A Simple Evaluation Model for Feature Subset Selection Algorithms

Huei Diana Lee, Maria Carolina Monard, Richardson Floriani Voltolini, Ronaldo Cristiano Prati, Wu Feng Chung

Laboratory of Computational Intelligence – LABIC,
Department of Computer Science,
ICMC – USP – São Carlos,
P.O. Box 668,
13560-970 – São Carlos, SP, Brazil

Laboratory of Bioinformatics – LABI,
West Paraná State University – UNIOESTE,
Itaipu Technological Park – PTI,
P.O. Box 1511,
85856-970 – Foz do Iguaçu, PR, Brazil
huei@unioeste.br, {mcmonard,rfv,prati}@icmc.usp.br, wufc@unioeste.br

Abstract

The aim of Feature Subset Selection – FSS – algorithms is to select a subset of features from the original set of features that describes a data set according to some importance criterion. To accomplish this task, FSS removes irrelevant and/or redundant features, as they may decrease data quality and reduce several of the desired properties of classifiers induced by supervised learning algorithms. As learning the best subset of features is an NP-hard problem, FSS algorithms generally use heuristics to select subsets. Therefore, it is important to empirically evaluate the performance of these algorithms. However, this evaluation needs to be multicriteria, *i.e.*, it should take into account several properties. This work describes a simple model we have proposed to evaluate FSS algorithms which considers two properties, namely the predictive performance of the classifier induced using the subset of features selected by different FSS algorithms, as well as the reduction in the number of features. Another multicriteria performance evaluation model based on rankings, which makes it possible to consider any number of properties is also presented. The models are illustrated by their application to four well known FSS algorithms and two versions of a new FSS algorithm we have developed.

Key Words: Feature Selection, Machine Learning, Multicriteria Evaluation.

1 Introduction

Supervised learning algorithms take as input a training set of N classified instances $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ for some unknown function $y = f(\mathbf{x})$, where the $\mathbf{x}_i$ values are typically vectors of the form $(x_{i1}, x_{i2}, ..., x_{iM})$, and x_{ij} denotes the value of the j-th feature (or attribute) X_j of $\mathbf{x}_i$. For classification purposes, the y values are drawn from a discrete set of N_{Cl} classes, *i.e.*, $y \in \{C_1, C_2, ..., C_{N_{Cl}}\}$. From that training set, a learning algorithm induces a *classifier*, which is a hypothesis (model) $\mathbf{h}$ about the true unknown function f. Given new $\mathbf{x}$ values, the classifier predicts the corresponding y values. Although in theory a greater number M of features should provide a greater discriminating power, this may not happen in the presence of irrelevant and/or redundant features, since such features frequently confuse the learning system. Feature Subset Selection — FSS — algorithms can be used as a preprocessing step aiming to remove features such that the prediction performance of the classifier $\mathbf{h}$ improves or is maintained inside reasonable limits considering the reduction in the number of features. Furthermore, by removing redundant and/or irrelevant features, FSS may provide a better understanding of the underlying unknown process, *i.e.* $y = f(\mathbf{x})$, which generates the data. A wide variety of FSS algorithms have been proposed and they can be evaluated in various ways. In fact, the FSS evaluation issue is a complex and multidimensional task.

In this work we describe a simple model we have proposed that takes into account two aspects, namely the predictive performance of the classifier induced using the subset of features selected by different FSS algorithms, as well as the reduction in the number of features. The proposed model has a simple graphical representation, and is illustrated by its application to four well known FSS algorithms and two versions of a new FSS algorithm we have developed. This work also presents a multicriteria performance evaluation model based on rankings, enabling us to consider any number of criteria in the evaluation.

This work is organized as follows: Section 2 briefly reviews the FSS problem. Section 3 presents the FSS algorithms based on Fractal Dimension we have proposed. Section 4 presents the proposed graphical evaluation model. Section 5 reports the results with six FSS algorithms in 11 data sets from UCI using the graphical evaluation model as well as the ones obtained using a model based on rankings. Section 6 presents conclusions and future work.

2 The Feature Subset Selection Problem

The learning task can be broadly divided into two sub-tasks: 1) to decide which attributes should be considered to describe the concept and 2) to decide how to combine these attributes. Thus, the selection of important attributes and the elimination of the irrelevant and/or redundant ones constitute an important problem in machine learning since, in practice, most learning algorithms are confused by the presence of irrelevant and/or redundant attributes.

The aim of feature subset selection is to extract as much information as possible from a given data set by keeping the smallest number of features that describe the data set as well as, or even better, than the original set of features do. This is achieved by removing irrelevant and/or redundant features according to some importance criterion. The FSS goal can be formalized as follows [20]: let $X' \subset X$ be a subset of features from X and $f'(\mathbf{x}')$ be the values associated with vectors corresponding to X'. The aim of FSS is to select a minimum feature subset X' such that $\mathbf{P}(C|y = f'(\mathbf{x}')) \approx \mathbf{P}(C|y = f(\mathbf{x}))$, where $\mathbf{P}(C|y = f'(\mathbf{x}')))$ and $\mathbf{P}(C|y = f(\mathbf{x}))$ are the probability distributions of the N_{C_l} possible classes given feature values of X' and X, respectively. This minimum subset X' is named the *optimal* subset [9]. Some advantages associated with FSS in supervised learning are related to reducing the potential hypothesis space; improving data quality, thus increasing the efficiency of the learning algorithm; improving predictive accuracy, and enhancing the comprehensibility of the induced classifier [20, 2].

There are two main FSS approaches that are external to the learning algorithm: the filter and the wrapper approaches. The main difference between them is related to the interaction of the FSS algorithm with the learning system [7]. On the one hand, FSS is performed as a separate process in the filter approach, which occurs before the application of the learning algorithm itself. The basic idea is to filter features before the induction takes place based on general characteristics from the data set, in order to select some features and discard others. Thus, filter methods are independent from the learning algorithm that simply takes as input the filtered data set. On the other hand, the wrapper approach uses the induction algorithm itself as a black box to evaluate candidate feature subsets, repeating the process on each feature subset until a stopping criterion is met. In general, wrappers are computationally more expensive than filters.

Regarding to feature evaluation, FSS algorithms can perform it in two main ways: individual evaluation of each feature and subset evaluation. Individual evaluation is computationally less expensive, as this approach assesses individual features and assigns them scores according to their degree of importance to the class. Nevertheless, this approach is incapable of detecting redundant features because these features are likely to have similar scores. The subset evaluation approach can handle both feature relevance and feature redundancy. However, unlike individual evaluation, in this approach evaluation measures are defined against a subset of features, thus exhibiting a high computational cost. Regardless of the approach (filter or wrapper) or the used evaluation method (individual or subset evaluation), FSS algorithms need to know what the meaning of a good attribute is, *i.e.*, to answer the following question: *The feature is important related to what?* Many importance measures have been proposed in the literature, and they can be broadly divided into five categories: dependency, consistency, information, distance and classifier accuracy rate [13, 4].

Most FSS methods for supervised learning consider as importance criterion feature relevance to determine the class attribute. However, it has been shown that feature relevance alone is insufficient for efficient FSS. Therefore, it is also necessary to explicitly treat feature redundancy [20]. To this end, we have proposed a FSS algorithm that treats the problem of redundancy using as importance criterion the Fractal Dimension of the data set, described next.

3 Fractal Dimension-Based Filter

Many objects that have a fractal behavior can be found in nature, such as clouds, leaves, flowers, topographies, mountains' chains, and others. Moreover, real world data sets frequently behave like statistically self-similar fractals. Therefore, it is natural the idea of applying concepts from fractal theory to support the analysis of such data sets. The use of the concept of Fractal Dimension — FD — is associated with the existence of redundancy in the data sets and with the possibility of these data sets to be well approximated by smaller dimensions. The main idea is to use the FD of the data set, which is relatively not affected by redundant attributes, to determine how many and which are the non redundant attributes according to the FD criterion. There are many ways of calculating the FD measure. For statistically self-similar fractals, as real world data sets, the FD can be obtained by way of the Correlation Fractal Dimension D_2, which can be calculated using the Box-Count Plot method [5]. This method consists in embedding the data set with a point set in an M-dimensional space, with M-grid cells of side r. Afterwards, focusing on the i-th cell, the number of points that fall into each cell ($C_{r,i}$) is counted, and the value $S_2(r) = \sum_i {C_{r,i}}^2$ is computed. The Correlation Dimension D_2 is defined by $D_2 = \frac{\partial log(\sum_i S_2(r))}{\partial log(r)}$, $r \in [r_{min}, r_{max}]$. In theory, exactly self-similar fractals are infinite. In practice, real world data sets which present a finite number of points are considered statistically self-similar fractals for a determined interval of scales $r \in [r_{min}, r_{max}]$, if they fulfill a construction rule in this interval. Therefore, the intrinsic dimension of a specific data set may be measured by the slope of the linear part of the resulting graph obtained from plotting $S_2(r)$ for different values of r [18]. In this work, the correlation dimension D_2 will be simply denoted as fractal dimension FD.

Our algorithm, called Fractal Dimension-Based Filter —FDimBF — is based on a recently proposed framework for FSS that decouples the selection of important attributes into two separate process: the analysis of relevance that is carried out as the first step of the process, and the analysis of redundancy carried out as the second step [20]. Figure 1 shows the general framework of FDimBF, which is described in details in [10].

Two different versions of the FDimBF algorithm were implemented to accomplish the relevance analysis. The first one performs this analysis using an information based measure, specifically the *information gain ratio* — FDimBF(1). The second one — FDimBF(2) — uses a distance based measure which ranks features using the *Manhattan distance*. For both FDimBF(1) and FDimBF(2) the redundancy analysis is carried out by the Fractal Dimension Reduction — FDR — algorithm [18]. The main idea of FDR is to discard features that have little influence over the fractal dimension of the data set, since the FD is relatively not affected by redundant features.

4 Evaluation Model for Feature Subset Selection Algorithms

When we apply FSS algorithms we would like to reduce the number of features needed for learning. This means that at least we have to consider two properties (or aspects) simultaneously: a reduction in the number of features *versus* the accuracy of the induced classifier using the subset of selected features. To this end, we have proposed an evaluation model for FSS algorithms'

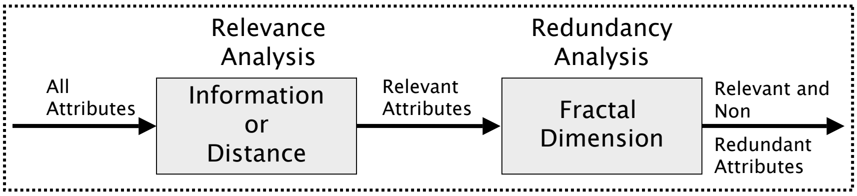

Figure 1: General framework of FDimBF

performance which considers these two main issues of the FSS problem: 1) predictive accuracy of the induced classifier using all features from the original data set and the ones induced using the subset of features selected by each FSS algorithm, and 2) size of the subset of selected features in relation to the original data set. In this framework, (illustrated in Figure 2) where EFS stands for the classifier's error without FSS and EMC represents the error of the data set's majority class whenever this error is less than 50%, otherwise EMC is set to 50%, we place the FSS algorithms´ performance into five categories: excellent (△△△), very good (△△), good (△), poor (◇) and very poor (▽). Using this model, which enables showing a neat visualization of results to the user, both versions of FDimBF were empirically evaluated and compared to four representative filter based FSS algorithms. The experiments are reported in the next section.

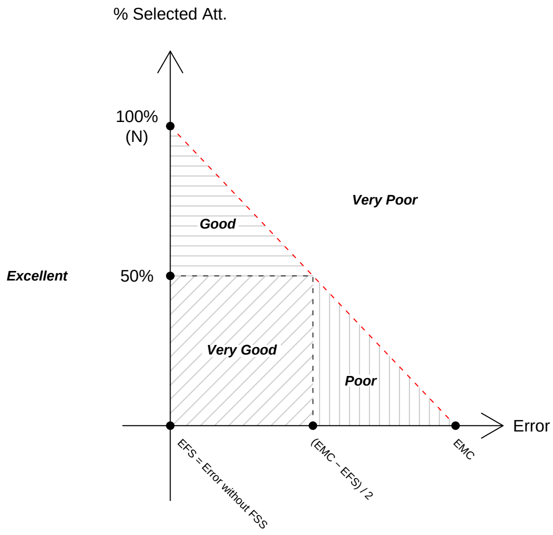

Figure 2: Evaluation framework for FSS algorithms

5 Experiments and Discussion

Table 1 describes the characteristics of the six FSS algorithms, where lines 2 and 3 show, respectively, the algorithms that perform individual evaluation and/or subset evaluation of features. Lines 4 to 7 show the importance measure considered during the attributes' evaluation process. In this work, 11 data sets from UCI [14] were selected for the empirical study. The main characteristics of the data sets used in the experiments are described in the second column of Table 2, which shows the number of examples (#Ex.), attributes (#Att.) and the EMC of each data set. Except for data sets Satimage, Segment, Vehicle and Waveform which have respectively 7, 7, 4 and 3 class labels, the other data sets have only two classes.

Two of the chosen FSS algorithms perform individual feature evaluation — ReliefF [8] and FCBF (*Fast Correlation-Based Filter*) [20]. The other two select important features using subset evaluation — CFS (*Correlation-Based Feature Selection*) [6] and CBF (*Consistency-Based Filter*) [12]. ReliefF searches for nearest neighbors of examples with different class labels, and features are weighed according to how well they differentiate these examples. Similar to FDimBF, FCBF selects features in two steps, using an information measure in the second step to remove redundant features. CFS evaluates the goodness of a subset of features by considering the individual predictive ability of each feature and the degree of correlation among them. Subset evaluation is also performed by CBF according to their inconsistency related to the class, searching for subsets that separate data into clusters such that each cluster has a large number of examples with the same class value. All algorithms were executed with default parameters values. Except for the FDimBF algorithm, they are available at Weka's environment [19]. Experiments were conducted following the four steps shown in Figure 3.

	ReliefF	CFS	FCBF	CBF	FDimBF(1)	FDimBF(2)
Individual	×		×		×	×
Subset		×	×	×	×	×
Information		×	×		×	
Distance	×					×
Dependency					×	×
Consistency				×		

Table 1: Characteristics of FSS algorithms

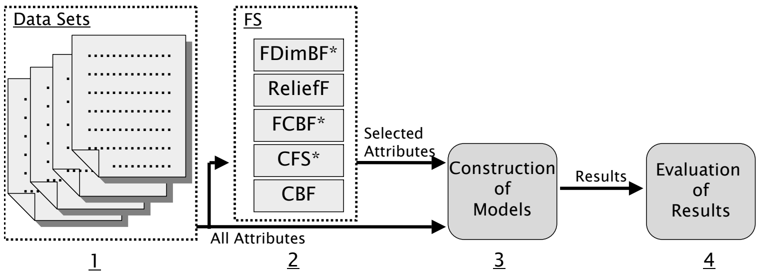

Figure 3: Experimental setup

In step 1, data sets were cleaned and prepared for the next step. The cleaning task consisted of removing unknown values conditioned to the major concentration of them in examples and/or attributes. From the total of 11 data sets, only two of them were data cleaned: Breast Cancer, resulting in 683 examples from the original of 699 examples and the same number of attributes, and Hungarian for which 3 attributes and 33 examples were removed, totalizing 261 examples and 10 attributes — Table 2. In step 2, subsets of attributes were selected from the original set of attributes considering both versions of the proposed algorithm FDimBF and the other four FSS algorithms — Table 1. Algorithms marked with $*$ in Figure 3 deal with both problems: relevance and redundancy of attributes. In step 3, classifiers were constructed using as input the subset of selected attributes in the previous step. To this end, the $\mathcal{C}4.5$ [17] learning algorithm, executed with its default values, was used. As for real world data prior knowledge about important features is not generally available, predictive accuracy of the constructed classifiers is commonly used as an indirect measure to evaluate the quality of the selected features. Thus, in final step 4, results were evaluated by estimating the error rate of the classifiers induced by $\mathcal{C}4.5$ using 10 fold cross-validation. To support this task, we used the SNIFFER environment for managing experiments, which is part of the DISCOVER project[1] [16, 1]. The algorithms' performance for the 11 data sets according to the framework proposed in Section 4 are presented in Table 2.

As can be observed in Table 2, according to the proposed evaluation model — Figure 2 — both versions of FDimBF were the ones that obtained the total greatest number of excellent and very good performances, 9 out of 11. Algorithms CFS and CBF obtained respectively 8 and 7 excellent or very good performances, followed by 4 from FCBF. Poor performances occurred uniformly among all considered FSS algorithms, and only 3 of them presented one very poor performance each: ReliefF and both versions of FDimBF. Regarding the number of selected important features, ReliefF chose all features in 8 of the 11 data sets. In fact, only both versions of FDimBF always promoted a reduction on the number of selected features for all data sets as shown in the last line of Table 2.

Therefore, under the proposed evaluation model, from the 66 considered cases (11 data sets × 6 FSS algorithms), 16 were excellent, 21 very good, 7 good, 4 poor, 3 very poor and 15 presented as subset of selected features the original ones. Thus, 66.67% of the cases were considered excellent, very good or good; 22.73% of the selected subsets were equal to the original set of features, and only 10.61% showed a poor or very poor performance. Hence, the majority of the FSS algorithms contributed to improving both the reduction in the quantity of features and the accuracy of the constructed models under the pro-

[1]A computational environment being developed at the Laboratory of Computational Intelligence — LABIC.

	#Ex.	#Att.	EMC	ReliefF	CFS	FCBF	CBF	FDimBF(1)	FDimBF(2)
#1 Breast Cancer	683	9	34.48	—	—	—	△△△	△△△	△△△
#2 Bupa	345	6	42.03	▽	◇	◇	◇	▽	△
#3 German	1000	24	30.00	—	△△	△△△	△△△	△△△	△△△
#4 Hungarian	261	10	36.05	—	△△△	△△△	△△△	△△	△△△
#5 Ionosphere	34	351	35.90	△	△△	△	△△	△△	△△
#6 Pima	769	8	34.98	—	△△	—	—	△△	▽
#7 Satimage	4435	36	75.80	—	△△△	—	△△△	△△	△△
#8 Segment	2310	19	85.70	△	△△	△	△△	△△	△△
#9 Sonar	208	60	46.60	—	△△△	△△	△△	◇	△△
#10 Vehicle	846	18	74.20	—	△	—	—	△△	△△
#11 Waveform	5000	21	66.10	—	△△△	△△△	△	△△	△△
Excellent (△△△)				0	4	3	4	2	3
Very Good (△△)				0	4	1	3	7	6
Good (△)				2	1	2	1	0	1
Poor (◇)				0	1	1	1	1	0
Very Poor (▽)				1	0	0	0	1	1
All Selected Attributes (—)				8	1	4	2	0	0

Table 2: Data sets and performance of algorithms according to the percentage of selected attributes *versus* model's error

posed framework to evaluate the performance of FSS algorithms.

Although the proposed evaluation model gives a global idea about the behavior of FSS algorithms regarding its two main aspects, *i.e.* the classifiers' predictive performance allied to a smaller number of features, it would be interesting to suggest an order of preference among them. One way of doing this is to consider, for all FSS algorithms we would like to compare, the ranking defined by each individual property (or aspect) we want to evaluate, and try to combine the individual rankings into a final one. The main advantage of using rankings is the possibility of considering different measures in different scales. By using rankings, the numerical value of a measure is not taken into account, but only the ordering they define [15, 3]. Thus, in the evaluation problem of FSS algorithms it is possible to directly compare the predictive performance and the number of selected features, as well as any other performance measures.

The main idea of using rankings for multicriteria evaluation can be described as follows: first, a rank of FSS algorithms is constructed separately for each aspect, the best performing algorithm obtaining rank 1, the second best rank 2, and so forth. In case of ties, average ranks for all tied algorithms are assigned. Afterwards, the following very simple method can be used to combine individual rankings: for each FSS algorithm, its individual ranking for each aspect is averaged, and the average ranking is used to order the FSS algorithms. This final ordering can be interpreted as a consensus of the performance of FSS algorithms on two (or any number) of aspects we would like to consider.

In our case, in order to compare both aspects, predictive classifier performance and the number of features selected for each algorithm in each data set, we start by calculating the algorithms' average error rate using the selected features as well as the number of selected features. In the second phase, we proceed to construct a ranking on the basis of this performance information, and finally we calculate the average ranking of each algorithm in all data sets. In other words, let E_j^i be the estimated error rate of algorithm j using the features selected from data set i, and A_j^i be the number of features selected by algorithm j from data set i. In the first case, we order the E_j^i values for each data set i, and assign the corresponding ranking position e_j^i. In case of a draw, *i.e.* $E_j^i = E_k^i$, we assign the averaged ranking position for the tied algorithms j and k. Afterwards, we calculate the average ranking of each algorithm as $\bar{e_j} = \frac{\sum_i e_j^i}{n}$, where n is the number of data sets, 11 in our case. Similarly for A_j^i, we calculate the a_j^i rankings and the average ranking $\bar{a_j} = \frac{\sum_i a_j^i}{n}$. Tables 3 and 4 show the results. Due to lack of space, we only show the average values $\bar{e_j}$ and $\bar{a_j}$. Individual measures E_j^i and A_j^i for each algorithm and data set can be found in [11].

In these tables, the first column identifies the algorithm, and the following 11 columns the rankings for each data set i. For example, column e_j^1 in Table 3 refers to data set #1 Breast Cancer — Table 2. For this data set, there are two draws in the first and second positions for both versions of FDimBF. Thus, the score of both algorithms is given by $\frac{1+2}{2} = 1.5$. Moreover, there are three draws in positions 4 to 6, and the score of these algorithms is given by $\frac{4+5+6}{3} = 5$. Finally, the last two columns of Table 3 show, respectively,

Algorithm$_j$	e_j^1	e_j^2	e_j^3	e_j^4	e_j^5	e_j^6	e_j^7	e_j^8	e_j^9	e_j^{10}	e_j^{11}	$\bar{e_j}$	$R(\bar{e_j})$
ReliefF	5	2	5	4.5	2.5	2	2.5	1.5	2	2	3	2.9	1.5
CFS	5	4	6	1	1	4	4	4	1	4	1	3.2	4
CBF	5	4	3.5	3	2.5	2	2.5	1.5	4	2	2	2.9	1.5
CBF	3	4	3.5	4.5	4	2	1	3	3	2	4	3.1	3
FDimBF(1)	1.5	6	1	6	6	5	5.5	5.5	6	5.5	5.5	4.9	6
FDimBF(2)	1.5	1	2	2	5	6	5.5	5.5	5	5.5	5.5	4.0	5

Table 3: Ranks for error rate on each data set

Algorithm$_j$	a_j^1	a_j^2	a_j^3	a_j^4	a_j^5	a_j^6	a_j^7	a_j^8	a_j^9	a_j^{10}	a_j^{11}	$\bar{a_j}$	$R(\bar{a_j})$
ReliefF	5	5.5	6	6	5.5	5	5.5	5.5	6	5	6	5.55	6
CFS	5	2	1	1.5	4	1.5	4	3	4	3	4	3	3
FCBF	5	2	4.5	5	5.5	5	5.5	5.5	5	5	5	4.82	5
CBF	3	2	4.5	4	3	5	3	4	3	5	3	3.59	4
FDimBF(1)	1.5	4	2.5	1.5	1	1.5	1.5	1.5	1	1.5	1.5	1.73	1
FDimBF(2)	1.5	5.5	2.5	3	2	3	1.5	1.5	2	1.5	1.5	2.32	2

Table 4: Ranks for percentage of selected features on each data set

the $\bar{e_j}$ average ranking values and the new ranking $R(\bar{e_j})$ correspondent to the $\bar{e_j}$ value. Table 4 shows similar information considering the number of selected features.

In a similar way, we can combine these results to obtain the evaluation of both aspects, as shown in Table 5. According to this result, algorithm CFS is the winner, FDimDF(2) the second best, FDimDF(1) the third best and so forth. It can be observed that this results are in concordance with the ones obtained by our proposed model — Table 2. However, rankings provide a better discrimination among all FSS algorithms.

Algorithm$_j$	$\bar{e_j}$	$\bar{a_j}$	$\frac{\bar{e_j}+\bar{a_j}}{2}$	$R(\frac{\bar{e_j}+\bar{a_j}}{2})$
ReliefF	2.9	5.55	4.23	6
CFS	3.2	3	3.1	1
FCBF	2.9	4.82	3.86	5
CBF	3.1	3.59	3.35	4
FDimBF(1)	4.9	1.43	3.17	3
FDimBF(2)	4	2.32	3.16	2

Table 5: Average ranks for error rate and selected features

As stated before, the main advantage of rankings lies in the possibility of considering various properties measured in different scales. Although not all of the properties that are used for the evaluation of classifiers are quantifiable, such as the interpretability of symbolic classifiers, it is still possible to measure the so-called syntactic complexity of the induced model which is related to the total number of rules (or branches in a decision tree), and the mean number of conditions in the rules. In what follows, we consider the overall syntactic complexity of the model induced by $\mathcal{C}4.5$ using the subset of features selected by each FSS algorithm with respect to the one induced using all features. The results for syntactic complexity are shown in Table 6. In this table, each line shows simultaneously both rankings for branches b_j^i and nodes d_j^i of the induced models. As before, j refers to the FSS algorithm and i to the data set. Complete results can be found in [11]. In the same way, column 14 shows the average for both branches and nodes. The next column shows de mean of these two previous values. Finally, column 16 shows the final ranking $R(\bar{s_j})$ for the syntactic complexity.

All results considering these three properties can be combined as shown in Table 7. As can be seen in this table, FDimBF(1) and CBF gain one position, being ranked in second and third places, respectively. On the other hand, FDimBF(2) looses two positions, being ranked in the fourth position. Taking into account all these results with the 11 data sets, it can be observed that CFS and FDimBF were always ranked within the top 50% of the best FSS algorithms.

6 Conclusion

In this work we proposed a simple multicriteria model to assist the user in selecting the more appropriate FSS algorithm for a given data set. This model has an easy graphical interpretation, and takes into account not only the performance of the classifier induced with the subset of selected features, but also the reduction in the number of selected features. The model was applied to six different FSS algorithms on 11 datasets. To

Algorithm$_j$	Measure	b_j^1 d_j^1	b_j^2 d_j^2	b_j^3 d_j^3	b_j^4 d_j^4	b_j^5 d_j^5	b_j^6 d_j^6	b_j^7 d_j^7	b_j^8 d_j^8	b_j^9 d_j^9	b_j^{10} d_j^{10}	b_j^{11} d_j^{11}	$\bar{b_j}$ $\bar{d_j}$	$\bar{s_j}$	$R(\bar{s_j})$
ReliefF	Branches	3	5	6	6	4	6	2	4	4	2	6	4.36	4.09	5.5
	Nodes	2	5	5	1	6	2	5	6	2	6	2	3.82		
CFS	Branches	6	1.5	1	2	1	1	3	1	5.5	4	4	2.73	2.95	1
	Nodes	1	3	2	3	5	3	6	5	1	5	1	3.18		
FCBF	Branches	5	3	3	4	3	5	1	3	5.5	3	5	3.68	3.36	4
	Nodes	4	1.5	4	5	2	4	4	2	3	1	3	3.05		
CBF	Branches	4	1.5	4	5	2	4	4	2	3	1	3	3.05	3.32	3
	Nodes	5	2	3	4	3	5	1	3	5.5	3	5	3.59		
FDimBF(1)	Branches	1	4	2	3	5	3	6	5	1	5	1	3.27	3	2
	Nodes	6	1.5	1	2	1	1	3	1	5.5	4	4	2.73		
FDimBF(2)	Branches	2	6	5	1	6	2	5	6	2	6	2	3.91	4.09	5.5
	Nodes	3	4	6	6	4	6	2	4	4	2	6	4.27		

Table 6: Ranks for syntactic complexity on each data set

Algorithm$_j$	$\bar{e_j}$	$\bar{a_j}$	$\bar{s_j}$	$\frac{\bar{e_j}+\bar{a_j}+\bar{s_j}}{3}$	$R(\frac{\bar{e_j}+\bar{a_j}+\bar{s_j}}{3})$
ReliefF	2.9	5.55	4.09	4.18	6
CFS	3.2	3	2.95	3.05	1
FCBF	2.9	4.82	3.36	3.69	5
CBF	3.1	3.59	3.32	3.34	3
FDimBF(1)	4.9	1.43	3	3.11	2
FDimBF(2)	4	2.32	4.09	3.47	4

Table 7: Average ranks for error rate, selected features and syntactic complexity

further discriminate the global results supplied by our model, we proposed the use of rankings and illustrate this idea taking into account the same performance measures than the ones used in the graphical model, and show how to further improve the final ranking estimation considering as a third aspect the syntactic complexity of the induced classifiers.

As future work, we plan to investigate other multicriteria methods to analyze the performance of algorithms using rankings, as well as evaluating and comparing different ranking methods for this task.

Acknowledgments: Richardson Voltolini is supported by an MSc scholarship (process number 04/04885-8) from FAPESP (Brazil). The remaining authors would like to thank the support of the Institute of Advanced Technologies and Innovation (Brazil) — ITAI — and the Brazilian Itaipu Technological Park Foundation — FPTI.

References

[1] G. E. A. P. A. Batista and M. C. Monard. Arquitecture and project description of the DISCOVER LEARNING ENVIRONMENT — DLE. Technical Report 187, ICMC-USP, 2003. (In Portuguese) `ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/RT_187.pdf`.

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

[3] Pavel Brazdil, C. Soares, and J. Pinto da Costa. Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. *Machine Learning*, 50(3):251–277, 2003.

[4] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(3):131–156, 1997.

[5] C. Faloutsos and I. Kamel. Beyond uniformity and independence: Analysis of r-trees using the concept of fractal dimension. In *Proc.of the 13th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 4–13, Minneapolis, MN, 1994.

[6] Mark A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proc. of the 17th Int. Conf. on Machine Learning*, pages 359–366, San Francisco, CA, 2000. Morgan Kaufmann.

[7] G. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In Morgan Kaufmann, editor, *Proc. of the 11th Int. Conf. on Machine Learning*, pages 167–173, San Francisco, CA, 1994.

[8] K. Kira and L. Rendell. A pratical approach to feature selection. In *Proc. of the 9th Int. Conf. on Machine Learning*, pages 249–256, Aberdeen, Scotland, 1992. Morgan Kaufmann.

[9] D. Koller and M. Sahami. Toward optimal feature selection. In *Proc. of the 13th Int. Conf. on Machine Learning*, pages 284–292, Bari, Italy, 1996.

[10] Huei Diana Lee. *Selection of important features for knowledge extraction from data bases*. PhD thesis, ICMC — USP, São Carlos, december 2005. (In Portuguese) `http://www.teses.usp.br/teses/disponiveis/55/55134/tde-15032002-113112/`.

[11] Huei Diana Lee, Maria Carolina Monard, Richardson Floriani Voltolini, and Feng Chung Wu. Experimental evaluation of the FDimBF algorithm based on fractal dimension with other features selection algorithms. Technical Report 264, ICMC-USP, 2005. (In Portuguese) `http://www.icmc.usp.br/~biblio/download/RT_264.pdf`.

[12] H. Liu and R. Setiono. A probabilistic approach to feature selection – a filter solution. In *Proc.of the 13th Int. Conf. on Machine Learning*, pages 319–327, Bari, Italy, 1996.

[13] Huan Liu and Hiroshi Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, Massachusetts, 1998.

[14] C. J. Merz and P. M. Murphy. UCI repository of machine learning datasets, 1998.

[15] Gholamreza Nakhaeizadeh and Alexander Schnabl. Development of multi-criteria metrics for evaluation of data mining algorithms. In *KDD*, pages 37–42, 1997.

[16] Ronaldo Cristiano Prati. The integration framework of the DISCOVER system, 2003. (In Portuguese) Master Thesis, ICMC-USP, `http://www.teses.usp.br/teses/disponiveis/55/55134/tde-20082003-152116/`.

[17] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993. California.

[18] C. Traina, E. P. M Sousa, and A. J. M. Traina. *New Generation of Data Mining Applications*, volume 1 of *1*, chapter Using Fractals in Data Mining, pages 599–630. Wiley-IEEE Press, New Jersey, NJ, 1 edition, 2005.

[19] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, California, 2000.

[20] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundance. *Journal of Machine Learning Research*, 5:1205–1224, 2004.