83.33%	33.33%	0.00%	16.67%	100.00%	100.00%
pima	8	62.50%	62.50%	87.50%	87.50%	37.50%	37.50%	75.00%	87.50%	100.00%
breast cancer2	9	55.56%	55.56%	44.44%	77.78%	44.44%	66.67%	88.89%	88.89%	100.00%
cmc	9	44.44%	44.44%	55.56%	55.56%	22.22%	22.22%	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%
smoke	13	0.00%	46.15%	0.00%	53.85%	61.54%	61.54%	84.62%	100.00%	100.00%
hungaria	13	38.46%	61.54%	30.77%	53.85%	30.77%	46.15%	76.92%	84.62%	84.62%
hepatitis	19	26.32%	36.84%	36.84%	84.21%	26.32%	63.16%	52.63%	63.16%	47.37%
Average	10.11	52.04%	60.91%	52.67%	75.12%	28.46%	33.03%	74.97%	90.34%	91.21%

Table 6.1.2: Proportion of Selected Features

6.2 Time for Selecting Features

All experiments were run in a standard *Indigo 2* Silicon Graphics workstation. Table 6.2.1 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)
ta	5	11.6	8.9	66.7	63.1	13.2†	30.0†	0.1	$\leq$ 0.01	0.7
bupa	6	28.7	23.7	189.7	164.1	28.3	53.9	0.1	$\leq$ 0.01	0.9
pima	8	81.9	89.2	1292.1	790.7	172.5	234.7	0.4	0.1	2.1
breast cancer2	9	69.7	51.7	312.5	283.2	49.8	139.9	0.2	$\leq$ 0.01	1.1
cmc	9	170.1	289.7	4801.3	4907.7	270.2	1985.3	0.6	0.2	5.5
breast cancer	10	116.4	85.9	606.6	723.3	55.0†	227.0†	0.4	1.2	1.6
smoke	13	671.9†	1016.0	1084.1†	35408.4	17082.9	2975.0	1.8	2.0	11.5
hungaria	13	83.6	104.8	314.2	1242.9	118.5	392.6	0.4	$\leq$ 0.01	0.9
hepatitis	19	77.2	149.6	700.4	583.0	138.3	310.7	0.7	$\leq$ 0.01	0.6
Total Time		1311.1	1819.5	9367.6	44166.4	17928.7	6349.1	4.7	3.5	24.9

Table 6.2.1: 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 with the $\leq$0.01 entry.

Considering the minimum and maximum time taken by the wrapper approach (8.9s for FSS(wb,$\mathcal{C}$4.5) using dataset ta and 35408.4s for FSS(wf,$\mathcal{CN}$2) using dataset hungaria repectivaly) and the maximum time taken by the filter approach (11.5s for FSS(f,ID3) using dataset smoke) it can be observed that the wrapper approach takes, in the best case, almost the same time as the filter approach worst case. However, considering worst cases here, the wrapper approach is 3079 times slower than the filter

approach.

It is also expected that forward selection should be faster than backward selection, since building classifiers when there are few features in the training data should be computationally faster. On the average, forward selection was faster than backward selection, although this is not true for each individual case.

Considering the wrapper approach, the time taken is related to the algorithm wrapped around, since this algorithm is executed several times. Table 6.2.2 shows the time taken by the three algorithms used in this approach for running ten-fold cross-validation and ten-fold stratified cross-validation using all features in the dataset.

Dataset	$\mathcal{C}4.5$	$\mathcal{CN}2$	$\mathcal{C}4.5$-rules
10-cv			
ta	0.5	6.9	2.1
bupa	1.6	8.1	2.7
pima	4.2	26.0	7.3
breast cancer2	1.3	8.0	2.6
cmc	5.6	133.5	100.8
breast cancer	3.2	13.8	7.2
smoke	13.5	443.9	533.1
hungaria	2.0	12.2	3.6
hepatitis	1.1	5.0	2.2
Average	3.7	73.0	73.5
10-strat-cv			
ta	0.6	7.1	1.9
bupa	1.7	8.1	3
pima	4.3	24.6	7.8
breast cancer2	1.4	8.2	2.7
cmc	6.0	130.2	102.0
breast cancer	3.3	14.0	540.9
breast cancer	14.2	436.2	3.5
smoke	2.1	12.7	3.7
hungaria	1.2	5.1	2
hepatitis			
Average	3.9	71.8	74.2

Table 6.2.2: Time Taken by $\mathcal{C}4.5$, $\mathcal{C}4.5$-rules and $\mathcal{CN}2$ for Running Ten-Fold Cross-Validation and Ten-Fold Stratified Cross-Validation Using all Features

As can be observed, the $\mathcal{CN}2$ inducer is the slowest.

6.3 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 five bars.

Each bar corresponds to the mean error divided by the standard deviation where ten-fold stratified cross-validation has been used. When the length of the bars are higher 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 2 and 3. The difference in standard deviations is given by

Equation 4.

$$\mathrm{m}(A_2 - A_1) = \mathrm{m}(A_2) - \mathrm{m}(A_1) \tag{2}$$

$$\mathrm{sd}(A_2 - A_1) = \sqrt{\frac{\mathrm{sd}(A_2)^2 + \mathrm{sd}(A_1)^2}{2}} \tag{3}$$

$$\mathrm{ad}(A_2 - A_1) = \frac{\mathrm{m}(A_2 - A_1)}{\mathrm{sd}(A_2 - A_1)} \tag{4}$$

Table 6.3.1 shows the results obtained by Equation 4, 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, $\mathcal{C}4.5$ and Column Importance used as filter 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) $-\mathcal{C}4.5$
ta	-0.52	-0.52	-0.52	0.00	0.00
bupa	0.76	0.76	4.05	0.00	0.00
pima	-0.47	-1.60	1.51	0.26	0.00
breast cancer2	-1.79	-1.79	-0.19	-1.18	0.00
cmc	-6.55	-6.55	0.00	0.00	0.00
breast cancer	-0.56	-0.56	-0.46	0.00	-1.40
smoke	-1.33	-2.14	-2.08	0.00	0.00
hungaria	-1.45	-1.45	-0.69	0.08	0.09
hepatitis	-2.39	-4.56	-0.51	-1.79	-0.19

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) $-\mathcal{CN}2$
ta	-0.54	-0.54	1.97	0.00	0.00
bupa	1.02	1.02	5.39	0.00	0.00
pima	-0.09	-0.09	0.44	-0.11	0.00
breast cancer2	-5.08	-2.18	-0.44	-1.18	0.00
cmc	-2.88	-2.88	0.00	0.00	0.00
breast cancer	-2.94	-2.94	0.00	-1.01	-0.38
smoke	-2.12	0.00	6.33	0.00	0.00
hungaria	-2.02	-0.77	-0.45	0.19	-0.70
hepatitis	-3.13	-1.60	1.59	-1.11	-0.33

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
ta	-4.52	-4.52	-0.70	0.00	0.00
bupa	6.25	6.25	3.35	0.00	0.00
pima	8.47	8.47	1.62	0.00	0.00
breast cancer2	2.75	1.10	0.65	0.83	0.00
cmc	13.32	13.32	0.00	0.00	0.00
breast cancer	42.55	42.55	0.00	0.00	0.00
smoke	1.76	2.00	-0.47	0.00	0.00
hungaria	6.61	0.57	-0.17	-0.17	0.11
hepatitis	1.34	1.54	-0.01	-1.51	0.07

Table 6.3.1: Difference in Standard Deviations of Errors

Figures 6.3.1, 6.3.2 and 6.3.3 show the corresponding graphs from Table 6.3.1.

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 three bars correspond to the algorithms used as filters against no feature selection.

Considering graphs from Figures 6.3.1 and 6.3.2, it can be observed that the wrapper approach outperforms the standard inducer 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

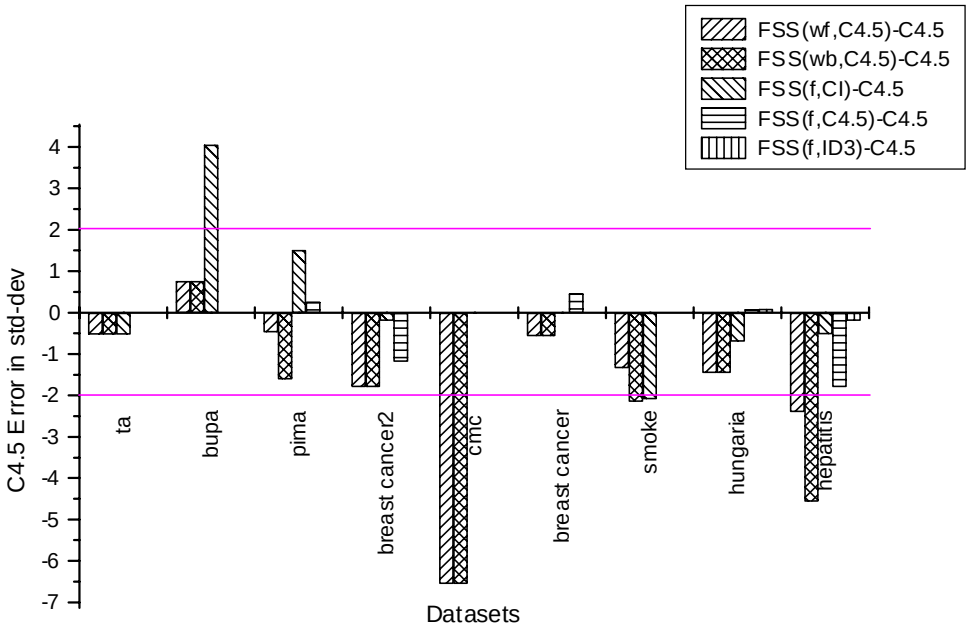

Figure 6.3.1: $\mathcal{C}4.5$ Difference in Standard Deviations of Errors

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 6.3.1:

- 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, one case where the standard inducer outperformed the CI used as filter
- For the smoke dataset, the CI used as filter outperformed the standard inducer once

Similarly for $\mathcal{CN}2$, we have —see Figure 6.3.2:

- For datasets bupa and smoke, the standard inducer outperformed the filter approach in 2 cases
- For breast cancer2, cmc, breast-cancer, smoke, hungaria and hepatitis, the wrapper approach out performed the standard inducer in 9 cases

However, for $\mathcal{C}4.5$-rules — see Figure 6.3.3, 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 6.3.2 shows improved accuracies at the significance level (95% confidence) for wrapper forward and backward selection compared with standard inducers: $\mathcal{C}4.5$, $\mathcal{C}4.5$-rules and $\mathcal{CN}2$.

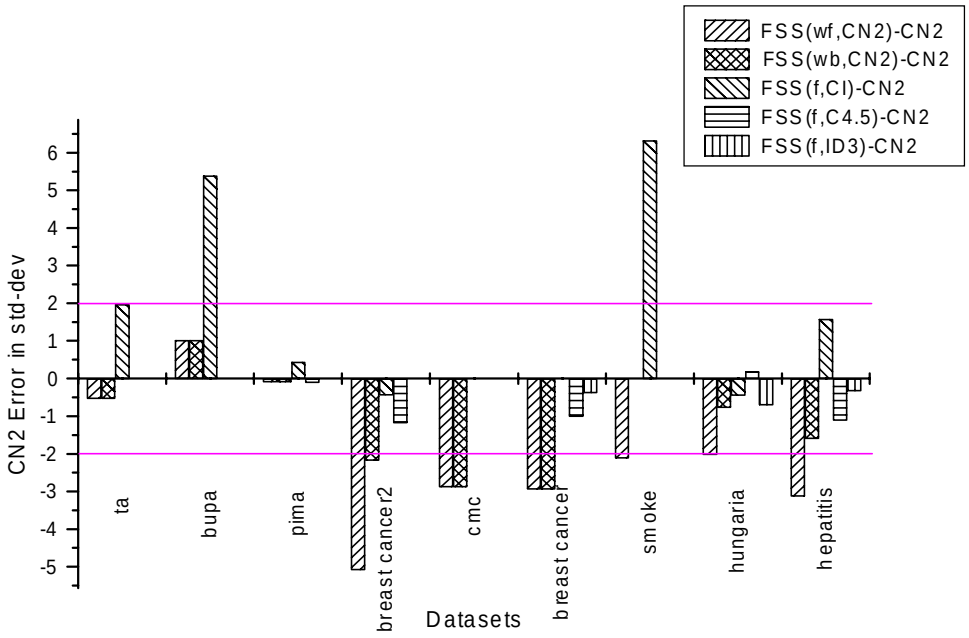

Figure 6.3.2: $\mathcal{CN}2$ Difference in Standard Deviations of Errors

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	△	▽
ta			△			△				2	0
bupa			▽			▽	▽	▽	▽	0	5
pima			▽			▽				0	2
breast cancer2		△	▽		△					2	1
cmc	△	△	▽	△	△	▽				4	2
breast cancer		△	▽		△	▽				2	2
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		15

Table 6.3.2: Improved Accuracies at the Significance Level

Observe that for the filter approach, Table 6.3.2 only shows the CI 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 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 6.3.2, 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.

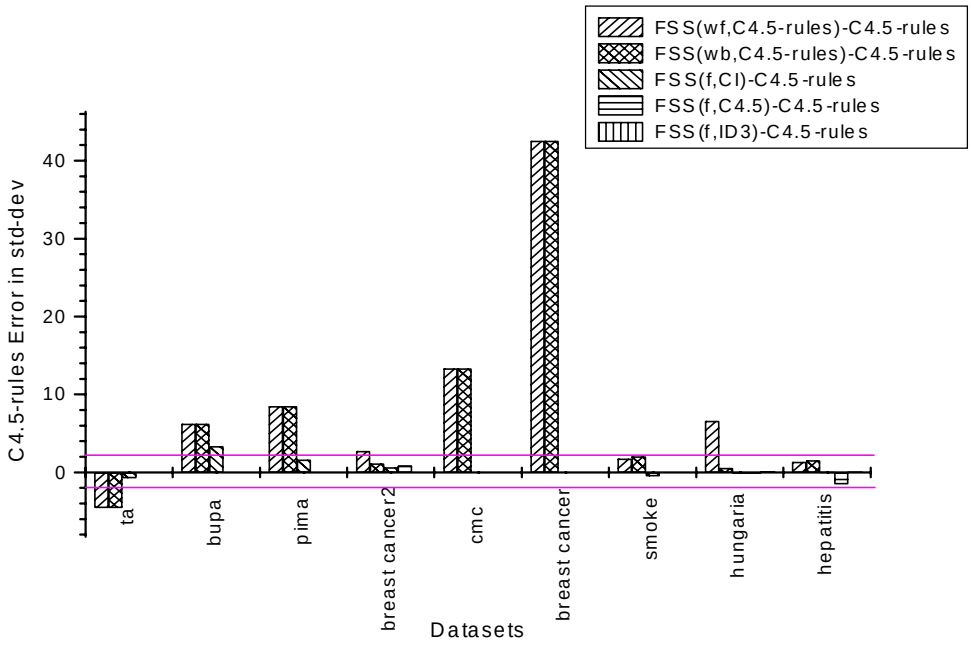

Figure 6.3.3: $\mathcal{C}4.5$-rules Difference in Standard Deviations of Errors

One possible explanation of $\mathcal{C}4.5$-rules behavior is that $\mathcal{C}4.5$-rules inducer generalizes the rules which represent the decision tree constructed by $\mathcal{C}4.5$. The generalization process is such that it leads to a production rule classifier that is usually about as accurate as a pruned tree, but more easily understood by people (Quinlan, 1993). A possible explanation could be that in the presence of few features, this generalization process increments the error rate. Futhermore, if we consider the only two cases where the wrapper approach outperform $\mathcal{C}4.5$-rules, *i.e.* wf and wb for ta dataset, we can see that the proportion of selected features by both wrappers is 0% — see Table 6.1.2 — pg. 21. This means that the default accuracy is superior to $\mathcal{C}4.5$-rules accuracy.

Finally, considering the total number of times that an improved accuracy at the 95% confidence level was found — row $\#\triangle$ in Table 6.3.2 — we can see that the best ranked FSS methods are forward and backward selection using as black boxes $\mathcal{CN}2$ ($\#\triangle = 6 + 3$) and $\mathcal{C}4.5$ ($\#\triangle = 2 + 3$) inducers.

Table 6.3.3 shows the difference in standard deviations of errors for both wrappers and considering only the $\mathcal{C}4.5$ and $\mathcal{CN}2$ inducer.

Dataset	FSS(wf,$\mathcal{C}4.5$) -FSS(wb,$\mathcal{C}4.5$)	FSS(wf,$\mathcal{C}4.5$) -FSS(wf,CN2)	FSS(wf,$\mathcal{C}4.5$) -FSS(wb,CN2)	FSS(wb,$\mathcal{C}4.5$) -FSS(wf,CN2)	FSS(wb,$\mathcal{C}4.5$) -FSS(wb,CN2)	FSS(wf,CN2) -FSS(wb,CN2)
ta	0.00	-1.93	-0.20	-1.93	-0.20	1.97
bupa	0.00	-0.35	4.38	-0.35	4.38	5.39
pima	-1.17	0.12	0.66	1.11	1.86	0.44
breast cancer2	0.00	3.91	4.00	3.91	4.00	-0.44
cmc	0.00	7.13	7.13	7.13	7.13	0.00
breast cancer	0.00	0.75	0.75	0.75	0.75	0.00
smoke	-0.05	2.12	6.41	5.69	9.28	6.33
hungaria	0.00	1.59	1.22	1.59	1.22	-0.45
hepatitis	-2.13	0.90	3.54	2.25	5.73	1.59

Table 6.3.3: Difference in Standard Deviations of Errors Between the Best Ranked FSS Methods

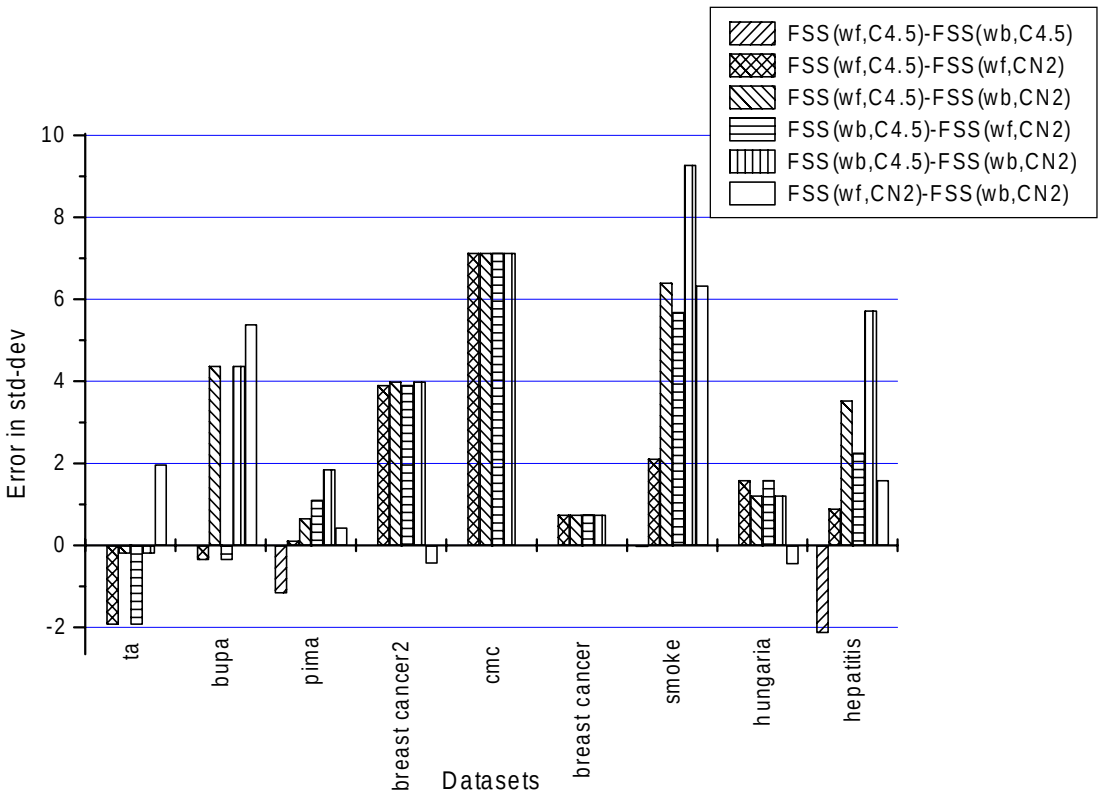

Figure 6.3.4: Difference in Standard Deviations of Errors Between Best Ranked FSS Methods

Figure 6.3.4 presents the graph for this table. For each dataset, the first bar in the graph shows the comparison of FSS(wf,$\mathcal{C}$4.5) against FSS(wb,$\mathcal{C}$4.5). The second bar corresponds to the comparison between FSS(wf,$\mathcal{C}$4.5) and FSS(wf, $\mathcal{CN}$2). The third bar shows the comparison of FSS(wf,$\mathcal{C}$4.5) against FSS(wb,$\mathcal{CN}$2). The fourth bar corresponds to the comparison between FSS(wb,$\mathcal{C}$4.5) against FSS(wf,$\mathcal{CN}$2). The last two bars show, respectively, the comparison between FSS(wb,$\mathcal{C}$4.5) against FSS(wb,$\mathcal{CN}$2) and the comparison between FSS(wf,$\mathcal{CN}$2) against FSS(wb,$\mathcal{CN}$2).

Examining Figure 6.3.4. it can be observed that for the 54 comparisons made, 17 of them showed that the wrapper approach, using $\mathcal{CN}$2 as black box, outperformed the wrapper using $\mathcal{C}$4.5 as black box; 2 cases showed that the FSS(wb,$\mathcal{CN}$2) was better than FSS(wf,$\mathcal{CN}$2).

Of these 17 cases, 7 presented the $\mathcal{CN}$2 forward selection as being better than when using $\mathcal{C}$4.5 as black box and 10 cases presented the backward selection as being better.

Only one case showed an outperformance of the $\mathcal{C}$4.5 selection above the $\mathcal{CN}$2 selection.

7 Conclusions

This work describes empirical results using the wrapper and filter approaches for Feature Subset Selection. As standard inducers for the wrapper approach, we used $\mathcal{C}4.5$, $\mathcal{C}4.5$-rules and $\mathcal{CN}2$ and for the filter approach, $\mathcal{C}4.5$, ID3 and the CI MineSet™ facility. All these inducers were run using the $\mathcal{MLC}$++ library, with its default options setting, on nine real world datasets.

At a conceptual level, the problem of FSS 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 remove features which are not essential since ML algorithms do not work well in the presence of many features. Futhermore, FSS can improve comprehensibility and can reduce the cost of processing huge quantities of data.

It should be observed that we have considered all the errors of equal importance not paying attention to unbalanced number of examples (Batista et al., 1999). 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).

Althought in this work the maximum number of features in a dataset was nineteen, we could observe that the time taken by the wrapper to select features is much times 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 and Monard, 1998; Baranauskas and Monard, 1999; Baranauskas et al., 1999a; Baranauskas et al., 1999b) where some experiments were done on datasets with a much larger number of features.

We are currently planning the application of Constructive Induction on some of the datasets used in this work. The objective of the next work is to compare the results obtained with the construction of new features with the results obtained in this work.

References

Aha, D. W. (1997). Lazy learning. *Artificial Intelligence Review*, 11:7–10.

Baranauskas, J. A. and Monard, M. C. (1998). Experimental feature selection using the wrapper approach. In *International Conference in Data Mining*, pages 161–170. WIT Press.

Baranauskas, J. A. and Monard, M. C. (1999). The $\mathcal{MLC}$++ wrapper for feature subset selection using decision tree, production rule, instance-based and statistical inducers: Some experimental results. Technical Report 87, ICMC-USP.

Baranauskas, J. A., Monard, M. C., and Horst, P. S. (1999a). Evaluation of CN2 induced rules using the wrapper approach. In *Proceedings Argentine Symposium on Artificial Intelligence – ASAI 99, part of JAIIO 99, the 28th International Conference of the Argentine Computer Science and Operational Research Society (SADIO)*, pages 141–154.

Baranauskas, J. A., Monard, M. C., and Horst, P. S. (1999b). Evaluation of feature selection by wrapping around the CN2 Inducer. In *Anais II Encontro de Inteligência Artificial – ENIA 99*, pages 315–326.

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.

Blake, C., Keogh, E., and Merz, C. (1998). UCI Irvine Repository of Machine Learning Databases. http://www.ics.uci.edu/~mlearn/MLRepository.html.

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

Bull, S. (1994). Analysis of attitudes toward workplace smoking restrictions. *Case Studies in Biometry*, pages 249–271.

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.

Felix, L. C. M., Rezende, S., Doi, C. Y., de Paula, M. F., and Romanato, M. J. (1998). $\mathcal{MLC}$++ biblioteca de aprendizado de máquina em $\mathcal{C}$++. Technical Report 72, ICMC-USP.

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 $\mathcal{C}$++*. 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.

Loh, W. and Shih, Y. S. (1997). Split selection methods for classification trees. *Statistica Sinica*, pages 815–840.

Mangasarian, O. L. and Wolberg, W. H. (1990). Cancer diagnosis via linear programming. *SIAM News*, 23(5):1–18.

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

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

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

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 and K-fold Stratified Cross-Validation

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

#!/bin/csh
#
# Author: Jose Augusto Baranauskas (jaugusto@icmc.sc.usp.br)
#         LABIC-ICMC-USP
#
# Summary: This script runs the MLC++ accuracy estimation AccEst in
# several datasets with several inducers. Accuracies are estimated
# using cross-validation (cv) and stratified-cross-validation
# (strat-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.accest" containing in each line one dataset name,
#        without extension (.names, .data and .test assumed)
#     b) file "inducers.accest" 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.accest file. If not
#        present, this file will be created by this script with
#        "all" features.
#        If user supplied, this file must contain, in each row,
#        a feature set separated by blank spaces. So, for each
#        feature set, this script will estimate accuracy for
#        the dataset.
#
# pos:
#     a) files $dataset.$inducer.accest.out, 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 and
#        strat-cv evaluation for each feature set present in the
#        $dataset.fss 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.accest`)
  # if there is no $dataset.fss file then use ALL features to
  # for accuracy estimation
  set fssfile = $dataset.fss
  if (! (-e $fssfile)) then
    echo "all" > $fssfile
  endif

  # Accuracy estimation
  foreach inducer (`cat inducers.accest`)
    set outfile = $dataset.$inducer.accest.out
    set Nfss = `cat $fssfile | wc -li`

    set stime = `date`
    echo "Start time.: $stime" > $outfile
    echo "Inducer....: $inducer" >> $outfile
    echo "Dataset....: $dataset" >> $outfile
    echo "FSS file...: $fssfile" >> $outfile
    echo "No. of FSS.: $Nfss" >> $outfile
    echo "Working dir: `pwd`" >> $outfile
    echo "Output file: $outfile" >> $outfile

    setenv INDUCER $inducer

    set i = 0
    while ($i < $Nfss)
      setenv DATAFILE $dataset.data
      setenv NAMESFILE $dataset.names
      setenv TESTFILE $dataset.test
      set i = `expr $i + 1`
      # get line $i in the feature file
      set featureset = `cat $fssfile | sed -n $i"p"`
```

```
      echo "-------------------------------------------" >> $outfile
      echo "FSS $i : $featureset" >> $outfile

      if ("$featureset" != "all") then
        setenv DUMPSTEM $dataset.$inducer.tmp
        libproject
        echo "$featureset -1" | project > /dev/null
        setenv DATAFILE $dataset.$inducer.tmp.data
        setenv NAMESFILE $dataset.$inducer.tmp.names
        setenv TESTFILE $dataset.$inducer.tmp.test
      endif

      libAccEst

      # Cross-validation
      setenv ACC_ESTIMATOR cv
      setenv CV_FOLDS $folds
      set stimeAccEst = `date`
      echo "-------------------------------------------" >> $outfile
      set et = `time AccEst >>& $outfile`
      echo "-------------------------------------------" >> $outfile
      echo "Start time.....: $stimeAccEst" >> $outfile
      echo "Stop time......:" `date` >> $outfile
      echo "Execution time : $et" >> $outfile
      echo "===========================================" >> $outfile

      # Stratified cross-validation
      setenv ACC_ESTIMATOR strat-cv
      setenv CV_FOLDS $folds
      set stimeAccEst = `date`
      echo "-------------------------------------------" >> $outfile
      set et = `time AccEst >>& $outfile`
      echo "-------------------------------------------" >> $outfile
      echo "Start time.....: $stimeAccEst" >> $outfile
      echo "Stop time......:" `date` >> $outfile
      echo "Execution time : $et" >> $outfile
      echo "===========================================" >> $outfile
    end #  featureset
    rm $dataset.$inducer.tmp.*
  end # inducer
end # dataset
```

A.2 Forward Wrapper Approach

```
fss-forw <loglevel>

#!/bin/csh
#
# Author: Jose Augusto Baranauskas (jaugusto@icmc.sc.usp.br)
#         LABIC-ICMC-USP
#
# Summary: This script runs the MLC++ wrapper using forward selection in
# several datasets and using several black box inducers supplied
# to the wrapper. Results are kept in files for later user evaluation.
#
# arguments:
#     a) MLC++ loglevel (optional)
#
# pre:
#     a) file "datasets" containing in each line one dataset name,
#        without extesion (.names, .data and .test assumed)
#     b) file "inducers" containing in each line one MLC++ inducer to
#        be wrapped forward around.
#
# pos:
#     a) files $dataset.fss.forw.$inducer.out, for each $dataset in the
#        "dataset" file and for each $inducer in the "inducers" file. Each
#        output file contains the MLC++ wrapper output.
#
# 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
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

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

foreach dataset (`cat datasets`)
  foreach inducer (`cat inducers`)
    set outfile = $dataset.fss.forw.$inducer.out
    set stime = `date`
    echo "Start time.: $stime" > $outfile
    echo "Inducer....: $inducer" >> $outfile
    echo "FSS Inducer: $fss_inducer" >> $outfile
    echo "Dataset....: $dataset" >> $outfile
    echo "Working dir: `pwd`" >> $outfile
```

```
    echo "Output file: $outfile" >> $outfile
    setenv INDUCER fss
    setenv FSS_INDUCER $inducer
    setenv DATAFILE $dataset.data
    setenv NAMESFILE $dataset.names
    setenv TESTFILE $dataset.test

    libinfo
    echo "--------------------------------------------" >> $outfile
    set et = `time Inducer >>& $outfile`
    echo "--------------------------------------------" >> $outfile
    echo "Start time.....: $stime" >> $outfile
    echo "Stop time......:" `date` >> $outfile
    echo "Execution time : $et" >> $outfile
  end # inducer
end # dataset
```

A.3 Backward Wrapper Approach

```
fss-back <loglevel>

#!/bin/csh
#
# Author: Jose Augusto Baranauskas (jaugusto@icmc.sc.usp.br)
#         LABIC-ICMC-USP
#
# Summary: This script runs the MLC++ wrapper using backward selection in
# several datasets and using several black box inducers supplied
# to the wrapper. Results are kept in files for later user evaluation.
#
# arguments:
#     a) MLC++ loglevel (optional)
#
# pre:
#     a) file "datasets" containing in each line one dataset name,
#        without extesion (.names, .data and .test assumed)
#     b) file "inducers" containing in each line one MLC++ inducer to
#        be wrapped backward around.
#
# pos:
#     a) files $dataset.fss.back.$inducer.out, for each $dataset in the
#        "dataset" file and for each $inducer in the "inducers" file. Each
#        output file contains the MLC++ wrapper output.
#
# 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
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

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

# Set MLC++ wrapper to backward search
setenv FSS_DIRECTION backward

foreach dataset (`cat datasets`)
  foreach inducer (`cat inducers`)
    set outfile = $dataset.fss.back.$inducer.out
    set stime = `date`
    echo "Start time.: $stime" > $outfile
```

```
    echo "Inducer....: $inducer" >> $outfile
    echo "FSS Inducer: $fss_inducer" >> $outfile
    echo "Dataset....: $dataset" >> $outfile
    echo "Working dir: `pwd`" >> $outfile
    echo "Output file: $outfile" >> $outfile
    setenv INDUCER fss
    setenv FSS_INDUCER $inducer
    setenv DATAFILE $dataset.data
    setenv NAMESFILE $dataset.names
    setenv TESTFILE $dataset.test

    libinfo
    echo "--------------------------------------------" >> $outfile
    set et = `time Inducer >>& $outfile`
    echo "--------------------------------------------" >> $outfile
    echo "Start time.....: $stime" >> $outfile
    echo "Stop time......:" `date` >> $outfile
    echo "Execution time : $et" >> $outfile
  end # inducer
end # dataset
```

A.4 Filter Approach

```
fss-filter <loglevel>

#!/bin/csh
#
# Author: Jose Augusto Baranauskas (jaugusto@icmc.sc.usp.br)
#         LABIC-ICMC-USP
#
# Summary: This script runs MLC++ inducers (as filters) in several datasets.
# In general, only the features are significant and the inducer used as filter
# is discarded. Results are kept in files for later user evaluation.
#
# arguments:
#     a) MLC++ loglevel (optional)
#
# pre:
#     a) file "datasets" containing in each line one dataset name,
#        without extesion (.names, .data and .test assumed)
#     b) file "filters" containing in each line one MLC++ inducer to
#        be used as filter.
#
# pos:
#     a) files $dataset.filter.$filter.out, for each $dataset in the
#        "dataset" file and for each $filter in the "filters" file. Each
#        output file contains the inducer (used as filter) output.
#
# 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
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
alias libInducer 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

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

setenv DISPLAY_STRUCT ascii
setenv DISP_CONFUSION_MAT ascii

foreach dataset (`cat datasets`)
  foreach filter (`cat filters`)
    set outfile = $dataset.filter.$filter.out
    set stime = `date`
    echo "Start time.....: $stime"       > $outfile
    echo "Filter Inducer: $filter"       >> $outfile
```

```
    echo "Dataset.......: $dataset"     >> $outfile
    echo "Working dir...: `pwd`" >> $outfile
    echo "Output file...: $outfile"     >> $outfile
    setenv INDUCER $filter
    setenv DATAFILE $dataset.data
    setenv NAMESFILE $dataset.names
    setenv TESTFILE $dataset.test

    libInducer
    echo "--------------------------------------------" >> $outfile
    set et = `time Inducer >>& $outfile`
    echo "--------------------------------------------" >> $outfile
    echo "Start time.....: $stime" >> $outfile
    echo "Stop time......:" `date` >> $outfile
    echo "Execution time : $et" >> $outfile
  end # filter
end # dataset
```

A.5 Column Importance Facility

fss-ci

```
#!/bin/csh
#
# Author: Jose Augusto Baranauskas (jaugusto@icmc.sc.usp.br)
#         LABIC-ICMC-USP
#
# Summary: This script runs the MineSet(TM) Column Importance Mining Tool
# in several datasets. Results are kept in files for later user evaluation.
#
# arguments:
#     none
#
# pre:
#     a) file "datasets" containing in each line one dataset name,
#        without extension (.schema assumed)
#
# pos:
#     a) files $dataset.MIndUtil.importance.out, for each $dataset in the
#        "dataset" file. Each output file contains the MineSet(TM) Column
#        Importance Mining Tool (MIndUtil_s) output
#
# 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.

setenv LOGLEVEL 1
setenv MAX_LABEL_VALS 500
setenv MODE auto-select
setenv DISC_TYPE entropy
setenv LABEL class
setenv DISC_MIN_SPLIT 0

# Select almost SELECT_N features in the dataset. This value is usually
# setup to a large value to get ALL relevant features
setenv SELECT_N 200

foreach dataset (`cat datasets`)
  set outfile = $dataset.MIndUtil.importance.out
  set stime = `date`
  echo "Start time.: $stime" > $outfile
  echo "Dataset....: $dataset" >> $outfile
  echo "Working dir: `pwd`" >> $outfile
  echo "Output file: $outfile" >> $outfile

  setenv FLAT_FILE $dataset.schema
  setenv OUTPUT_FILE $dataset.MIndUtil.auto-select.tmp

  echo "" >> $outfile
  echo "--------------------------------------------" >> $outfile
  echo "FSS Inducer: MIndUtil_s" >> $outfile
  echo "--------------------------------------------" >> $outfile
  set et = `time MIndUtil_s >>& $outfile`
  echo "--------------------------------------------" >> $outfile
```

```
  cat $OUTPUT_FILE >> $outfile
  echo "-------------------------------------------" >> $outfile
  echo "Start time.....: $stime" >> $outfile
  echo "Stop time......:" `date` >> $outfile
  echo "Execution time : $et" >> $outfile
  rm $OUTPUT_FILE
end # dataset
```