Evaluation of $\mathcal{CN}2$ Induced Rules Using Feature Selection*

José Augusto Baranauskas, Maria Carolina Monard/ILTC, and Paulo Sérgio Horst

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

Abstract Selecting a good subset of features is an important issue in Machine Learning and Data Mining. The Feature Subset Selection problem can be defined as finding a minimum set of k relevant features that describe the dataset as well as the original m features do, where $k \leq m$. Several approaches have been pursued in Machine Learning for feature selection, among them filter and wrapper approaches. In the first case features are selected independently of the induction algorithm while in the wrapper approach the same inducer that will use the feature subset provides the estimated accuracy for selection.
In this work we address the filter approach through $\mathcal{ID}3$ decision trees and the MineSet™ Column Importance mining tool and the wrapper approach through the wrapper method implemented in the $\mathcal{MLC}$++ library. Experimental results of its application on several real world datasets using the $\mathcal{CN}2$ symbolic inducer are presented and discussed, considering not only accuracy on evaluating results but also the sort of rules generated by $\mathcal{CN}2$ for each approach. In general, they confirm the superiority of the wrapper approach. However, for high dimensionality data the wrapper computational cost is excessive and, in these cases, filters can be considered as reasonable substitutes.

1 Introduction

One important issue to be considered when using Machine Learning — ML — in Data Mining is related to pre-processing the data before it is presented to the learning algorithm. Feature transformation and feature selection are usually applied in data pre-processing for real world applications.

Feature transformation, also known as constructive induction, is the process through which a new set of features is created from the original ones, thus augmenting the feature's space. On the other hand, feature selection does not generate new features but selects a subset of the original features, reducing the

* This work was partially supported by the Brazilian Cooperative Research Network (RECOPE/FINEP), CAPES, FMRP-USP and FAEPA-HCFMRP-USP.

feature's space. The data subset resulting from feature selection maintains the same number of instances while keeping only a subset of features such that its predictive performance is similar, or better, to the one obtained using the full set of features. This is known as the Feature Subset Selection — FSS — problem.

There are several reasons for doing FSS before applying Machine Learning algorithms. One of them is that it can improve accuracy since many inducers degrade in performance when given too many features. Another reason is that FSS often improves comprehensibility, which is the ability for humans to understand the classification rules induced by symbolic learning algorithms. Furthermore, FSS can reduce measurement cost since in some domains measuring features may be expensive. Still, finding the *best* feature subset typically involves NP-hard optimization problems that are approximated by heuristic search for a *good* feature subset, and numerous search heuristics have been proposed.

There are three main approaches that have been pursued for FSS, known as embedded, filter and wrapper approaches. In the first case, the feature selection process is embedded within the basic induction algorithm; some ML algorithms are capable of doing their own FSS while searching for a solution. The filter approach filters out features before using the induction algorithm whereas the wrapper approach generates a set of candidate features and uses the induction algorithm as a black box to evaluate the feature set (Blum & Langley, 1997).

As induction algorithms differ considerably in their emphasis on focusing on relevant features, it is expected that in most cases the wrapper approach should provide a better feature subset than the filter approach. Nevertheless, the computational cost of wrapper methods can be very high because they have to call the induction algorithm for each feature set considered.

In this work we address the filter and wrapper approaches using two different methods for filtering, $\mathcal{ID}3$ decision trees and the MineSet$^{\text{TM}}$ Column Importance mining tool, and the $\mathcal{MLC}$++ FSS wrapper. Experimental results of their application on eleven real world datasets using the $\mathcal{CN}2$ symbolic inducer are presented and discussed. Although, as expected, accuracy results confirm the superiority of the wrapper approach, for high dimensionality data the wrapper computational cost is excessive. In these cases, filters can be considered as reasonable substitutes.

This work is organized as follows. In Sect. 2 we describe the feature subset selection problem along with filter and wrapper approaches for FSS used in this work. In Sect. 3 we present the experimental methodology and describe in detail the experiments carried out for measuring both approaches. Section 4 reports results and discussion. Finally, Sect. 5 presents our conclusions.

2 The Feature Subset Selection Problem

In supervised learning, an inducer is given a set of n training instances and each instance $\mathbf{X}$ is an element of the set $F_1 \times F_2 \times \ldots \times F_m$ where F_j is the domain of the jth feature. Training instances are tuples $(\mathbf{X}, Y)$ where Y is the label, output or class. The Y values are typically drawn from a discrete set of classes

$\{1, \ldots, K\}$ in the case of *classification* or from the real values in the case of *regression*. In this work we will refer to classification. Given a set of training instances the learning algorithm (*inducer*) outputs a *classifier* such that, given a new instance, it accurately predicts the label Y.

One of the central problems in supervised learning is the selection of useful features. Although most learning methods attempt to either select features or assign them degrees of importance, both theoretical analysis and experimental studies indicate that many algorithms scale poorly to domains with large numbers of irrelevant features. This suggests the need for additional methods to select a useful subset of features when many of them are available (Langley, 1996).

The general FSS approach can be considered as a state space search, with each node (state) in the search space specifying a subset of possible features. The space defined is partially ordered considering each child state including one more feature than its parent. According to this view, any FSS method can be characterized by: start node (or nodes) in the space; nodes evaluation; organization of the search and halting criterion.

Having a partial ordering on the FSS space, there are two immediate approaches that determine the starting node as well as the direction of the search: *forward* and *backward selection*. In forward selection, the starting point is an empty set of features and features are added successively. In backward selection, the starting point contains all features and features are removed successively. Several other variations can be used such as selecting any other starting point and move outward from it.

Related to the value (or *goodness*) of a node, several metrics can be used, such as feature's ability to discriminate among classes, the estimated prediction accuracy and others.

Other important issue is the organization of the search. An exhaustive search of the space is impractical when many features are present since there are 2^m possible subsets (states) of m attributes. More realistic methods such as hill-climbing or best-first search may be used to traverse the space. Finally, as the goal node is unknown, some other halting criterion should be defined.

As stated ealier approaches for feature selection that have been developed can be grouped into three categories: *embed,filter* and *wrapper*. It follows a brief description of filter and wrapper approaches for feature selection addressed in this work.

2.1 The Filter Approach

In this approach, features are filtered independently of the induction algorithm. This means that FSS is done as a pre-processing step, totally ignoring the effects of the selected features subset on the performance of the induction algorithm.

Any filtering algorithm can be used as a FSS filter to select features in large feature space for other inducers that take more time to search their solution space. The set of features selected by the filtering algorithm is the output of the FSS filter process. The remaining unused features are then deleted from the

dataset, reducing its dimension. After that, this reduced dataset can be used by any other inducer. However, features that are considered good for a given filtering algorithm are not necessarily useful for other family of algorithms that may have an entirely different inductive bias.

For filtering we have used two methods for filtering: $\mathcal{MLC}$++ $\mathcal{ID}3$ decision trees inducer (Quinlan, 1986) and MineSet™ Column Importance — $\mathcal{CI}$ — mining tool (Rathjens, 1996). $\mathcal{MLC}$++ is a library of $\mathcal{C}$++ classes and tools developed at Stanford University (Kohavi et al., 1994). This library provides general machine learning algorithms as well as a variety of tools that can be used by end users. Some algorithms support visual output of the classifiers and may generate output for Silicon Graphics' MineSet™ product.

The output of the $\mathcal{ID}3$ filter is simply the subset of features selected by the tree (the decision tree itself is discarded). For filtering purposes, we used $\mathcal{ID}3$ in its simplest form to obtain the covering tree using all the data, *i.e.* no pruning and no error estimation on test instances were used.

The Column Importance mining tool partitions the data in a similar way than decision trees. In $\mathcal{CI}$ a measure called *purity*, which varies from 0 to 100, informs how well features discriminate the different classes. Each set in the partition has its own purity measure, and the purity measure within the partition is a combination of these individual measures. As in decision trees, the relevance of a feature depends on the features previously considered as relevants (Rathjens, 1996, Chap. 17).

One difference between $\mathcal{CI}$ and decision trees algorithms is related to the discretization process. The Column Importance algorithm uses a global discretization process, *i.e.* it discretizes all continuous features before partitioning the data. The decision tree algorithm instead does not pre-discretize features, it finds thresholds as the tree is built making only binary splits. The $\mathcal{CI}$ discretization type used in the experiments performed was entropy, the MineSet™ default.

As with decision trees, all of the data was used by Column Importance (again, default MineSet™ settings) to find out relevant features. The $\mathcal{CI}$ tool allows to determine how important almost k ($k \leq m$) features are in discriminating the class. In our experiments we used $k = m$ which means that almost as many as existing features in the dataset could be selected by $\mathcal{CI}$.

In both filters considered the start node is the empty set of features using forward selection with add operator. The goodness of a node is the $\mathcal{ID}3$ or $\mathcal{CI}$ inductive bias based on entropy-like measures. The search method is a greedy-search. The halting criterion is when no improvement is possible with further data partitioning.

The main disadvantage of the filter approach is that it totally ignores the effects of the selected feature subset on the performance of the induction algorithm. For this reason, it is claimed that the filter approach should be replaced with the wrapper approach that takes the induction algorithm itself into account (Kohavi, 1997).

2.2 The Wrapper Approach

In contrast with filter methods, wrapper methods generate a feature subset as candidate, run the induction algorithm with only those features on the training set, and use the accuracy of the resulting description to evaluate the feature subset. This process is repeated for each feature subset candidate until the criterion for halting the search is reached.

The argument in favor of the wrapper approach is that the same inducer that will use the feature subset should provide a better-estimated accuracy for itself than by another method that may have a different inductive bias. On the other hand, the wrapper approach can be very expensive since the induction algorithm should be called for each feature set considered.

In this work, we use the FSS wrapper method also implemented in $\mathcal{MLC}$++. The FSS algorithm in $\mathcal{MLC}$++ exists as a wrapper around the induction algorithm and it is responsible for conducting the search for a good feature subset. The goodness of a candidate subset (node) is evaluated using the induction algorithm itself as a black box. The goal of the search is to find the state with the highest evaluation, using a heuristic function to guide it.

In the $\mathcal{MLC}$++ FSS wrapper, the start node can be the empty or the full set of features using forward or backward selection with either add or delete operators, respectively. The goodness of a node is the estimated prediction accuracy using cross-validation. The search method can be either hill-climbing or best-first. The halting criterion is to stop execution after a fixed number of nodes expansions that do not yield a node with a better estimated accuracy than the current best-estimated node.

As search is very expensive using only add/delete operators, compound operators, applicable to forward as well as backward selection, were introduced in $\mathcal{MLC}$++ (Kohavi & Sommerfield, 1995). They are dynamic operators that directly connect a node to nodes considered promising given the evaluation of its children. Experimental results reported in (Kohavi, 1997) show that in general the $\mathcal{MLC}$++ wrapper runs faster when compound operators are used having similar accuracy than when only add/delete operators are used.

3 Experimental Setup

This section briefly describes the inducer used in this study as well as the datasets considered on evaluating wrapper and filter approaches for feature selection.

3.1 Inducer

The $\mathcal{CN}2$ inducer (Clark & Niblett, 1989; Clark & Boswell, 1991) available in the $\mathcal{MLC}$++ library (Kohavi et al., 1996) was used as black box inducer to the wrapper algorithm for FSS. $\mathcal{CN}2$ 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 feature values, the $\mathcal{CN}2$ inducer uses the method of simply replacing unknown values with the most commonly occurring value if the feature is nominal. For continuous features, the midvalue of the most commonly occurring sub-range replaces the unknown value.

Observe that unordered rules induced by $\mathcal{CN}2$ may overlap, *i.e.* different rules may cover the same instance. In fact, to classify a new instance, all rules are tried and those which fire are collected. If more than one class is predicted by fired rules, the method used by $\mathcal{CN}2$ 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.

3.2 Datasets

Experiments were conducted on eleven real world domains. Most of datasets are from the UCI Irvine Repository (Blake et al., 1998) while dna and genetics datasets are from the StatLog Project (Taylor et al., 1994). Besides the fact they are from different domains, the chosen datasets have different dimensionality, different types of features and some of them have missing values.

Table1. Datasets Summary Descriptions

Dataset	#I	#F(c,n)	#C	Majority Error	Missing Values
bupa	345	6 (6,0)	2	42.03%	N
pima	769	8 (8,0)	2	34.98%	N
breast cancer	699	10 (10,0)	2	34.48%	Y
hungaria	294	13 (13,0)	2	36.05%	Y
crx	690	15 (6,9)	2	44.49%	Y
letter	15000	16 (16,0)	26	95.92%	Y
hepatitis	155	19 (6,13)	2	20.65%	Y
anneal	898	38 (6,32)	5	23.83%	Y
sonar	208	60 (60,0)	2	46.63%	N
genetics	3190	60 (0,60)	3	48.12%	N
dna	3186	180 (0,180)	3	48.09%	N

Table 1 summarizes the datasets employed in this study. It shows, for each dataset, the number of instances (#I), the number of features (#F) continuous (c) and nominal (n), the number of classes (#C), the majority error and if the dataset has missing values. Datasets are presented in ascending order of the number of features, as they will be in the remaining tables and figures.

3.3 Description of Experiments

It should be observed that in the following experiments no dataset has been previously pre-processed in any way, for instance by removing or replacing missing values or even by transforming features. As usual, no attempt was made to tune filters, wrappers and the standard inducer such that the experiments can be replicated elsewhere.

Filtering: Considering each dataset, all data was submitted to the $\mathcal{ID}3$ and $\mathcal{CI}$ algorithms used as filters for feature selection. Both algorithms were run with default setting for all parameters. In order to indicate the experiment we are referring to, the notation FSS(*f,inducer*) is used, where *f* indicates filter feature selection and *inducer* $\in$ {$\mathcal{ID}3$, $\mathcal{CI}$ } indicates the inducer that has been used as filter.

Wrappring: For each dataset, $\mathcal{CN}2$ was applied through the wrapper as a black box inducer. $\mathcal{CN}2$ was run with default setting for all parameters. We ran the $\mathcal{MLC}$++ wrapper using best-first search (since it is a more robust method than hill-climbing), compound operators as well as forward and backward selection, obtaining for each algorithm and dataset the best selected features. In (Baranauskas & Monard, 1998), we report some results obtained considering the wrapper approach but using only forward selection with several inducers. Initial results using backward and forward selection with the $\mathcal{CN}2$ inducer, but only a subset of the datasets considered in this work, are reported in (Baranauskas et al., 1999)

All other $\mathcal{MLC}$++ wrapper parameters were used with default values where the estimated accuracy for each node is computed using ten-fold cross-validation; the termination condition for the search is a consecutive sequence of five node expansions that do not generate a feature subset with an estimated accuracy of at least 0.1% better than the previous best subset and a feature is selected if it does not hurt performance, *i.e.* if the accuracy estimate is the same with and without that feature. Again, in order to indicate the wrapper experiment we are referring to, the notation FSS(*method*,$\mathcal{CN}2$) is used, where *method* $\in$ {wf, wb} indicating results of running $\mathcal{CN}2$ (as a black box inducer) with wrapper forward and backward selection of features, respectively.

Accuracy Estimation: Afterwards, using for each case both filters and wrappers selected features, $\mathcal{CN}2$ was applied to the correspondent reduced dataset and accuracy was measured using ten-fold stratified cross-validation. For comparisons baseline, the $\mathcal{CN}2$ standard algorithm, which has its own embed feature selection procedure, was applied to the original dataset and accuracy was also measured using ten-fold stratified cross-validation.

As the wrapper approach is very slow for datasets with a large number of features, and due to limited computational resources, we ran the wrapper once using all the data. This means that accuracy results for the wrapper, but not for $\mathcal{CN}2$ standard inducer (using all features) or those filtered, may be overly optimistic.

4 Results

In what follows it should be observed that the result for dna dataset with backward selection was not available, after running longer than 40 days in background. This result is reported as "N/A" in the tables and it has not been considered on any average calculations.

Table2. Proportion of Selected Features and Time (in seconds) for FSS

Dataset	#F	Proportion of FSS				Time for FSS			
		(wf,$\mathcal{CN}2$)	(wb,$\mathcal{CN}2$)	(f,$\mathcal{ID}3$)	(f,$\mathcal{CI}$)	(wf,$\mathcal{CN}2$)	(wb,$\mathcal{CN}2$)	(f,$\mathcal{ID}3$)	(f,$\mathcal{CI}$)
bupa	6	83.33	83.33	100.00	16.67	189.7	164.1	0.9	0.1
pima	8	87.50	87.50	100.00	75.00	1292.1	790.7	2.1	0.4
breast cancer	10	50.00	90.00	80.00	90.00	697.7	564.3	1.1	0.5
hungaria	13	30.77	53.85	84.62	84.62	314.2	1242.9	0.9	0.2
crx	15	33.33	46.67	80.00	86.67	464.4	3628.7	1.8	0.7
letter	16	56.25	56.25	100.00	75.00	33446.1	68115.1	58.5	27.0
hepatitis	19	36.84	84.21	47.37	52.63	700.4	583.0	0.6	0.2
anneal	38	60.53	89.47	26.32	18.42	87607.7	71581.7	2.0	0.8
sonar	60	20.00	83.33	23.33	35.00	5726.9	28153.0	2.5	1.6
genetics	60	8.33	80.00	75.00	86.67	42479.4	550329.7	2.5	51.8
dna	180	8.33	N/A	50.56	47.22	349135.5	N/A	31.6	240.5
Average	38.64	43.20	75.46	69.74	60.72	17291.9	72515.3	7.3	8.3

Proportion of Selected Features: Table 2 shows proportion of selected features for each of the 43 cases considered as well as time taken for running each experiment. It can be observed that the feature space has been reduced, except for FSS(f,$\mathcal{ID}3$) for bupa, pima and letter datasets, where all the features were selected by this filter.

Considering the wrapper approach, in all cases, not only on average, the number of selected features using backward selection is always greater or equal than using forward selection, *i.e.* #FSS(wb,$\mathcal{CN}2$) $\geq$ #FSS(wf,$\mathcal{CN}2$). On average, forward selection picks up 43.20% features against 75.46% on backward selection, an increasing factor of 74.68%. This result seems to confirm the idea that going backwards from the full set of features would favor to capture interactive features (Kohavi, 1997). Also, on average, filters have selected 69.74% and 60.72% of features (for FSS(f,$\mathcal{ID}3$) and FSS(f,$\mathcal{CI}$), respectively), which are intermediate values between the ones obtained by wrapper forward and backward selection.

It should be observed that in our experiments, for some datasets, a larger number of features than the ones needed to reach the halting criterion have been selected by the wrapper. This is due to the fact that we have used $\mathcal{MLC}$++ FSS wrapper default parameter setting, which is zero for the *complexity penalty* parameter. This parameter allows penalizing feature subsets with many features such that, if the accuracy of two feature subsets is the same, the subset with small number of features is preferred. However, at most one of those features would be selected if a complexity penalty factor greater than zero was used.

In fact, not only for the wrapper approach, but also for both filters studied (no pruning in $\mathcal{ID}3$ and allowing $\mathcal{CI}$ to select up to all features present in the dataset) we decided to favor larger feature subsets. The main reason is that inducers can deal with extra features but they cannot make up for features that have been discarded.

Time for Selecting Features: Still considering Table 2 the time taken, in seconds, for selecting features is also shown related to a standard *Indigo 2* Silicon Graphics workstation[1]. It is expected that wrapper forward should be less expensive than backward selection, since building classifiers when there are few

[1] Run time for dna dataset was not considered on average calculations.

features in the training data should be computationally faster. Obviously, filters were much faster than wrappers. For wrappers, although not true for each individual case, the overall picture indicates that backward is about 4 times slower than forward selection. On the other hand, on average, filters were around 10^3 times faster than wrappers. In any case, the wrapper approach is very slow. The overall time taken for running the experiments was almost 25 hours of uninterrupted CPU processing time for all datasets (not considering FSS(wb,$\mathcal{CN}2$) for dna dataset).

Accuracy: Next, we compare the $\mathcal{CN}2$ standard inducer (no FSS) with its FSS wrapper using forward and backward selection, as well as filtering by $\mathcal{ID}3$ and $\mathcal{CI}$. Table 3 shows the accuracy (mean and standard deviation of the ten-fold stratified cross-validation) for $\mathcal{CN}2$ using all features and using only those features selected by both wrappers and filters. In order to determine whether the difference between them, *i.e.* (FSS(wf,$\mathcal{CN}2$)$-\mathcal{CN}2$), (FSS(wb,$\mathcal{CN}2$)$-\mathcal{CN}2$), (FSS(f,$\mathcal{ID}3$)$-\mathcal{CN}2$) and (FSS(f,$\mathcal{CI}$)$-\mathcal{CN}2$), is significant or not, we show in Fig. 1 one graph with four consecutive bars for every dataset. Each bar corresponds to the mean accuracy divided by the standard deviation. The variance of the significance test is the average variance of both algorithms and a standard normal distribution is assumed. Any bar having length greater than two indicates that this result is significant at 95% confidence level. When the bar is above zero it means that the first algorithm outperforms the second; if the bar is below zero, then the second algorithm outperforms the first. Whenever the bar length is above (below) two it means that the first (second) algorithm significantly outperforms the second (first).

Although the quality of the results varies across datasets, in general, the wrapper approach outperforms the standard inducer. On total average, the standard inducer accuracy has improved from 79.99% to 85.20% for forward selection and to 82.11% for backward selection. This represents 26.04% and 10.59% relative reduction in error rates, respectively.

For both filters, although on average they do not outperform the standard inducer, they do not hurt badly accuracy performance. For instance, considering total average, the $\mathcal{CN}2$ accuracy decays from 79.99% to 79.98% for $\mathcal{ID}3$ filter and to 77.84% for $\mathcal{CI}$ filter representing 0.05% and 10.74% relative increasing in error rates, respectively, showing that in general for the datasets considered $\mathcal{ID}3$ outperforms $\mathcal{CI}$ filter.

Disregarding anneal dataset, there is no real damage on average accuracy using FSS(f,$\mathcal{ID}3$) even $\mathcal{CN}2$ does not have the same inductive bias than $\mathcal{ID}3$. For dna dataset both $\mathcal{ID}3$ filter and wrapper forward significantly outperform the standard inducer.

Table 4 shows improved accuracies at significance level for wrappers and filters compared with the $\mathcal{CN}2$ standard inducer. Improvements above two standard deviations are reported with $\triangle$ and those below with $\bigtriangledown$.

The overall scenario shows that the wrapper for FSS has significantly outperformed the standard inducer in 8 of 11 experiments for forward selection and in 4 of 10 experiments for backward selection for all studied domains. In contrast,

Table3. Accuracies for No FSS, Wrapper Forward and Backward Selection and Filtering by $\mathcal{ID}3$ and Column Importance

Dataset	$\mathcal{CN}2$	FSS			
		(wf,$\mathcal{CN}2$)	(wb,$\mathcal{CN}2$)	(f,$\mathcal{ID}3$)	(f,$\mathcal{CI}$)
bupa	67.82±2.11	65.81±1.83	65.81±1.83	67.82±2.11	55.63±2.40
pima	74.62±1.38	74.75±1.43	74.75±1.43	74.62±1.38	74.09±1.02
breast cancer	94.99±1.42	96.85±0.60	95.57±0.97	94.70±0.86	94.28±1.08
hungaria	77.93±3.06	83.66±2.60	80.65±3.94	79.98±2.79	80.59±1.94
crx	83.20±1.21	86.83±1.23	85.83±0.86	83.34±1.19	83.35±1.35
letter	70.34±0.30	76.17±0.20	76.17±0.20	70.34±0.30	71.56±0.33
hepatitis	81.75±3.83	90.99±1.66	87.13±2.81	82.95±3.49	76.69±2.39
anneal	90.42±1.08	96.44±0.94	96.44±0.94	85.18±0.60	82.85±1.34
sonar	71.19±3.30	81.32±3.46	75.52±2.75	72.59±3.20	72.12±2.73
genetics	79.53±1.45	89.47±0.53	83.26±1.73	78.57±2.13	76.27±1.96
dna	88.15±0.62	94.89±0.41	N/A	89.68±0.53	88.77±0.60
Average	79.99	85.20	82.11	79.98	77.84

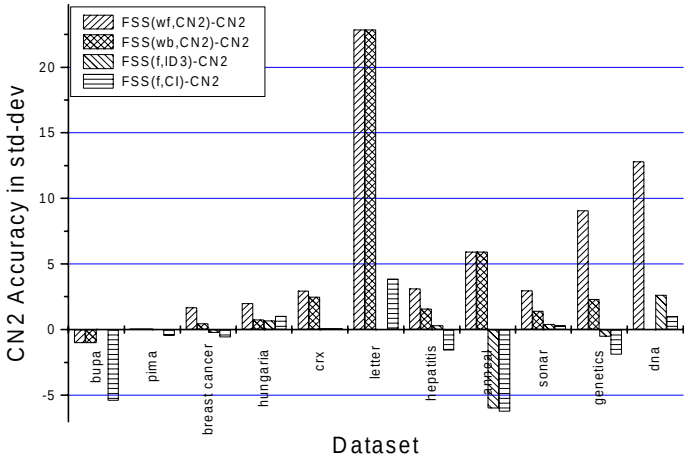

Figure1. Absolute Difference in Standard Deviations of Accuracies

$\mathcal{ID}3$ and $\mathcal{CI}$ filters have significantly outperformed the $\mathcal{CN}2$ inducer in only 1 of 11 experiments each one for dna and letter datasets, respectively. However, the standard inducer significantly outperforms filters in 3 experiments (1 for $\mathcal{ID}3$ in the anneal dataset and 2 for $\mathcal{CI}$ in bupa and anneal datasets).

Generated Rules: Besides accuracy, another interesting point is related to the sort of rules generated by $\mathcal{CN}2$ when using all features and the ones selected by filter and wrapper methods. Aiming to verify the effect of both methods in the general form (not the meaning) of the rules generated we measured, for each induced hypothesis, the average number of rule conditions (conjunctions) in the hypothesis and it was verified that they were similar.

Next, we measured the number of covered instances and the number of rules generated, shown in Tables 5 and 6, respectively. Figure 2 show from these tables, respectively, the relative difference compared with the $\mathcal{CN}2$ standard inducer.

Table4. Accuracies at Significance Level

Dataset	FSS(wf,$\mathcal{CN}2$) $-\mathcal{CN}2$	FSS(wb,$\mathcal{CN}2$) $-\mathcal{CN}2$	FSS(f,$\mathcal{ID}3$) $-\mathcal{CN}2$	FSS(f,$\mathcal{CI}$) $-\mathcal{CN}2$	Count $\triangle$	Count $\bigtriangledown$
bupa				$\bigtriangledown$	0	1
pima					0	0
breast cancer					0	0
hungaria	$\triangle$				1	0
crx	$\triangle$	$\triangle$			2	0
letter	$\triangle$	$\triangle$		$\triangle$	3	0
hepatitis	$\triangle$				1	0
anneal	$\triangle$	$\triangle$	$\bigtriangledown$	$\bigtriangledown$	2	2
sonar	$\triangle$				1	0
genetics	$\triangle$	$\triangle$			2	0
dna	$\triangle$	N/A	$\triangle$		2	0
Count $\triangle$	8	4	1	1	14	
Count $\bigtriangledown$	0	0	1	2		3

Table5. Number of Covered Instances

Dataset	#Instances	$\mathcal{CN}2$	FSS			
			(wf,$\mathcal{CN}2$)	(wb,$\mathcal{CN}2$)	(f,$\mathcal{ID}3$)	(f,$\mathcal{CI}$)
bupa	345	418	374	374	418	262
pima	769	1058	1113	1113	1058	1054
breast cancer	699	2032	1738	1949	1972	2121
hungaria	294	430	283	373	469	481
crx	690	1145.56	613.83	1133	1137.98	1121.96
letter	15000	17870	20172	20172	17870	17909
hepatiti	155	285.25	197.75	256.25	259	302.75
anneal	898	1380	1428	1412.66	1495.05	1715.9
sonar	208	344	293	305	311	287
genetics	3190	5469	4587	4491	4805	6026
dna	3186	9511	8781	N/A	9362	10099
Average	2312.18	3631.16	3598.23	3157.89	3559.73	3761.78

Although as expected results are different for each dataset, it can be observed that both methods have a tendency for increasing the number of rules in the induced hypothesis, compared with the hypothesis generated by $\mathcal{CN}2$ on its own. But, in general, it is difficult to conclude about the number of covered instances.

Other point to consider is to verify if the hypothesis induced after FSS is more general (or specific) than the hypothesis induced by $\mathcal{CN}2$ on its own. As the average number of rule conditions is similar then we considered reasonable to assume the following, were <condition> stands for *FSS method is not worst at significant level than the standard $\mathcal{CN}2$ inducer.*

Specialization if <condition> **and** the relative difference on the number of rules is above 10%.

Generalization if <condition> **and** the relative difference on the number of rules is below 10%.

Table 7 shows the cases for which each one of these rules hold where $\bigtriangledown$ indicates the cases where $\mathcal{CN}2$ outperforms, at 10% level, the method considered. In general, as the wrapper approach is solely based on accuracy to select relevant

Table6. Number of Generated Rules

Dataset	$\mathcal{CN}2$	FSS			
		(wf,$\mathcal{CN}2$)	(wb,$\mathcal{CN}2$)	(f,$\mathcal{ID}3$)	(f,$\mathcal{CI}$)
bupa	34	33	33	34	41
pima	44	56	56	44	54
breast cancer	14	21	15	15	15
hungaria	22	34	27	27	30
crx	35	15	47	37	36
letter	242	313	313	242	226
hepatiti	15	21	14	14	20
anneal	36	49	48	85	90
sonar	28	22	26	23	18
genetics	64	110	56	64	83
dna	130	105	N/A	143	153
Average	60.36	70.82	63.50	66.18	69.64

Table7. Specialization (S) and Generalization (G) Cases

Dataset	FSS(wf,$\mathcal{CN}2$)		FSS(wb,$\mathcal{CN}2$)		FSS(f,$\mathcal{ID}3$)		FSS(f,$\mathcal{CI}$)	
	S	G	S	G	S	G	S	G
bupa							$\bigtriangledown$	$\bigtriangledown$
pima	$\surd$		$\surd$				$\surd$	
breast cancer	$\surd$							
hungaria	$\surd$		$\surd$		$\surd$		$\surd$	
crx		$\surd$	$\surd$					
letter	$\surd$		$\surd$					
hepatitis	$\surd$						$\surd$	
anneal	$\surd$		$\surd$		$\bigtriangledown$	$\bigtriangledown$	$\bigtriangledown$	$\bigtriangledown$
sonar		$\surd$				$\surd$		$\surd$
genetics	$\surd$			$\surd$			$\surd$	
dna		$\surd$	N/A	N/A	$\surd$		$\surd$	

features, there is a tendency for overfitting the data. This shows in the number of cases where specialization holds for both wrappers.

5 Conclusions

The practical objective of Feature Subset Selection is to remove extraneous features, not necessarily to select the optimal subset, leaving the dataset reduced to a manageable dimension. In this work we focused in two approaches for FSS, namely filter and wrapper approaches using the $\mathcal{CN}2$ inducer for evaluation. As filters we considered the $\mathcal{MLC}$++ $\mathcal{ID}3$ and the MineSet^{TM} Column Importance mining tool and as wrapper the $\mathcal{MLC}$++ FSS wrapper. Several experiments on eleven datasets of different dimensionality were discused. The experiments were carried out in such a way that more features than necessary, rather than fewer, may had been considered. The reason of this decision is that inducers can eventually cope with extra features but cannot make up for features that have been discarded. Results obtained show that, as expected, wrappers using forward and backward selection provide a better feature subset for accuracy estimation than both filters. However, for large datasets the wrapper method is very slow. For Data Mining, a special constraint for feature selection is that the size of a dataset

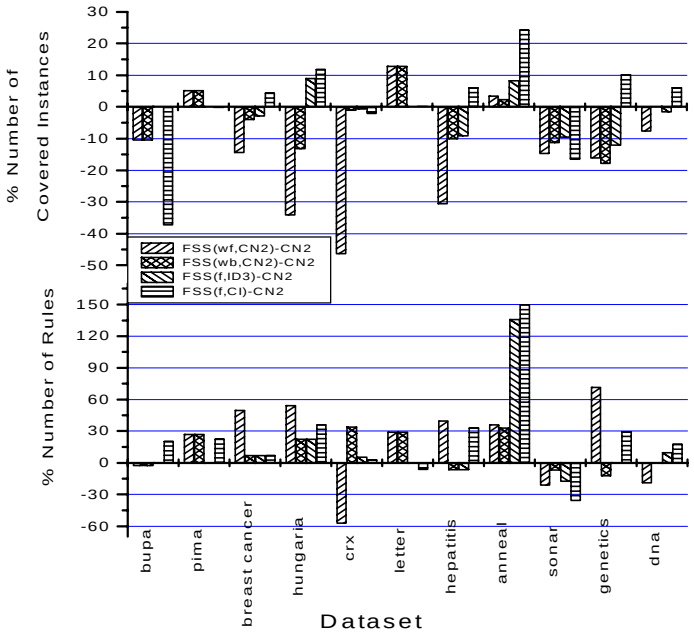

Figure2. Proportions of Number of Covered Instances and Number of Rules

is normally very large, both vertically and horizontally. For this sort of data, the running time would make the wrapper approach infeasible.

It is worth noticing that both filters are very fast and they did not degrade significantly the accuracy performance when compared with the standard inducer. For instance, $\mathcal{ID}3$ filter did not hurt performance except in one of the eleven datasets analyzed here. As it is expected that FSS should be a fast preprocessing task then the filter approach seems to be more appropriate than the wrapper approach. At least, filters should be considered prior to wrappers for FSS when dealing with many features. Most studies in supervised Machine Learning discuss accuracy on an unseen test set as the performance component, but in our experiments besides the goodness of feature subsets we also considered the general syntactic form of the rules induced in each case. Indeed, for Data Mining is also important to pay careful attention to the task of gaining the user/expert's acceptance and validation of the extracted knowledge.

We are currently working on the field of knowledge *interestingness* which also takes into account *comprehensibility* of the knowledge produced. This is being carried out considering not only the widely-used criteria of rule coverage, completeness and confidence factors but also considering new additional factors influencing rule interestingness (Freitas, 1998).

Acknowledgments: We would like to thank Jaqueline Brigladori Pugliesi for helpful comments on the draft of this paper and Ronaldo Prati for extracting $\mathcal{ID}3$ features.

References

Baranauskas, J. A. & Monard, M. C. (1998). Experimental feature selection using the wrapper approach. In *Proceedings of the International Conference on Data Mining*, pages 161–170, Rio de Janeiro, RJ.

Baranauskas, J. A., Monard, M. C., & Horst, P. S. (1999). Evaluation of feature selection by wrapping around the $\mathcal{CN}2$ inducer. *Encontro Nacional de Inteligência Artificial (ENIA)*. Accepted.

Blake, C., Keogh, E., & Merz, C. J. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html.

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

Clark, P. & 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. & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, 3(4):261–283.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R., editors (1996). *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence, Menlo Park, CA.

Freitas, A. A. (1998). A multi-criteria approach for the evaluation of rule interestingness. In *Proceedings of the International Conference on Data Mining*, pages 7–20, Rio de Janeiro, RJ.

KDD (1995). *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, Menlo Park, CA. American Association for Artificial Intelligence.

Kohavi, R. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324.

Kohavi, R. & Sommerfield, D. (1995). Feature subset selection using the wrapper model: Overfitting and dynamic search space topology. In (KDD, 1995), pages 192–197.

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

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

Langley, P. (1996). *Elements of Machine Learning*. Morgan Kaufmann Publishers, Inc.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1:81–106. Reprinted in Shavlik and Dieterich (eds.) Readings in Machine Learning.

Rathjens, D. (1996). *MineSet™ User's Guide*. Silicon Graphics, Inc.

Taylor, C., Mitchie, D., & Spiegelhater, D. (1994). *Machine Learning, Neural and Statistical Classification*. Paramount Publishing International.