Metalearning for Choosing Feature Selection Algorithms in Data Mining: Proposal of a New Framework

Article *in* Expert Systems with Applications · January 2017

DOI: 10.1016/j.eswa.2017.01.013

Antonio Rafael Sabino Parmezan
University of São Paulo

Huei Diana Lee
Universidade Estadual do Oeste do Paraná, Foz do Iguaçu, Brazil

Metalearning for Choosing Feature Selection Algorithms in Data Mining: Proposal of a New Framework

Antonio Rafael Sabino Parmezan[a], Huei Diana Lee[b,c], Feng Chung Wu[b,c]

[a]*Laboratory of Computational Intelligence, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Av. Trabalhador São-carlense, 400, 13566-590 São Carlos, SP, Brazil*

[b]*Laboratory of Bioinformatics, Centro de Engenharias e Ciências Exatas, Universidade Estadual do Oeste do Paraná, Av. Tarquínio Joslin dos Santos, 1300, 85867-900 Foz do Iguaçu, PR, Brazil*

[c]*Coloproctology Service, Faculdade de Ciências Médicas, Universidade Estadual de Campinas, Rua Tessália Vieira de Camargo, 126, Cidade Universitária Zeferino Vaz, 13083-887 Campinas, SP, Brazil*

Abstract

In Data Mining, during the preprocessing step, there is a considerable diversity of candidate algorithms to select important features, according to some criteria. This broad availability of algorithms that perform the Feature Selection task gives rise to the difficulty of choosing, *a priori*, between the algorithms at hand, the most promising one for a particular problem. In this paper, we present the proposal and evaluation of a new architecture for the recommendation of Feature Selection algorithms based on the use of Metalearning. Our framework is very flexible since the user can adapt it to its proper needs. This flexibility is one of the main advantages of our proposal over other approaches in the literature, which involve steps that cannot be adapted to the user's local requirements. Furthermore, it combines several concepts of intelligent systems, including Machine Learning and Data Mining, with topics derived from expert systems, as user and data-driven knowledge, with meta-knowledge. This set of solutions coupled with leading-edge technologies allows our architecture to be integrated into any information system, which impact on the automation of services and in reducing human

Email addresses: `antoniorafaelparmezan@gmail.com` (Antonio Rafael Sabino Parmezan), `hueidianalee@gmail.com` (Huei Diana Lee), `wufengchung@gmail.com` (Feng Chung Wu)

effort during the process. Regarding the Metalearning process, our framework considers several types of properties inherent to the data sets, as well as, Feature Selection algorithms based on many information, distance, dependence and consistency measures. The quality of the methods for Feature Selection was estimated according to a multicriteria performance measure, which guided the ranking process of these algorithms for the construction of data metabases. Proposed by the authors of this work, this multicriteria performance measure combines any three measurements on a single one, creating an interesting and powerful tool to evaluate not only FS algorithms but also to assess any context where it is necessary a combination to maximize a measure or minimize it. The recommendation models, represented by decision trees and induced from the training metabases, allowed us to see in what circumstances a Feature Selection algorithm outperforms the other and what aspects of the data present greater influence in determining the performance of these algorithms. Nevertheless, if the user wishes, any other learning algorithm may be used to induce the recommendation model. This versatility is another strong point of this proposal. Results show that with the characterization of data, through statistical, information and complexity measures, it is possible to reach an accuracy higher than 90%. Besides yielding recommendation models that are interpretable and robust to overfitting, the developed architecture is less computationally expensive than approaches recently proposed in the literature.

Keywords: data quality, algorithm recommendation, symbolic metamodels, machine learning

1. Introduction

Computer systems have played an increasingly important role in various areas of knowledge, contributing to generate and store an increasing amount of data. Usually, these data are organized in computer Databases[1] (DB), which can be used for anything from a simple consultation of a specific information held in the DB to the extraction of knowledge by building models that represent patterns contained in these data. Regarding the latter, the knowledge can be extracted using, for example, Machine Learning (ML) methods

[1] In this paper, the term computer databases will be used without distinction to the terms data, data set or set of examples.

in the process of Data Mining (DM) (Liu and Motoda, 2013; Han et al., 2011).

The DM process can be divided into three main steps. The first step is the preprocessing of data, in which tasks like data reducing and cleaning are performed. For a long time, this step was underestimated with regard of the second step, known as extraction of patterns. However, it is of fundamental importance, since the quality of the data, provided as input for the second step, directly influences the quality of the generated models. In the second step, the objective is to extract patterns contained in the data so that the concept embedded in them can be represented by a model. In this step we can employ, for example, ML algorithms, a subarea of Artificial Intelligence. In the third step, the patterns extracted and represented in the built models are evaluated, validated and consolidated. During this step, we must ensure that the extracted patterns are statistically significant and reliable. It is also important that the extracted knowledge is validated with the previous knowledge of the domain so that possible conflicts are removed. Finally, the gained knowledge can be consolidated and made available to the user.

Several ML methods have been proposed for the construction of these models, focusing on some measures of performance, such as accuracy and understandability. However, factors such as the curse of dimensionality, which refers to the exponential increase in the complexity of a given problem according to the increase of its dimensionality (Zhao et al., 2010), with reference to the number of features[2] used to represent a data set, can hinder the direct application of these methods (Nogueira, 2009; Lee et al., 2006; Alpaydin, 2004).

Thus, preprocessing tasks for the Dimensionality Reduction of these data sets, such as Feature Selection, play an important role within the DM process (Liu and Motoda, 2013).

The first step of DM is regarded as one of the most expensive steps as it may consume nearly 80% of the whole process (Pyle, 1999), and its correct planning and execution are of great importance to ensure that the data is of good quality. A major task in this step is the Feature Selection (FS), which can contribute to not only removing redundant and irrelevant features, but a to better understandability of the generated results (Liu and Motoda, 2008).

In the literature, there is the proposition of various FS algorithms (Liu

[2] The terms features, attributes and variables are used interchangeably in this paper.

and Motoda, 2013, 2008). From the practical point of view, the preference for a particular FS algorithm should be based on two main aspects: the technical knowledge about the FS algorithm, dependent on the computer specialists, and the domain knowledge, in general, dependent on the domain specialists.

However, the large availability of FS algorithms may not be used to its maximum, since the difficulty of choosing, *a priori*, between algorithms with different characteristics, increases according to the particularities of the problem, as no algorithm can be considered the best regardless of the problem at hand.

In addition to the knowledge of the domain and computer specialists, it is also common to base the choice of FS algorithms on massive empirical evaluations (Muñoz and González-Navarro, 2011; Covões, 2010; Lee et al., 2006; Liu et al., 2004; Molina et al., 2002).

As mentioned, the number of features used to represent the set of examples is one of the factors that can directly influence the predictive performance of the constructed model, since various ML algorithms do not work well in the presence of a large number of features (Liu and Motoda, 2008; Lee and Monard, 2003). Thus, it is desirable that FS tasks, which aim to reduce the original set of features, can be performed in order to generate subsets with equal representation potential, or possibly better than the original, so as to minimize the effects of the curse of dimensionality.

However, given the great diversity of algorithms that perform the FS task, another problem arises, the choice of the most adequate FS algorithm for each problem or domain in question.

In this context, the Metalearning, defined as the modeling process in meta level (Brazdil et al., 2009), can assist the user in automatic and systematic indication for the choice of FS algorithms, through the construction of models that associate the performance of the algorithms with the morphological properties of the data sets.

In this work we present the proposal and evaluation of a novel architecture of recommendation of Feature Selection algorithms based on the investigation of the use of Metalearning, considering characteristics intrinsic to the data sets and the Feature Selection algorithms, as aid for the suitable recommendation of theses algorithms.

The rest of this paper is organized as follows: Section 2 introduces concepts on the problem of Feature Selection that are important in the process of Data Mining. In Section 3, we briefly present fundamentals on the Metalearning theme. In Section 4 the related work are discussed and in Section 5, we described the proposed architecture for recommendation of Feature Selection algorithms. In Section 6, we specify the configuration of the experiments, which includes the main algorithms used as well as the considered data sets. Results and discussion are presented in Section 7, while in Section 8, we present conclusions and future work.

2. Feature Selection and Importance Measures

FS is a research topic in constant development and plays an important role in the DM preprocessing step (Witten et al., 2011; Kohavi and John, 1997; Fayyad et al., 1996), because the proper choice of a FS method can enhance the quality of the data provided as input for the algorithms of pattern extraction.

The simplest and most used format for the description of a data set is called the attribute-value format, as shown in Table 1. In this table, each row represents an example or occurrence E_i within the data set, for $i = 1 \ldots N$, and each column represents an attribute (or feature) A_j, for $j = 1 \ldots M$, that has specific values for each example (a_{ij}). Additionally, there may be an attribute of special significance $\mathbb{C}$, called target or class attribute, which represents the concept to be learned and described by the built models using the methods of ML.

Table 1: Attribute-value format

	Attributes				
Examples	A_1	A_2	$\ldots$	A_M	Class ($\mathbb{C}$)
E_1	a_{11}	a_{12}	$\ldots$	a_{1M}	c_1
E_2	a_{21}	a_{22}	$\ldots$	a_{2M}	c_2
E_3	a_{31}	a_{32}	$\ldots$	a_{3M}	c_3
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
E_N	a_{N1}	a_{N2}	$\ldots$	a_{NM}	c_N

Thus, the data set size is measured according to two dimensions: number of examples (N) and number of attributes (M). These attributes can take quantitative or qualitative values.

In FS, the set of features that represents the data set can be identified in two ways. In the first strategy, called ranking, the goal is to determine an ordering of features that originally represent the data set. In the second strategy, a subset of features, preferably smaller than the original set, is selected.

Both approaches, ranking and Feature Subset Selection (FSS), employ some criteria or measure of importance to perform the feature selection (Liu and Motoda, 2013; Parmezan et al., 2011a,b; Lee et al., 2006).

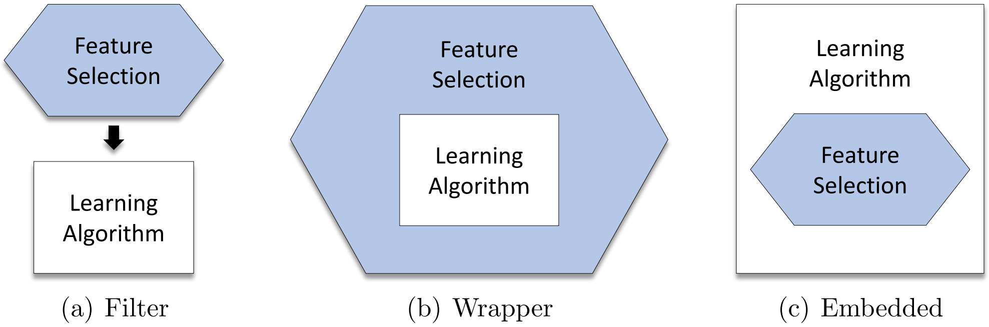
The task to select important features and remove the non-important ones is an essential part of the problems to be addressed in DM, as the extraction of patterns covers two main issues (Lee, 2005):

1. Which features are going to be used to represent the concept mapped by the patterns to be extracted;
2. How these features will be combined.

It is important to note that it is not enough that relevant features are chosen, but that the redundancy problem between them is also treated.

The FS can be formalized as follows (Yu and Liu, 2004): Consider $S' \subset S$ as a subset of features of S, and $f(x)$ and $f'(x')$ the values of the classes associated with the vectors corresponding to S and S', respectively. The objective of FS consists in selecting a minimum subset of features S' such that $P(C|c = f'(x')) \approx P(C|c = f(x))$, where $P(C|c = f'(x'))$ and $P(C|c = f(x))$ are the probability distributions of the N possible classes, given the values of the features of S' and S, respectively. This minimum subset S' is called the optimal subset of features.

In this context, there are three main approaches for FSS, which are outlined in Figure 1.



(a) Filter (b) Wrapper (c) Embedded

Figure 1: Approaches for Feature Subset Selection (modified from Covões (2010))

In the Filter approach (Figure 1(a)), the choice of important features is performed in a preprocessing step using general characteristics of the data set.

This process is independent from the learning algorithm (pattern extraction) and thus prior to the construction of models.

Differently, the other two FS approaches are influenced by the learning algorithm. In this sense, the Wrapper approach (Figure 1(b)), uses a model built to evaluate each subset of features chosen as a possible candidate, using its performance, usually the predictive one. Therefore, this approach is highly expensive and dependent on the algorithm used for such evaluation, which is generally the same employed to induce the final model.

In the Embedded approach (Figure 1(c)), the FS task is performed internally to the learning algorithm. In this case, the learning algorithm receives as input the original set of features and selects only the important features, according to the predefined importance criteria of this algorithm, for inducing the model.

As mentioned, both strategies of FS, ranking and FSS, employ some criteria or measure of importance to perform feature selection. According to Liu and Motoda (2013), the Importance of a Feature can be defined as follows:

Definition 1. *A feature is said to be important if, when removed, the considered measure of importance regarding the rest of the features is deteriorated.*

In Figure 2 we illustrate the hierarchy of importance measures. Two main categories are represented: the ones that are dependent on the learning algorithm, as the predictive performance, and those that are independent, called separability measures.

Information measures determine the information gain from a feature. An example of this kind of measure is entropy. As for the distance ones, also called measures of separability, divergence or discrimination, the difference between conditional probabilities of two classes is calculated. An example of this kind of measure is the Euclidean distance.

Another type of measure, known as correlation or association measure, is also called dependence measure and qualifies the ability to predict a value of a feature from the value of another. The correlation coefficient constitutes an example of this classic dependence measure. Finally, the consistency measures possess different characteristics from other measures, as they are highly dependent on the data set and show a preference for consistent hypotheses that can be defined from the smallest possible number of features. Thus, they seek to find the minimum subset of features that meet the inconsistency proportion accepted, commonly defined by the user.

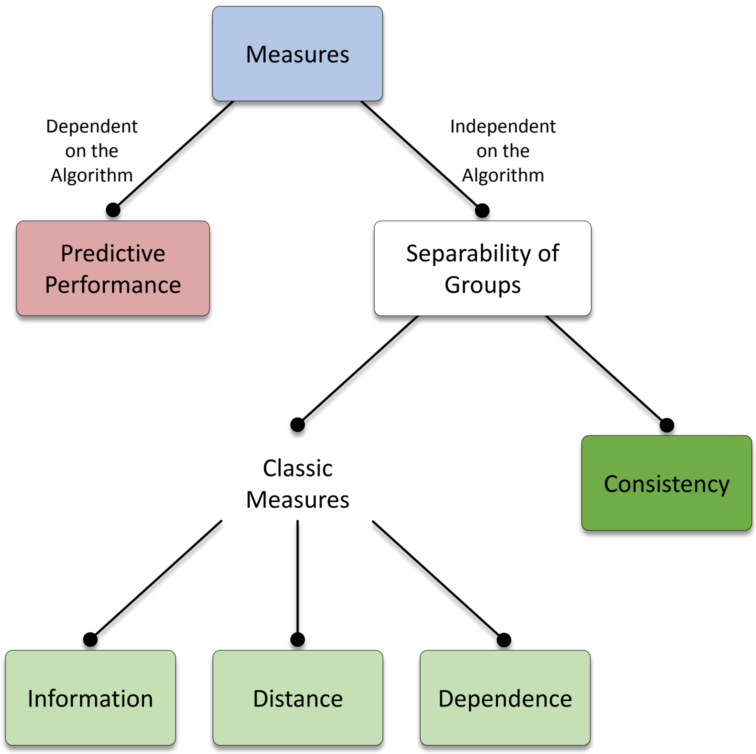


Figure 2: Hierarchy of importance measures (modified from Lee (2005))

3. Metalearning for Algorithm Recommendation

Metalearning was originally described by Maudsley (1979) in agreement with Definition 2, where the concept that describes the process of becoming aware of oneself as a learner and applying this knowledge toward becoming a more effective learner is introduced.

Definition 2. *Metalearning is the process by which learners become aware of and increasingly in control of habits of perception, inquiry, learning, and growth that they have internalized.*

Originally focused on Social Psychology, the Definition 2 was propagated over time to other areas of knowledge. In this context and according to Lemke et al. (2015) and Vilalta and Drissi (2002), nowadays there are different interpretations of the term Metalearning. In Educational Psychology and Physical Science, for example, Metalearning begins with raising awareness of learning, listening for feedback, praising advancement, and getting lots of practice. This is how one learns to learn and, for an individual, such process includes things as knowing and choosing the best way to learn, selecting the best sources of information, making to continuous reflection, and feeling the need to moving to a reinforcement learning environment.

When the above concept is extended to the ML area, the Metalearning is synthesized by Definition 3 (Brazdil et al., 2009).

Definition 3. *Metalearning is the study of principled methods that exploit meta-knowledge*[3] *to obtain efficient models and solutions by adapting Machine Learning and Data Mining processes.*

In other words, the Metalearning can be seen as a "hot research topic", which has emerged from the need to improve the generalization ability and stability of learned models and support DM automation in issues related to algorithm and parameter selection. It is the process of generating meta-knowledge that relates the performance of ML algorithms to the characteristics of the problem, *i.e.*, characteristics of its data sets (Bhatt et al., 2013, 2012).

The idea behind using Metalearning to support algorithm selection is not new. One of the first and seminal contributions has been provided by Rice (1976). The main question in his work was to predict which ML algorithm has the best performance for an exclusive data set and it was studied relating the inductive bias of each algorithm to the morphology of the investigated data. Formally, the algorithm selection problem is formalized as follows:

Definition 4. *For a particular problem instance $E_i \in E$, with features $a(E_i) \in A$, find the selection mapping $\Omega(a(E_i))$ into the algorithm space Δ, such that the selected algorithm $\delta \in \Delta$ maximizes the performance mapping $y(\delta, a(E_i)) \in Y$.*

From Definition 4, given a problem subset of the problem space E, a subset of the algorithm space Δ, and the performance measure space Y, Metalearning is applied to determine the Ω automatically (*i.e.*, the mapping of problems to algorithms) so as to have high algorithm performance. In this sense, the meta-knowledge in Metalearning is acquired from a set of training examples, such as portrayed in Figure 3.

[3] Meta-knowledge refers to any kind of knowledge that is derived in the course of employing a given learning system.

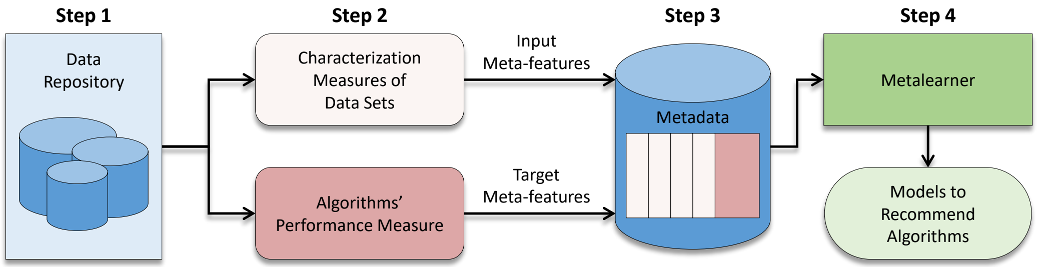

Step 1
Data Repository
Step 2
Characterization Measures of Data Sets
Algorithms' Performance Measure
Input Meta-features
Target Meta-features
Step 3
Metadata
Step 4
Metalearner
Models to Recommend Algorithms

Figure 3: Metalearning to obtain meta-knowledge for algorithm selection (modified from Brazdil et al. (2009))

In Figure 3, each training example, or meta-example, is related to a particular problem (a data set from the Data Repository), and stores: (1) the descriptive characteristics of the problem, so-called Input Meta-features; and (2) information on the performance of the most promising FS algorithm when applied to the problem, which are called of Target Meta-features. From a set of such examples (Input Meta-features associated with Target Meta-features), a ML algorithm (*i.e.*, a Metalearner), automatically acquires knowledge, associating the problem's characteristics with the algorithms' performance.

As for the ML area, the increase of availability of algorithms in the last years, and many of them with the need for multiple parameter configuration, also increased the difficulty in choosing which algorithms would be the most suitable for a particular application.

Thus, generally, Metalearning can be applied into two main aspects:

1. Assist ML to increase efficiency through previous experience in different problems, domains and tasks, more specifically in the aid to choose algorithms based on the relation between characteristics of the data sets and performance of the algorithms for a certain problem;
2. Help determine the best parameter setting, among the many possible, for algorithms that perform specific tasks.

Metalearning differs from the traditional view of learning, also called base-learning, in the scope of the level of adaptation. Whereas learning at the base-level is focused on accumulating experience on a specific learning task (*e.g.*, credit rating, medical diagnosis, bankruptcy prediction, fraud detection, among others), learning at the meta-level is concerned with accumulating experience on the performance of multiple applications of base-learning system. If a base-learner fails to perform efficiently, one would expect the learning mechanism itself to adapt in case the same task is presented again (Brazdil et al., 2009). Therefore, ML algorithms applied for base learning help us to obtain better and deeper understanding of collected data, thus assisting in driving some DM tasks, such as classification, regression, clustering and anomaly detection. On the other hand, Metalearning techniques extend this concept by providing methods to automate the DM process – composed of the preprocessing, patterns extraction, and postprocessing steps – as a whole.

Another interesting matter is how Metalearning deals with bias, *i.e.*, how assumptions influence the choice of hypotheses to explain the data. While in base-learning the bias is fixed, in Metalearning we try to choose the correct bias dynamically. For example, in a typical inductive learning scenario, applying a base-learner (*e.g.*, decision trees, nearest neighbors, or support vector machines) on some data produces a predictive function (hypothesis) that depends on the fixed assumptions embedded in the learner. Learning takes place at the base level because the quality of the predictive function normally improves with an increasing number of examples. Nevertheless, successive applications of the learner on the same data always produces the same hypothesis, independently of performance (Brazdil et al., 2009). In contrast, learning from experience when different biases are appropriate for a particular problem is the main goal of Metalearning.

In some situations, ensemble methods, *i.e.* committees of data models, are also acknowledged as Metalearning systems. Since they do not automate decisions concerning applications of learning models they should rather belong to the additional or post-learning stage of data analysis. The combination of data models, although does not favor the understanding into the hidden structures in the data, leads to a very rough granularity of knowledge, thus exploring only a small subspace of all possible models.

The result of Metalearning's application can be based then on several measures for algorithm recommendation, such as execution time, performance of the induced model or complexity of this model.

The traditional application area of Metalearning for algorithm recommendation is classification (Lemke et al., 2015). Other applications were also studied, such as regression, ordering, restriction satisfaction, optimization and time series prediction (Kück et al., 2016; Lemke and Gabrys, 2010; Smith-Miles, 2009; Wang et al., 2009; Prudencio and Ludermir, 2004). However, the use of Metalearning for the FS task is still largely unexplored (Filchenkov and

Pendryak, 2015; Shilbayeh and Vadera, 2014; Wang et al., 2013; Parmezan et al., 2012a,b).

For whatever the Metalearning problem is, there are three main aspects to be considered (Lemke et al., 2015):

1. Choice of the Metalearning algorithm: Refers to the algorithm that is going to be used for the construction of the metamodel. Usual choices include decision trees, neural networks, support vector machines, and nearest neighbors;
2. Selection of the appropriate meta-knowledge or characterization of the data sets: In general, meta-knowledge refers to the relationship between algorithm performance in a certain domain with the characteristics of the task, for example, the data used as input for these algorithms. The most common form of meta-knowledge includes the extraction of morphological characteristics from the data and their relationship with the algorithm performance. It is important to note that the definition of meta-features is dependent on the problem to be treated. They can be arranged into five categories (Reif et al., 2014): simple, statistics, based on information theory, landmarking and based on models. Table 2 lists the most commonly used meta-features, which are mathematically detailed in the Supplementary Material[4];
3. Configuration and maintenance of metabases: As the central concept of Metalearning is to thrive on the knowledge obtained considering data of similar problems or of other domains, an important question is the existence of enough metadata available.

Besides those three aspects, we still have to consider, like any other domain representation problem, the extraction of meta-features, which need to be representative for the problem at hand. The Metalearning efficiency depends directly on the description of the data sets for which the algorithm recommendation will be performed. An inappropriate choice of characterization measures could compromise the whole Metalearning process, generating a recommendation model that is inadequate for the desired purpose.

In Table 2, measures presented in the simply category aim to describe general characteristics of the data sets, while measures based on information theory seek to characterize the nominal attributes and their relationship

[4] https://goo.gl/Ke7LQb.

Table 2: Usual measures of characterization of data sets

Category	**Measure**	**Description**
Simple	1. Number of attributes 2. Number of qualitative attributes 3. Number of quantitative attributes 4. Number of examples 5. Number of classes	Describe properties obtained from the attribute-value representation.
Statistics	1. Average asymmetry of the attributes	Seeks to characterize how and how much the data distribution departs from symmetry condition. Depending on the value assumed by this coefficient it is possible to categorize the distribution as symmetric, moderate asymmetric or strong asymmetric. If the distribution is not symmetric, it may be positive or negative asymmetry.
	2. Average kurtosis of the attributes	Expresses the degree of flatness of a distribution, which in most cases is regarded in relation to a normal distribution. The shape of the distribution curve according to kurtosis can be categorized as mesokurtic, platykurtic or leptokurtic.
	3. Average correlation between attributes	Quantifies the degree of the linear relation between random attribute pairs.
	4. Average coefficient of variation of the attributes	Provides the variation of the obtained data related to the average. The lower the value, the more homogeneous is the data.
Information	1. Class entropy	Indicates the approximate amount of information needed to identify the label of the class of an example from the original data set.
	2. Average entropy of the attributes	Estimates the amount of information that a particular attribute has to offer on the prediction of the class.
	3. Average conditional entropy between classes and attributes	Includes the uncertainty degree of the class when the values of a random attribute are unknown.
	4. Average mutual information between classes and attributes	Measures the reduction of the entropy caused by the partition of the examples according to the values of a given attribute. Because it favors attributes with more values, this information measure is considered biased.
	5. Signal/noise ratio	Estimates the amount of not useful information of a data set and is defined as the subtraction of the average mutual information, between classes and attributes, from the average entropy of the attributes and then the result is divided again by the average mutual information between classes and attributes.

with the class attribute. As for measures of the statistics category, they are applied in order to verify the order, description, and distribution of numerical data. These usual characterization measures present a low cost for their computation and may be used in different application domains (Reif et al., 2014; Peng et al., 2002).

In contrast, the landmarking characterization consists in the employment of simple algorithms of classification, called landmarkers, on data sets to obtain important information about the nature of the domain to which they are applied. Commonly, landmarking is used to determine the proximity of a data set related to others, through performance similarity of landmarkers. Among the most common landmarkers, the Naïve Bayes is highlighted and its performance can be interpreted as an independence measure between attributes (Bensusan and Giraud-Carrier, 2000; Pfahringer et al., 2000).

The characterization based on models, as well as landmarking, also uses classification algorithms to represent the data sets. However, it does not directly consider measures of performance of the learning algorithm, but the structure of the classifier itself, known as the induced hypothesis or model. Methods for induction of decision trees are frequently employed for this task, since from the constructed tree it is possible to determine a set of metafeatures derived from the number of intern nodes, the number of leaf nodes, the depth of the tree and its height.

At first, it may seem that the use of characterization based on models is more advantageous than other types of characterization methods, as its application allows the data set to be summarized by a data structure that embeds the complexity and performance of the constructed model and not only the data distribution. However, there is a computational cost associated to the construction of these models and, in some cases, the obtained representation can mask important properties which assist in the explanation of the performance of the investigated algorithm and, consequently, in its correct recommendation. Similarly, this problem also affects the landmarking characterization. Still, the estimative of complexity of the data set is a relevant issue and that, if considered, can enhance the quality of data metabase.

In order to provide support to this issue, this paper proposes, in addition to the use of the characterization measures presented in Table 2, the inclusion of other seven measures (Reif et al., 2014; Ho et al., 2006; Lee et al., 2006) little disseminated in the Metalearning field for FS. Table 3 presents a summary of the properties of those seven measures, four of which are organized into a new category called complexity. All mathematical formulations of these measures can be found in the Supplementary Material.

Table 3: Measures for characterization of the complexity and dimensionality of data sets

Category	**Measure**	**Description**
Statistics	1. Balancing of the data set	Includes the level of balance of the data set and is defined by the ratio between the number of examples of the classes with the lowest and highest number of examples. The lower the value, in the range of zero and one, the bigger the imbalance.
	2. Majority class error	Refers to the error in the case of new examples being classified as belonging to the majority class.
Information	Equivalent number of attributes	Ratio between the entropy of the class and the average mutual information between classes and attributes.
Complexity	1. Fractal dimension of the data set	Expresses the inherent dimension of the data set, so that it can be approximated in smaller dimensions due to the presence of redundancy in the original data.
	2. Fisher's discriminant	Quantifies the Fisher discrimination rate. A high value indicates the existence of a transformation vector that can separate the examples belonging to different classes after the projection of the examples in this new attribute space.
	3. Volume of the overlapping region	Represents the overlap of the limits defined by examples of each class. A low value shows that the attributes can discriminate the examples of different classes.
	4. Dispersion of the data set	Indicates the dispersion degree of the data by calculating the ratio between the number of examples and the number of attributes of the data set.

The measures shown in Table 3 are intended to describe how the data are geometrically structured, considering inherent properties such as separability of classes, overlap in attribute space, topology and density.

4. Related Work

The concept of Metalearning used in DM is commonly related to the indication of pattern extraction algorithms, based mainly on morphological characteristics of the input data sets and the construction of hybrid models or ensembles, of the same or different paradigms (Lemke et al., 2015).

The description of data sets based on the application of theoretical and mathematical measures was approached, initially, in StatLog project (Michie et al., 1994). In this project, 15 meta-features, which contemplated simple measures, statistical and of information, were used to describe 21 databases. This set of measures was applied in several studies, all with the general purpose of recommending ML algorithms. Subsequent extensions to StatLog project were proposed by Lindner and Studer (1999) and Sohn (1999). In this sense, the MetaL[5] project expanded the set of measures proposed in StatLog (15 + 8) to develop tools to aid the selection of algorithms for DM. The project promoted the development of various aspects related to research in Metalearning, especially regarding the choice and use of characterization measures, as well as the combination of pattern extraction algorithms. Still in the MetaL project, the Data Mining Advisor system was developed with the aim of ranking a set of classification algorithms for the users (Giraud-Carrier, 2005).

In recent years, the foundations of Metalearning area have been explored in other application domains. For example, in time series, to select the most promising methods for the task of prediction of values (Kück et al., 2016; Lemke and Gabrys, 2010); in bioinformatics, providing support for next generation DNA sequencing (Stiglic et al., 2010); in regression, to assist the estimation of the parameters required by support vector machines (Gomes et al., 2012); and in clustering, in order to determine the best number of groups (Lee and Olafsson, 2013).

The problem about recommendation of FS algorithms using Metalearning, approached in this work, is relatively new and, therefore, there are few studies that offer significant advances towards solving it. This gradual advancement of the community can be explained by the lack of generality of the problem, *i.e.*, recommending FS algorithms is different from recommending ML algorithms, because in the FS context the performance of the algorithm to be recommended is biased towards the paradigm of the induced model using the selected subset of features. This fact makes the recommendation problem of FS algorithms even more challenging.

[5] MetaL: *Metalearning assistant for providing user support in machine learning and data mining*, http://www.metal-kdd.org, 2002.

Besides, it is important not only to evaluate the predictive performance of the model constructed from the selected subset of features, but also the percentage of reduction of attributes and the learning time for the construction of such model. Another important aspect is related to the construction of meta-features, which should allow the prediction of the performance of FS algorithms and, consequently, provide the discrimination between these predictions.

In Parmezan et al. (2012a) a study was conducted for the construction of a symbolic recommendation model of the best FS algorithm. In this study we considered two FS methods based on consistency and dependence measures. For the construction of the metabase, 30 data sets and 10 characterization measures were used, of which five are simple category measures and the rest are of the statistical category. The constructed recommendation model showed low accuracy, but enabled to see the importance of the use of statistical measures in the data characterization step. This preliminar recommendation model was enhanced in Parmezan et al. (2012b), in which it was proposed an architecture of recommendation of FS algorithms that uses 10 measures, five statistical and five of information, to characterize 17 data sets. Although the generated recommendation model has shown a considerably accuracy rate, it was found that some of the employed measures were incapable to represent the implicit knowledge contained in the data sets.

In Wang et al. (2013) it was presented a lazy method for the recommendation of FS algorithms. This method identifies, with the aid of a distance measure and using the k-Nearest Neighbors (kNN) classification algorithm, the k most similar data sets to a given new data set. Then, the method estimates, according to a multicriteria metrics, the performance of all candidate FS algorithms on the similar k data sets, and based on this information orders the r FS algorithms considered the most promising. The main disadvantage of this similarity-based method is the need to estimate the value of the parameter k, which is particular to each problem. Besides, the cost to identify the k most similar historical data sets can be high, because in the worst case the algorithm will consider all the meta-examples in the metabase for comparison. The training metabase was constructed from 115 data sets represented using 13 characterization measures, of which four are simple category measures, three of the statistical and six of the information category. It was then evaluated using four FS algorithms, each with different strategies and search directions. These four algorithms are based on measures of consistency, distance, dependence and probabilistic significance. The results showed the effectiveness of the proposed method for recommendation of FS algorithms.

In Shilbayeh and Vadera (2014) it was developed a Metalearning framework that is able to learn about which FS algorithms work best for a given data set. Part of this framework was implemented and tested using a metabase which describes 26 data sets. This description involved six characterization measures, of which half are simple category measures and the rest are of the information category. The extracted measures were linked to three FS approaches under different search techniques, along with applying six classifiers and fed to a symbolic ML algorithm in order to create the recommendation model. The biggest problem of this metamodel is that the indication of a FS algorithm involves the use of a specific classifier, so that users need to know beforehand what ML algorithm will be used to extract patterns. Anyway, the results were promising and showed that the meta-knowledge produced appears to be useful.

The authors in Filchenkov and Pendryak (2015) introduced a new approach for meta-features engineering, which showed to be useful for FS algorithm recommendation. A metabase was built using 84 data sets each one represented by 79 meta-features. These characteristics were extracted from usual measures (simple, statistical, and information theory meta-features) and measures based on models. This last group of characterization measures covered meta-features built with decision tree, kNN, and perceptron. Three FS algorithms based on consistency, dependence, and probabilistic significance were investigated with different strategies and search directions. Leave-one-out cross-validation was adopted to estimate the performance of base classifiers, which were induced by four learning algorithms: Naïve Bayes, C4.5, PART, Bayes Net and IB3. As a result, they found a meta-feature set which showed the best result in predicting proper FS algorithms.

Zabashta et al. (2016) developed a method, which for each rank list, it determines which rank aggregation algorithm is the best for this rank list. The paper extended the previous research on this problem (Zabashta et al., 2015) by suggesting new approaches involving meta-feature selection. The experiments showed that the best approach was based on the combination of the basic approaches for meta-feature description and application of FS algorithms to the resulting set. In this context, the authors provide the steps for implementing the selection system of the rank aggregation algorithm to the problem of FS.

An extended Metalearning framework for ranking and selection of algorithms for clustering gene expression microarray data was presented in Vukicevic et al. (2016). The modified framework has several improvements compared to the original one, such as extended algorithm and meta-feature spaces, as well as some techniques for meta-feature selection and parameter optimization of meta-algorithms. The Metalearning system was tested using 504 algorithms over 30 data sets and provided good results in the prediction of algorithm performance for specific problems.

Cui et al. (2016) proposed a general meta-modeling recommendation system using Metalearning. The system automates the process of meta-modeling choosing by intelligently adapting the learning bias to problem characterizations. In the paper, new meta-features, such as the gradient-based features for characterizing the geometrical properties of the response surface, were introduced. To further improve the performance of meta-learning for the meta-modeling recommendation, different types of FS methods, including singular value decomposition, stepwise regression and ReliefF, were studied. The intelligent system was evaluated on 44 data sets, and the results showed that it can reach 94% correlation on model rankings, and a 91% hit ratio on the best model recommendation. The authors also highlighted that the computational cost of meta-modeling ranking is significantly reduced from an order of minutes to seconds compared to traditional trial-and-error and ensemble process.

In (Luo, 2016) it was developed a software to automate the building of ML predictive models with big clinical data. The computational system, namely PredicT-ML, include a new method to automatically select algorithms, FS techniques, and hyper-parameter values for a given DM problem. According to the authors, PredicT-ML regards the identification of a FS method to be a hyper-parameter and automatically selects the algorithm and its parameter values. For healthcare administrators, the principal advantage of this software lies in the possibility of using predictive models to improve outcomes and reduce costs for various diseases and patient populations.

Despite the studies conducted in Cui et al. (2016), Luo (2016), Vukicevic et al. (2016), and Zabashta et al. (2016) are not closely related to the main goals of this paper, they present significant advances on the Metalearning theme, mainly because their methods support the FS task. Table 4 compares our proposal with five approaches previously described, which focus the problem of FS algorithms recommendation. The following criteria are considered:

Table 4: Some properties of frameworks designed to recommend Feature Selection algorithms published in related work

Paper	Methods of FS	#DS	#Characterization Measure(s)	Performance Measure(s)	Metalearner	Type of Recommendation	#P	Statistical Test(s)
Parmezan et al. (2012a)	Filter algorithms based on measures of consistency and dependence	30	10 (five simple and five of statistical)	Predictive Performance	J48	Best algorithm	—	—
Parmezan et al. (2012b)	Filter algorithms based on measures of consistency and dependence	17	10 (five of statistical and five of information)	Predictive Performance	J48	Best algorithm	—	—
Wang et al. (2013)	Filter algorithms based on measures of consistence, distance, dependence and probabilistic significance	115	13 (four simple, three of statistical and six of information)	Multicriteria Performance Measure	kNN	Rank of algorithms	2	Friedman and Holm
Shilbayeh and Vadera (2014)	Algorithms based on measures of dependence, distance and predictive performance	26	6 (three simple and three of information)	Predictive Performance	J48	Best algorithm	—	—
Filchenkov and Pendryak (2015)	Filter algorithms based on consistency, dependence, and probabilistic signicance	84	79 (four simple, five of statistical, six of information, and 64 based on models)	Multicriteria Performance Measure	—	—	—	—
Our Proposal	Filter algorithms based on measures of consistency, distance, dependence, and information	150	21 (five simple, six of statistical, six of information, and four of complexity)	Multicriteria Performance Measure	J48	Rank of algorithms	—	Kruskal-Wallis and Dunn

- **Methods of FS:** The FS algorithms to be recommended and the importance measures codified by each one them;
- **#DS:** The number of data sets selected to assist in the construction of metabase(s);
- **#Characterization Measure(s):** The amount of measures used to extract meta-features;
- **Performance Measure(s):** The measures chosen to evaluate the performance of FS algorithms;
- **Metalearner:** The supervised ML algorithm used to induce recommendation models;
- **Type of Recommendation:** The manner the recommendation of FS algorithms is performed, *i.e.*, suggesting the best algorithm or a ranking of algorithms;
- **#P:** The number of parameters that must be specified;
- **Statistical Test(s):** The method employed to verify the occurrence of significant differences among metamodels.

In this work, we propose an architecture that uses not only the usual measures for characterization of the data sets, but also includes seven complexity and dimensionality measures not much disseminated in the Metalearning theme for FS. The quality of the FS methods was determined according to a multicriteria performance measure, which guided the ranking process of these algorithms for the construction of the metabases.

It should be emphasized that, although a few publications in Table 4 study a lot of characterization measures, the most important issue in the building of training metabases is not the amount of meta-features that describe the problems, but the discrimination power they offer.

The proposed recommendation models in this paper, represented by decision trees and induced from the training metabases, are interpretable and robust to overfitting, and less computationally expensive than newly designed approaches in the literature.

5. Proposed Architecture

The proposed architecture for the recommendation of FS algorithms based on the use of Metalearning was organized in four steps, as shown in Figure 4.

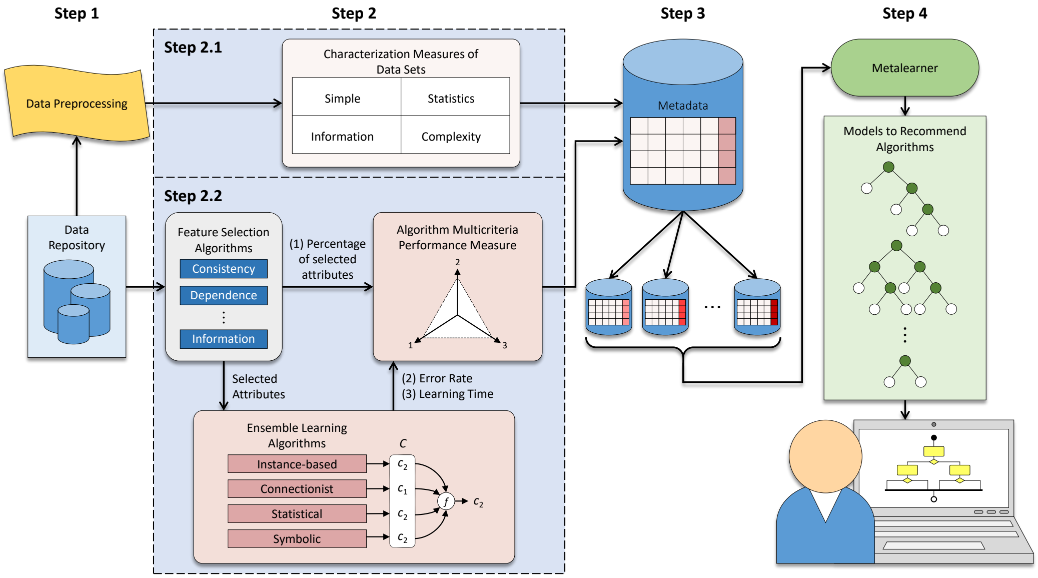


Figure 4: Proposed architecture of recommendation for Feature Selection algorithms

Step 1 – Preprocessing of Data Sets:

In this step, a repository containing some representative problems for which the subsequent recommendation will be held is constructed. In the Data Repository in Figure 4, each problem (data set) is expressed in the attribute-value file format (Table 1), where features represent important characteristics in the data domain. Each Original Data Set with N examples × M features from the Data Repository will feed Data Preprocessing (Step 1 in Figure 4, for more details see Figure 5) as well as the Feature Selection Algorithms in Step 2 (Step 2.1 in Figure 4).

The Original Data Sets from the Data Repository may present missing values for one or more features. Since the lack of these values hinders the extraction of measures (Step 2.2 in Figure 4), it is necessary to use techniques to ensure the completeness of the data and yield the Data Set with Full Values. There are several methods in the literature available to treat missing values (Purwar and Singh, 2015). However, the

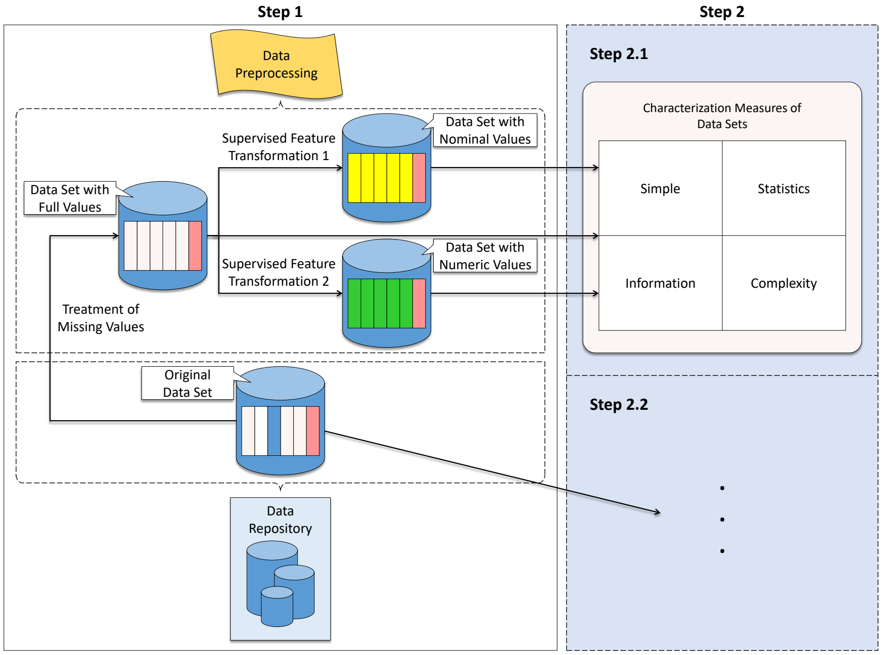

Step 1
Step 2
Data Preprocessing
Step 2.1
Characterization Measures of Data Sets
Simple
Statistics
Information
Complexity
Data Set with Nominal Values
Supervised Feature Transformation 1
Data Set with Full Values
Supervised Feature Transformation 2
Data Set with Numeric Values
Treatment of Missing Values
Original Data Set
Step 2.2
Data Repository

Figure 5: Details of the data sets preprocessing activity initially illustrated in Figure 4

simplest of them replaces all the missing values for nominal and numeric attributes in a data set with, respectively, the modes and means from the data set.

After completing the data, from each Data Set with Full Values, other two sets with the same dimensionality ($N \times M$) of the Original Data Set are derived as depicted in Figure 5. The difference between both obtained sets lies in the fact that the Data Set with Nominal Values is only described by nominal features, while the Data Set with Numeric Values consists of numeric features only. This procedure can be performed using supervised feature transformation methods (Witten et al., 2011), and its execution is necessary because some characterization measures work with a specific type of data, *i.e.*, qualitative or quantitative. There is no consensus about which supervised algorithm for feature transformation should be used in this step, but some of the most promising ones are available in Weka[6] tool. The three data sets

[6] http://www.cs.waikato.ac.nz/ml/weka.

built in Step 1 will feed the next one. In particular, the Data Set with Full Values is submitted to the Characterization measures that do not act directly over the features values.

Step 2 – Extraction of Measures:

Two procedures (Step 2.1 and Step 2.2 in Figure 4) are applied to each problem of the Data Repository treated in Step 1:

Step 2.1: Characterization of each preprocessed data set (Data Set with Full Values, Data Set with Nominal Values and Data Set with Numeric values), in agreement with Step 2.1 in Figure 5. This procedure is conducted according to the measures belonging to one of the following categories: (1) simple, (2) based on statistics, (3) based on information theory and (4) complexity;

Step 2.2: Calculation of the multicriteria evaluation measure regarding the performance with and without FS (Original Data Sets), based on learning algorithms. This procedure, indicated in Step 2.2 in Figure 4, consists of three substeps:

1. **Algorithms for FS:** Choice of two or more FS algorithms whose recommendation is desired. The FS algorithms can be performed in two ways, ranking or FSS, employing some importance measure based on consistency, dependence, distance, information or predictive performance;
2. **Individual Performance Measures:** From the original sets and the selected subsets of features, models are built using a system based on ensembles of classifiers, which predicts the class of an example by electing the majority of votes produced by base classifiers. It is recommended as base classifiers, the use of classification methods of different learning paradigms. In general, the literature splits the inductive learning into four paradigms: instance-based, connectionist, statistical and symbolic. The application of classifiers ensembles aims to extract three individual performance measures of FS algorithms:
    - Error rate, estimated in accordance to some validation method, for example holdout, leave-one-out or cross-validation;

   - Learning time, in seconds, needed for the construction of the model using the original data sets and the subsets of attributes selected by FS algorithms;
   - Percentage of selected features by FS algorithms.

   When FSS algorithms do not hold any feature as important, all features of the original data set should be considered.

3. **Multicriteria Performance Measure:** In order to obtain an estimate of the quality of FS algorithms, we developed in this paper a multicriteria measure that allows us to evaluate, for each data set, the FS algorithms related to the three individual measures of performance described. An illustration of the operations of this measure can be seen in Figure 6.

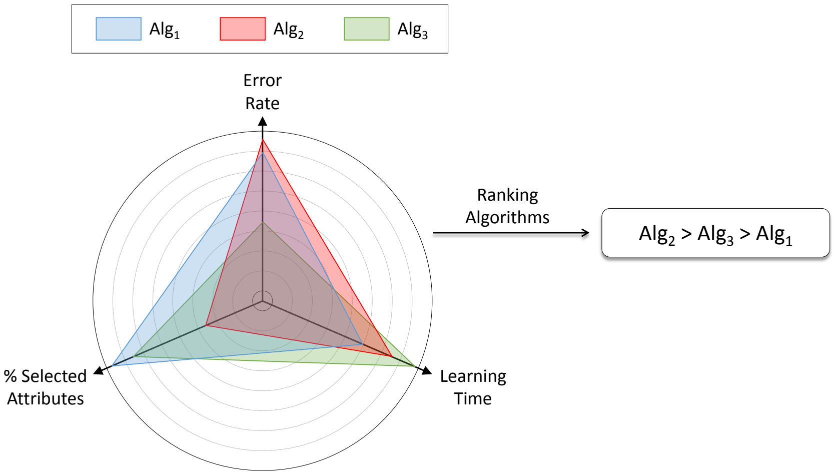
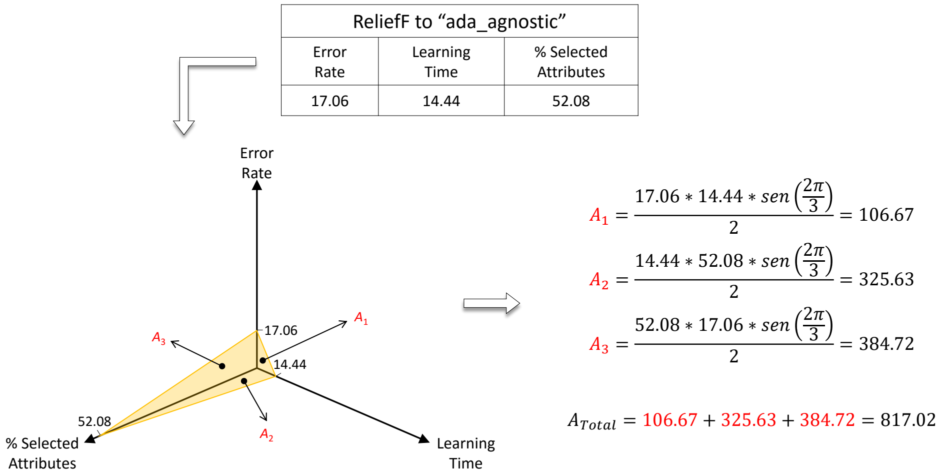


Figure 6: Algorithms multicriteria performance measure

In this figure we illustrate a radar chart consisting of three axes, each one representing an individual performance measure. The area of each irregular triangle, formed from the meeting of edges with vertices that represent the measures, can be calculated, as shown in Figure 7, using Equation 1.

$$A = \frac{a * b * \sin\left(\frac{2\pi}{3}\right)}{2} \tag{1}$$



ReliefF to “ada_agnostic”		
Error Rate	Learning Time	% Selected Attributes
17.06	14.44	52.08

$$A_1 = \frac{17.06 * 14.44 * sen\left(\frac{2\pi}{3}\right)}{2} = 106.67$$

$$A_2 = \frac{14.44 * 52.08 * sen\left(\frac{2\pi}{3}\right)}{2} = 325.63$$

$$A_3 = \frac{52.08 * 17.06 * sen\left(\frac{2\pi}{3}\right)}{2} = 384.72$$

$$A_{Total} = 106.67 + 325.63 + 384.72 = 817.02$$

Figure 7: Multicriteria performance measure of the ReliefF algorithm applied to the “ada_agnostic” data set

In Equation 1, a and b denote the sides of the triangle and $^{2\pi}/_{3}$ corresponds to the angle formed between them. This equation expresses the area theorem, which is used to calculate the area of any triangle when two sides of this triangle are known, as well as the angle composed by them. For each investigated algorithm, three areas are computed whose total sum reflects on its quality. The lower the value of the total area, the more efficient the algorithm is for the FS task.

Step 3 – Construction of Data Metabases:

The results obtained in Step 2 are associated to compose meta-examples, each consisting of morphological meta-features (characterization measures of the data sets) and nominal target attribute (ranking of the most appropriate algorithms for the FS task according to the multicriteria performance measure), respectively.

In this work we defend the idea of treating FS algorithms recommendation problem as binary classification. Therefore, when the target attribute is constituted by three or more labels (FS algorithms), it is suggested to decompose this attribute for generation of new metabases. For example, if we have interest in the recommendation of the best FS algorithm from three available, we can decompose this problem using *a priori* knowledge. Figure 8 illustrates this reasoning assuming that the algorithms are based on measures of consistency, information and predictive performance.

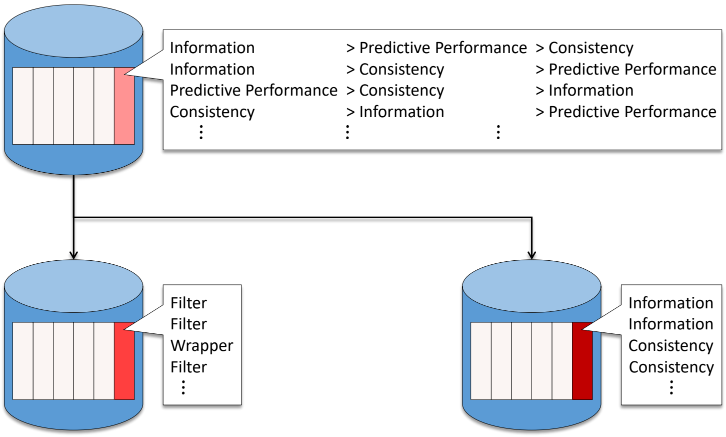

Information > Predictive Performance > Consistency
Information > Consistency > Predictive Performance
Predictive Performance > Consistency > Information
Consistency > Information > Predictive Performance
⋮ ⋮ ⋮
Filter
Filter
Wrapper
Filter
⋮
Information
Information
Consistency
Consistency
⋮

Figure 8: Example of decomposition of the nominal target attribute

In the example shown in Figure 8 two metabases are generated through the decomposition, differing only with respect to the nominal target attribute. In the first metabase, the class attribute took the labels "Filter" and "Wrapper", which were elected in accordance with the characteristics of the FS algorithm that occupies the first position in the ranking of the considered algorithms. Similarly, in the second metabase the aforementioned attribute included the labels "Consistency" and "Information". In summary, the first metabase describes the approach in which the FS is performed. If wrapper, the algorithm based on predictive performance measure is the most promising; if filter, this decision is carried out using the second metabase, which describes cases where the best FS algorithm may be based on consistency or information.

Step 4 – Induction and Use of the Recommendation Models: From the metabases generated in Step 3, the metamodels are induced using a symbolic supervised ML algorithm. As we discuss with more details throughout this paper, decision trees, in addition to being interpretable, provide results that are hardly surpassed by other learning algorithms. The recommendation models built in this step map the meta-knowledge embedded in the metabases, and will be used to guide the choosing of FS algorithms.

Afterwards, in order to recommend a FS algorithm to a new data set, the considered characteristics are extracted to form a meta-example of consultation. This meta-example is provided, respecting a preferred hierarchical order, to the induced metamodels, which in turn recommend the most suitable FS algorithm among the ones present in the recommendation model.

To assist in the construction of the symbolic recommendation models of FS algorithms we developed an application in Java programming language (Deitel and Deitel, 2014) using the Eclipse[7] development environment, integrated with the tool Measure Distance Exponent (MDE) (Traina et al., 2003) and employing libraries from Weka[8] and Java/R Interface (JRI)[9].

6. Empirical Evaluation

The proposed framework was applied to recommend FS algorithms based on measures of consistency, dependence, information and distance. In this section we describe the main algorithms used, the experimental setup, including the evaluation mode of the implemented architecture, as well the considered data sets.

6.1. Feature Selection and Learning Algorithms

Four FS algorithms were considered is this work:

CBF – Consistency-Based Filter (Dash and Liu, 2003)**:** This method generates subsets of features and evaluates them regarding its size and inconsistency related to the class. The selected subset is the one that, within the maximum number of attempts *max_tries*, has the smallest size and the smallest inconsistency with regard to the class. In other words, it is considered a probabilistic algorithm characterized by evaluating subsets of features according to their consistency with respect to a given label. Usually, the search favors small subsets of features that present high consistency with the class. This algorithm has a time complexity of $O(N \cdot M^2)$;

[7] https://eclipse.org.

[8] http://www.cs.waikato.ac.nz/ml/weka.

[9] http://www.rforge.net/JRI.

CFS – Correlation-based Feature Selection (Hall, 2000)**:** This algorithm considers the individual predictive ability of each feature and the correlation degree between these features, including the class. Thus, it is possible to select subsets of features that are formed by relevant and not redundant features. Therefore, relevance weights are assigned to subsets of features according to the separability evaluation measures, such as the Symmetrical Uncertainty measure (Press et al., 1992). This algorithm has a time complexity of $O(N \cdot M^2)$ (Hall, 1999);

InfoGain (Das, 2001)**:** This algorithm selects features through individual evaluation. The basic idea of this method is to compute the information gain (Han et al., 2011), based on the entropy measure (Mitra et al., 2002), to evaluate the relevance of a feature related to the class. Finally, an ordering (ranking) of these features is generated in descending order with respect to their relevance to the class. The t first features are chosen to compose the selected subset of important features, where t is defined according to some criteria, for example, a percentage of the original number of features. This algorithm has a time complexity of $O(M \cdot T_2)$, where T_2 is the time to calculate the information gain;

ReliefF (Kononenko, 1994)**:** This method works by random sampling of examples of the data set and the location of the nearest neighbor of the same class and the nearest neighbor of the opposite class. Then, the values of the nearest neighbors are compared to the samples class and used to update the weights of importance of each feature in relation to the class. This process is repeated m times, where m is the number of times the algorithm looks for examples in the data set. The algorithm works with quantitative and qualitative features, but is limited to problems of two classes. For this reason the original Relief algorithm (Kira and Rendell, 1992) was extended in six variations and the version called ReliefF, which has time complexity of $O(m \cdot N \cdot M)$, is capable of dealing with incomplete data sets, with noises and with multiple classes (Spolaôr et al., 2011).

In Table 5, a summary of the main characteristics of these filter algorithms is presented. The information on this table is organized as follows: the first two lines indicate the evaluation strategy of the features, *i.e.*, ranking or subset of features. The remaining lines show the categories of importance measures employed by each one of the FS algorithms.

Table 5: Characteristics of the Feature Selection algorithms

	CBF	***CFS***	**InfoGain**	**ReliefF**
Individual Evaluation			✓	✓
Subset Evaluation	✓	✓		
Consistency Measure	✓			
Dependence Measure		✓		
Distance Measure				✓
Information Measure			✓	

As for the learning algorithms, we considered the following seven:

ADTree – Alternating Decision Tree (Freund and Mason, 1999)**:** Constitutes a kind of symbolic classification model, where an example is mapped to a path along the tree, from the root to one of the leaf nodes. However, unlike decision trees, the classification which is associated with the path is not the leaf node label but it is obtained based on the sign (positive or negative) of the sum of all the existing prediction nodes over the way. An ADTree gets its name because it is formed of alternating layers of prediction and partitioning nodes. One of the most interesting characteristics in this variant of decision trees is that, in addition to classification, it also provides a measure of confidence called of margin;

CART – Classification and Regression Trees (Breiman et al., 1984)**:** The trees generated by this algorithm are always binary, which can be traveled from the root node to the leaf nodes just answering simple questions such as yes/no ones. It is based on a non-parametric technique that induces both classification and regression trees. Besides, as is the case with the J48 algorithm, CART uses an exhaustive search method to define the threshold values to be employed in the nodes to divide the continuous attributes. In contrast to the approach adopted by other types of decision trees, which use pre-pruning, CART expands the tree thoroughly, performing post-pruning by reducing a cost-complexity factor. A large capacity to search by data relationships, even when they are not apparent, as well as the production of results in the form of decision trees of great simplicity and readability, are the main advantages of CART;

J48 (Witten et al., 2011)**:** Consists of an algorithm for decision trees induction from the symbolic paradigm. This method uses the strategy of dividing to conquer, *i.e.*, a problem is decomposed into simpler subproblems and this strategy is used recursively on each decomposed subproblem. The decision tree is a disjunction of conjunctions, in which each branch of the tree, from the root to the leaf, is a conjunction of conditions on the features and the set of branches of the tree is disjoint. Thus, each node of the tree specifies a test on some feature of the example and each branch of the node one of the possible values of the feature. J48 does not assume any particular distribution for the data and enables that a complex decision (predicting the value of the class) to be decomposed in a series of elementary decisions;

***k*NN – k-Nearest Neighbors** (Fix and Hodges, 1951)**:** The kNN is an instance-based learning algorithm that consists on finding, according to some similarity measure, the k examples that are nearest to an unlabeled example. The new example classification is decided on the labels of those k nearest examples. From this initial idea, several versions of the algorithm have been proposed. Among these variations, the most common considers that when $k = 1$ the new example will be classified as belonging to the same class of the single nearest example according to the similarity measure. In contrast, if $k > 1$ is considered, for example, $k = 3$, the predominant class of the three nearest neighbors (examples) will be assigned to the new example;

MLP – Multilayer Perceptron (Haykin, 2009)**:** Constitutes a kind of artificial neural network belonging to the connectionist paradigm, which was inspired by the study of neural connections of the nervous system and is based on the combination of highly connected simple units. In this model, there can be one or more layers of neurons between the layers of data input and results output. These intermediate layers are units that do not interact directly with the environment and work as combiners of characteristics. If there are appropriate connections between the input units and a considerable set of intermediate units, one can always find the representation that will produce the correct mapping between data input and results output (classification). One of the most used algorithms for training this type of network is the backpropagation;

Naïve Bayes (Witten et al., 2011)**:** Consists of a learning algorithm from the statistical paradigm. It assumes that the data follow a normal distribution and employs a probabilistic model in order to find an approximation of the induced concept. The Naïve Bayes can be understood as a specialized Bayesian network as it stands on two important assumptions: features are equally important and also statistically independent, so that when the class is known, knowing the value of a feature does not add any information on the value of another. In practical terms, the algorithm classifies a new example according to the most probable class, given its data set;

SVM – Support Vector Machines (Vapnik, 1999)**:** Although the SVM models present a structure similar to neural networks, they differ in how the learning is conducted. While neural networks work by minimizing the empirical risk, *i.e.*, the error minimization of the induced model on the training data, the SVM are based on the principle of minimizing the structural risk, which seeks the lowest training error while minimizing an upper bound on the generalization error of the model (model error when applied to test data). The generalization concept is best understood in the case of binary classification. Thus, given two classes and a set of points belonging to these, the SVM determines the hyperplane that separates them, so as to place the largest possible number of points of the same class on the same side, while the distance from each class to that decision surface is maximized. In situations where the samples are not linearly separable, the solution focuses on mapping the input data to a higher-dimensional space (feature space). This mapping is achieved by the use of a kernel function.

6.2. Experimental Setup

From the result of a systematic review of published works in the area of Data Mining, we selected 150 benchmark data sets of public access obtained from data repositories at the University of California at Irvine (Bache and Lichman, 2013), from *Uniwersytet* Warszawski[10], and from *Universidad* Pablo de Olavo[11]. These data sets are frequently reported in the literature

[10] http://tunedit.org/repo.

[11] http://www.upo.es/eps/bigs/datasets.html.

and come from different areas such as engineering, biological sciences, humanities, linguistics, health and computing. A summary of the characteristics of these sets is shown in Table 6, in which, for each data set, the following is described: number of examples (#E), total number of attributes (#A.Total), as well as number of qualitative (#A.QL) and quantitative attributes (#A.QT), number of classes (#C), Error of the Majority Class (EMC) and existence of missing values (?).

The experiments were conducted in four steps, as proposed in Figure 4.

Step 1 – Preprocessing of Data Sets:

Of the 150 data sets selected through the systematic review, 41 underwent a simple treatment of missing values. It consisted on the following: for attributes with missing values belonging to the same class, absent values were replaced by the mode, when regarding nominal data, and by the average, when regarding numerical data. Subsequently, supervised approaches to transform the attributes were applied over the 150 data sets with complete values:

1. Transformation of numeric data into nominal data by applying the attributes discretization method Minimum Description Length proposed in Fayyad and Irani (1993), which uses recursive heuristics of entropy minimization to modify the data;
2. Transformation of nominal data into numeric data by using the transformation method developed in Breiman et al. (1984), which converts nominal attributes into binary of the numeric type. Particularly, the method transforms a nominal attribute with k values into k binary attributes considering the values of the class attribute.

As a result of this step, 150 data sets were obtained composed of only nominal attributes (transformation 1), which will be summarized by characterization measures that operate on qualitative values, and 150 data sets composed by only numeric attributes (transformation 2), which will be synthesized by measures that work with quantitative values. Characterization measures that do not act directly over the attribute values were applied considering the original data sets.

Step 2 – Extraction of Measures:

In this step, two categories of measures were extracted:

Table 6: Summary of characteristics of the data sets

ID	Data Set	#E	#A.Total	#A.QL	#A.QT	#C	EMC(%)	?
1	ada_agnostic	4562	48	0	48	2	24.81	
2	ada_prior	4562	14	8	6	2	24.81	✓
3	adult_census	32561	14	8	6	2	24.08	✓
4	anneal	898	38	32	6	5	23.83	
5	anneal_original	898	38	32	6	5	23.83	✓
6	arrhythmia	452	279	73	206	13	45.80	✓
7	audiology	226	69	69	0	24	74.78	✓
8	australian	690	14	8	6	2	44.49	
9	autos	205	25	10	15	6	67.32	✓
10	backache	180	31	26	5	2	13.89	
11	balance	17	3	0	3	2	47.06	
12	balance_scale	625	4	0	4	3	53.92	
13	balloons	76	4	4	0	2	46.05	
14	bank	600	10	8	2	2	45.67	
15	biomed	209	8	1	7	2	35.89	✓
16	blood_transfusion	748	4	0	4	2	23.80	
17	breast_cancer	286	9	9	0	2	29.72	✓
18	breast_wisconsin	699	9	0	9	2	34.48	✓
19	bridges_version1	105	11	8	3	6	58.10	✓
20	bridges_version2	105	11	11	0	6	58.10	✓
21	bupa	345	6	0	6	2	42.03	
22	calories	40	2	0	2	3	50.00	✓
23	car	1728	6	6	0	4	29.98	
24	cars_with_names	406	8	2	6	3	37.44	✓
25	colic	368	22	15	7	2	36.96	✓
26	colic_original	368	27	20	7	2	33.70	✓
27	collins	500	22	2	20	15	84.00	
28	colon	62	2000	0	2000	2	35.48	
29	companies	79	6	0	6	9	78.48	
30	contact_lenses	24	4	4	0	3	37.50	
31	contraceptive_method	1473	9	7	2	3	57.30	
32	credit_a	690	15	9	6	2	44.49	✓
33	credit_g	1000	20	13	7	2	30.00	
34	cylinder_bands	540	39	21	18	2	42.22	✓
35	dermatology	366	34	33	1	6	69.40	✓
36	diagnosis	120	6	5	1	4	66.67	
37	ecml_2004	90	27679	0	27679	43	94.44	
38	ecoli	336	7	0	7	8	57.44	
39	eggs	48	3	1	2	2	50.00	
40	embryonal	60	7129	0	7129	2	35.00	
41	eucalyptus	736	19	5	14	5	70.92	✓
42	eye_movements	10936	27	3	24	3	61.03	
43	fertility	100	9	6	3	2	12.00	
44	flags	194	28	26	2	8	64.43	
45	gamma_telescope	19020	10	0	10	2	35.16	
46	gcm_test	46	16063	0	16063	14	86.96	
47	german	1000	20	13	7	2	30.00	
48	gina_agnostic	3468	970	0	970	2	49.16	
49	gina_prior	3468	784	0	784	2	49.16	
50	gina_prior2	3468	784	0	784	10	88.96	
51	glass	214	9	0	9	6	64.49	
52	grub_damage	155	8	6	2	4	68.39	
53	haberman	306	3	1	2	2	26.47	
54	hayes_roth_train	132	4	0	4	3	61.36	
55	heart_c	303	13	7	6	2	45.54	✓
56	heart_h	294	13	7	6	2	36.05	✓
57	heart_statlog	270	13	0	13	2	44.44	
58	hepatitis	155	19	13	6	2	20.65	✓
59	hepatobiliary_disorders	536	9	0	9	4	66.79	
60	hypothyroid	3772	29	22	7	4	7.71	✓
61	ionosphere	351	34	0	34	2	35.90	
62	iris	150	4	0	4	3	66.67	
63	irish_educational_transitions	500	5	3	2	3	35.00	✓
64	kdd_japanese_vowels_test	5687	14	0	14	9	79.08	
65	kdd_japanese_vowels_train	4274	14	0	14	9	85.82	
66	kdd_synthetic_control	600	60	0	60	6	83.33	
67	kr_vs_kp	3196	36	36	0	2	47.78	
68	labor	57	16	8	8	2	35.09	✓
69	lenses	24	4	4	0	3	37.50	
70	letter	20000	16	0	16	26	95.94	
71	leukemia	72	7129	0	7129	2	34.72	
72	leukemia_3classes	72	7129	0	7129	3	47.22	
73	leukemia_test	34	7129	0	7129	2	41.18	
74	leukemia_train	38	7129	0	7129	2	28.95	
75	lung_cancer	32	56	56	0	3	59.38	✓

Table 6: Summary of characteristics of the data sets

ID	Data Set	#E	#A.Total	#A.QL	#A.QT	#C	EMC(%)	?
76	lymphography	148	18	15	3	4	45.27	
77	lymphoma_11classes	96	4026	0	4026	11	76.04	✓
78	lymphoma_2classes	45	4026	0	4026	2	48.89	✓
79	lymphoma_9classes	96	4026	0	4026	9	52.08	✓
80	madelon	2600	500	0	500	2	50.00	
81	mammographic_masses	961	5	4	1	2	46.31	✓
82	mfeat_factors	2000	216	0	216	10	90.00	
83	mfeat_fourier	2000	76	0	76	10	90.00	
84	mfeat_morphological	2000	6	0	6	10	90.00	
85	mfeat_pixel	2000	240	240	0	10	90.00	
86	mfeat_zernike	2000	47	0	47	10	90.00	
87	molecular_biology_promoters	106	57	57	0	2	50.00	
88	monks_problems_1test	432	6	6	0	2	50.00	
89	monks_problems_1train	124	6	6	0	2	50.00	
90	monks_problems_2test	432	6	6	0	2	32.87	
91	monks_problems_2train	169	6	6	0	2	37.87	
92	monks_problems_3test	432	6	6	0	2	47.22	
93	monks_problems_3train	122	6	6	0	2	49.18	
94	mushroom	8124	22	22	0	2	48.20	✓
95	nursery	12960	8	8	0	5	66.67	
96	oh0_wc	1003	3182	0	3182	10	80.66	
97	oh10_wc	1050	3238	0	3238	10	84.29	
98	oh15_wc	913	3100	0	3100	10	82.80	
99	oh5_wc	918	3012	0	3012	10	83.77	
100	optdigits	5620	64	0	64	10	89.82	
101	page_blocks	5473	10	0	10	5	10.23	
102	parkinsons	195	21	0	21	2	24.62	
103	pasture_production	36	22	1	21	3	66.67	
104	pendigits	10992	16	0	16	10	89.59	
105	pima	768	8	0	8	2	34.90	
106	postoperative_patient	90	8	8	0	3	28.89	✓
107	primary_tumor	339	17	17	0	21	75.22	✓
108	prnn_crabs	200	7	1	6	2	50.00	
109	red_white_wine	6497	12	1	11	7	56.35	
110	satimage	1286	36	0	36	6	76.05	
111	segment	2310	19	0	19	7	85.71	
112	semeion	1593	256	256	0	10	89.83	
113	shuttle_landing_control	15	6	6	0	2	40.00	✓
114	sick	3772	29	22	7	2	6.12	✓
115	solar_flare1	323	12	12	0	6	72.76	
116	solar_flare2	1066	12	12	0	6	68.95	
117	sonar	208	60	0	60	2	46.63	
118	soybean	683	35	35	0	19	86.53	✓
119	spambase	4601	57	0	57	2	39.40	
120	spectf_test	269	44	0	44	2	20.45	
121	spectf_train	80	44	0	44	2	50.00	
122	spectrometer	531	101	1	100	48	89.64	
123	spect_test	187	22	22	0	2	8.02	
124	spect_train	80	22	22	0	2	50.00	
125	splice	3190	60	60	0	3	48.12	
126	sponge	76	44	44	0	3	7.89	✓
127	squash_stored	52	24	3	21	3	55.77	✓
128	squash_unstored	52	23	3	20	3	53.85	✓
129	sylva_agnostic	14395	216	0	216	2	6.15	
130	sylva_prior	14395	108	0	108	2	6.15	
131	tae	151	5	2	3	3	65.56	
132	tic_tac_toe	958	9	9	0	2	34.66	
133	tr11_wc	414	6429	0	6429	9	68.12	
134	tr12_wc	313	5804	0	5804	8	70.29	
135	tr23_wc	204	5832	0	5832	6	55.39	
136	tr31_wc	927	10128	0	10128	7	62.03	
137	tr41_wc	878	7454	0	7454	10	72.32	
138	tr45_wc	690	8261	0	8261	10	76.81	
139	trains	10	32	32	0	2	50.00	✓
140	vehicle	846	18	0	18	4	74.23	
141	vote	435	16	16	0	2	38.62	✓
142	vowel	990	12	2	10	11	90.91	
143	wap_wc	1560	8460	0	8460	20	78.14	
144	waveform	5000	40	0	40	3	66.16	
145	wdbc	569	30	0	30	2	37.26	
146	white_clover	63	31	4	27	4	39.68	
147	wine	178	13	0	13	3	60.11	
148	wine_quality_white	4898	11	0	11	7	55.12	
149	yeast	1484	8	0	8	10	68.80	
150	zoo	101	17	16	1	7	59.41	

1. Morphological properties of the data sets according to measures listed on Table 7, composed by measures presented in Tables 2 and 3;
2. Multicriteria measure to evaluate the performance with and without FS.

Table 7: Characterization measures of data sets

Category	ID	Measure
Simple	1	Number of attributes
	2	Number of qualitative attributes
	3	Number of quantitative attributes
	4	Number of examples
	5	Number of classes
Statistics	6	Average asymmetry of the attributes
	7	Average kurtosis of the attributes
	8	Average correlation between attributes
	9	Average coefficient of variation of the attributes
	10	Balancing of the data set
	11	Majority class error
Information	12	Class entropy
	13	Average entropy of the attributes
	14	Average conditional entropy between classes and attributes
	15	Average mutual information between classes and attributes
	16	Signal/noise ratio
	17	Equivalent number of attributes
Complexity	18	Fractal dimension of the data set
	19	Fisher's discriminant
	20	Volume of the overlapping region
	21	Dispersion of the data set

Regarding the performance measure of the algorithms, the following settings were adopted:

1. **Algorithms for FS:** CBF for the consistency measure, CFS for the correlation measure, InfoGain for the information measure and ReliefF for the distance measure. All these algorithms were executed considering their parameters configured with default values. The Best First search method with forward direction was employed for FSS algorithms. As for feature ranking algorithms, a

threshold of 50% of the total number features, sorted in descending order of importance, was considered. It is important to note that this value of 50% of reduction of the dimensionality may vary for more or less, since the original number of features belongs to the domain of integers;

2. **Individual Performance Measures:** In this work, we used as base classifiers, the algorithms J48, *k*NN, MLP and Naïve Bayes. These classifiers were executed considering their parameters configured with default values, except for *k*NN, with the *k* parameter established as five near neighbors, and MLP, in which the amount of hidden layers was defined as the number of classes of the data set. Then, three individual performance measures of the FS algorithms were extracted:

   - Error rate, estimated by 10-fold stratified cross-validation with ten repetitions (10x10 fold-cross-validation);
   - Learning time[12], in seconds, for the construction of models with and without FS;
   - Percentage of selected features by the FS algorithms.

   We considered all features of the original data set when algorithms of subset evaluation did not hold any feature as important.

3. **Multicriteria Performance Measure:** The proposed evaluation measure was applied, considering the individual performance measures computed, in order to estimate the quality of FS algorithms. These performance estimates were used as sorting criteria in the development of rankings, which display the FS algorithms in order of preference with respect to the data sets.

Step 3 – Construction of Data Metabases:

The results obtained in Step 2 were associated to compose 150 meta-examples (metadata), which were described by 21 morphological meta-features and one nominal target attribute.

In order to address the problem as binary classification, three metabases were derived, from the constructed metadata, whose values differed

[12] Timing performed in Core i7-3930k 3.20 GHz computer with 64 GB memory and operational system Ubuntu 14.10 under the same processing conditions for all measurements.

only in relation to the target attribute. In the first metabase (MB1), the class attribute took the labels "Ranking" and "FSS", elected considering the ordering of the FS algorithms. For example, for a particular data set the obtained performance ordering was CFS > ReliefF > InfoGain > CBF, therefore the characteristics extracted from these data were associated with the "FSS" label. This association occurred because CFS is on the top of the algorithm performance ordering, indicating that the evaluation by subsets is the best approach to select features in this data set. Similarly, in the second metabase (MB2) the attribute in question covered the labels "InfoGain" and "ReliefF", whereas the nominal target attribute tied to the third metabase (MB3) included the labels "CBF" and "CFS";

Step 4 – Induction and Use of the Recommendation Models: The recommendation models were induced on the three training metabases using the algorithm J48. The quality of the metamodels was estimated in terms of predictive performance using 10x10-fold cross-validation.

Figure 9 shows the activity diagram referring to the steps followed in the recommendation of the best FS algorithm to users. According to the information shown in this figure, given a new data set, the needed characteristics are extracted to form a meta-example of consultation. This meta-example is provided to the recommendation model generated from MB1, which suggests the most promising FS approach for the data set being analyzed. Then, according to the suggestion of the previous metamodel, the meta-example will be tested in another recommendation model. In this sense, if the suggested approach is "Ranking", the metamodel induced on the MB2 will recommend the best algorithm of ranking of features ("InfoGain" or "ReliefF"). Otherwise, if the suggested approach is "FSS", the metamodel, constructed from MB3, will recommend the best algorithm for the selection of a subset of features ("CBF" or "CFS").

7. Results and Discussion

As mentioned, for the 150 selected data sets, 21 meta-features were extracted, of which 14 are commonly used in the theme of classification algorithms recommendation, and the application of the remaining seven to Met-

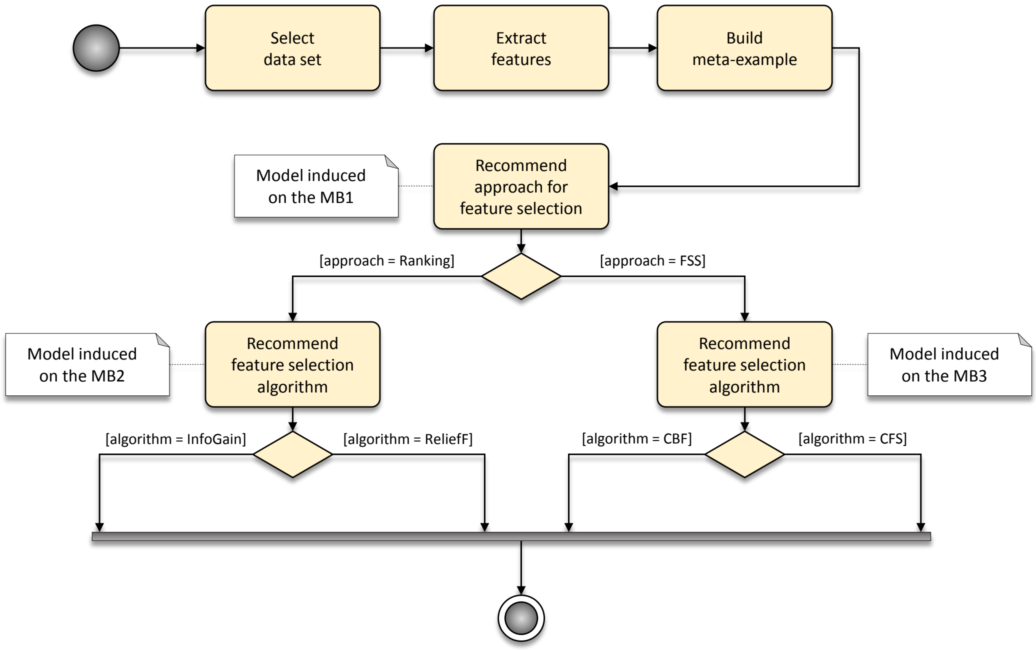
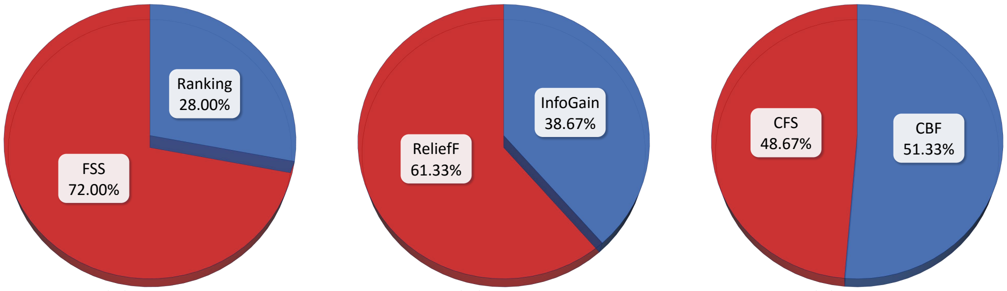


Figure 9: Activity diagram

alearning is proposed in this paper. Posteriorly, three training metabases were constructed as follows: each meta-example corresponded to one of the 150 data sets, and each feature corresponded to one of the 21 characteristics. Additionally, in each of these metabases, a nominal target attribute (or class attribute) was included. In the first metabase (MB1), the class attribute was labeled as “Ranking” or “FSS”, while in the second metabase (MB2), the attribute in question was labeled as “CBF” or “CFS”. As for the third metabase (MB3), the nominal target attribute was labeled as “InfoGain” or “ReliefF”. These labels indicate, for each investigated application domain, the most promising approach (MB1) and FS algorithm (MB2 and MB3) according to the multicriteria measure used to analyse the performance of FS algorithms.

There was a concern regarding the construction of data metabases in which the class attribute labels were nearly evenly distributed among the meta-examples. Such caution aims at minimizing the impact that the lack of representativeness of the minority group has on the differentiation of groups through the pattern extraction algorithm.

In Figure 10, for each constructed metabase, the distribution of labels, which compose the referred class attributes, is graphically represented.



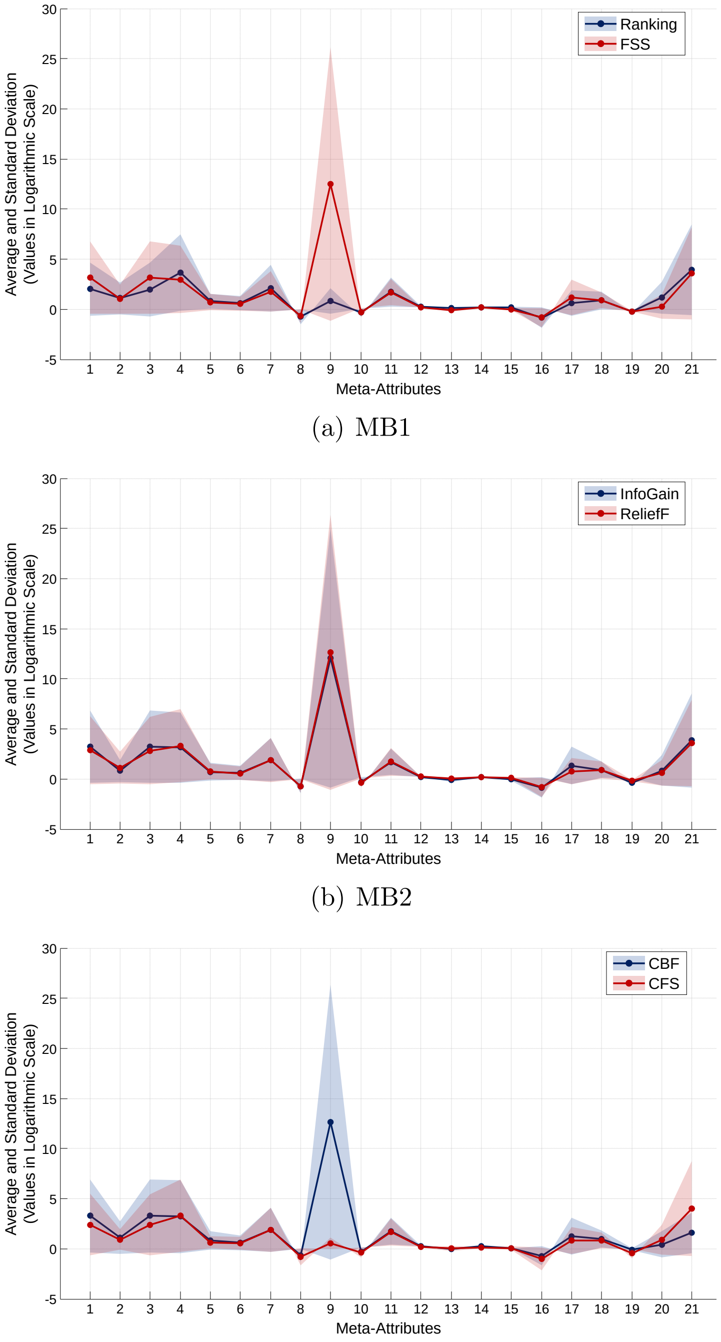
(a) MB1 (b) MB2 (c) MB3

Figure 10: Distribution of the nominal target attributes of the constructed metabases

According to Figure 10(a), the nominal target attribute of MB1 is formed by 28% of "Ranking" labels and 72% of "FSS" labels, with a majority error correspondent to 28% over the "FSS" group. In Figure 10(b), the nominal target attribute of MB2 is composed by 38.67% of "InfoGain" labels, 61.33% of "ReliefF" labels and a majority error of 38.67% on the top of "ReliefF" group. As for MB3, Figure 10(c) shows 48.67% of CFS labels, 51.33% of "CBF" labels and a majority error of 48.67% over the "CBF" group.

In Figure 11, for each training metabase MB1, MB2 and MB3, the average and the standard deviation values of the 21 extracted characteristics are presented in logarithmic scale. These characteristics are orderly disposed according to the identifiers exhibited in Table 7. In this figure, meta-features labeled as "Ranking", "InfoGain" and "CBF" are represented in blue while meta-features labeled as "FSS", "ReliefF" and "CFS" are shown in red.

As can be seen, for all three construted metabases (MB1, MB2 and MB3), 18 characteristics, out of the total of 21 meta-features, exhibit very close average and standard deviation values. The measure "average coefficient of variation of the attributes" (meta-attribute 9) is one that shows a large difference between values presented in MB1 and MB3. Although the value for "Ranking" is smaller than for "FSS" (MB1), this meta-feature seems to distinguish well "CBF" from "CFS" but not "InfoGain" from "ReliefF". The other two meta-features, "number of quantitative attributes" (meta-attribute 3) and "dispersion of the data set" (meta-attribute 21), also seem to not be uniform in MB1 and MB3, respectively. As for MB2, most of the meta-attributes values appear to be evenly distributed between "InfoGain" and "ReliefF".



(a) MB1

(b) MB2

(c) MB3

Figure 11: Average and standard deviation of the values for the extracted characteristics

In order to have a more accurate idea about the discrimination power of each meta-feature with regard to the the FS approach (Ranking or FSS, InfoGain or ReliefF, and CBF or CFS), we have applied hypothesis tests[13], with significance level of 5%, to find if there were any statistical significant difference between then. In the case the data presented Gaussian distribution, we applied the unpaired *t* test; otherwise, we applied the the Mann-Whitney test, as all the data were unpaired.

The analysis of the results showed statistically significant differences among the following meta-features for the following metabases:

- MB1 (Ranking or FSS): 4, 5, 13, 15, 20, 21;
- MB2 (InfoGain or ReliefF): 5, 15, 21;
- MB3 (CBF or CFS): 1, 2, 3, 12, 13, 16, 17, 19, 21.

Althought visually it is not possible to note a major difference between most of the meta-features, the statistical analysis showed some interesting results: meta-features 5 and 15 ("number of classes" and "average mutual information between classes and attributes") are both significant, seeming to discriminate well between MB1 and MB2, while meta-attribute 13 ("average entropy of the attributes") was outlined as discriminant for MB1 and MB3. It is also interesting to note that, in all three metabases, meta-feature 21 ("dispersion of the data set") has shown discriminative power. Such meta-features exhibit a considerable distance regarding the typical values, indicating that they may *a priori* provide aid to discriminate between FS approaches in the recommendation model.

In this first analysis, the aim was to verify the feasibility of finding intrinsic properties of the data sets that could lead to the correct indication of the approaches, ranking or FSS, and of the FS algorithms based on measures of consistency, dependence, information and distance. In this sense, it is important to highlight that there is no consensus regarding what would be the most appropriate dimension size of a metabase. However, the general rule is that the larger the proportion of meta-examples per characteristics, better must be the representation quality of the metabase.

[13] All statistical tests presented in this paper were performed using GraphPad InStat version 3.05 for Windows, http://www.graphpad.com.

In this context, we first generated recommendation metamodels from MB1, MB2 and MB3 using all the 21 meta-features.

Figure 12 shows the FS approaches recommendation model induced from MB1 and using algorithm J48. The decision nodes in this metamodel, such as in the other recommendation models presented in this section, correspond to the characteristics selected by J48 and are illustrated according to the identifiers shown in Table 7.

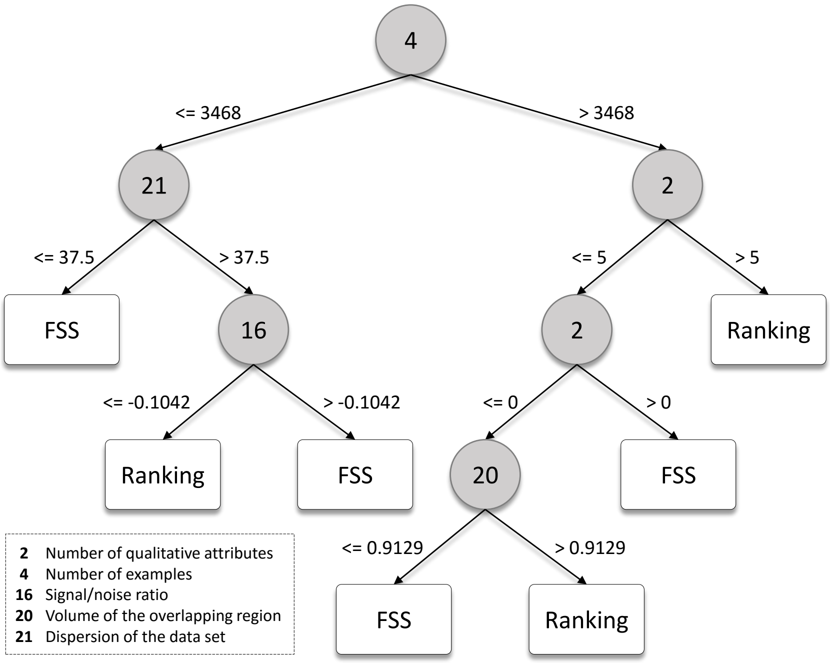


Figure 12: Symbolic metamodel of recommendation of approaches for FS

The features elected by J48 as decision nodes may be interpreted as relevant regarding the class and ordered according to the number of times they appeared in the rules generated from the tree, *i.e.*, it is related to the height they appear on the tree.

An interesting factor of J48 is that it constructs, in general, small trees, which are more understandable and robust to overfitting. This is explained by the fact that the data set is recursively reduced by the algorithm and, in some cases, several conjunctions that could be constructed are eliminated due to the regions in which the classes frequently overlap. Another important characteristic is that J48 can generate only one decision node for highly unbalanced data sets.

The model of approaches recommendation for FS outlined in Figure 12 presented seven classification rules. It is observed that out of a total of 21 characteristics, only five ("number of qualitative attributes", "number of examples", "signal/noise ratio", "volume of the overlapped region" and "dispersion of the data set") were chosen as decision nodes in the metamodel. Establishing as importance criteria the number of times that the characteristics appear in the induced rules, the most important meta-feature of this recommendation model was "number of qualitative attributes".

Regarding quality, the model of approaches recommendation for FS obtained a predictive performance average of 82.67%, with standard deviation of 10.52% for a Error of the Majority Class (EMC) of 28.00%. These results, summarized in Table 8, were estimated using 10x10-fold cross-validation.

Table 8: Statistics of the predictive performance of the metamodel induced over MB1

Statistics	**Predictive Performance (%)**
Minimum	66.67
Maximum	100.00
Average	82.67
Standard Deviation	10.52

Figure 13 presents the recommendation model of FS algorithms based on measures of information and distance, which was generated from MB2.

The recommendation model represented in Figure 13 included also seven classification rules. Out of a total of 21 meta-features, only six ("number of qualitative attributes", "number of classes", "average coefficient of variation of the attributes", "balancing of the data set", "volume of the overlapped region" and "dispersion of the data set") were selected as decision nodes in the metamodel, and none of those characteristics excelled the others.

It is interesting to note that in ReliefF, the noise influence in the data is mitigated through the contribution of the closest k neighbors of the same class as the currently considered example and of the closest k neighbors of each one of the different classes of the sampled example, instead of considering only one of the closest neighbors. This method searches for the closest examples using the Manhattan distance, and assigns weight to the features according to how well they distinguish examples from different classes. This process, such as in Relief, is also repeated m times. Usually, m is defined according to the number of examples present in the data set.

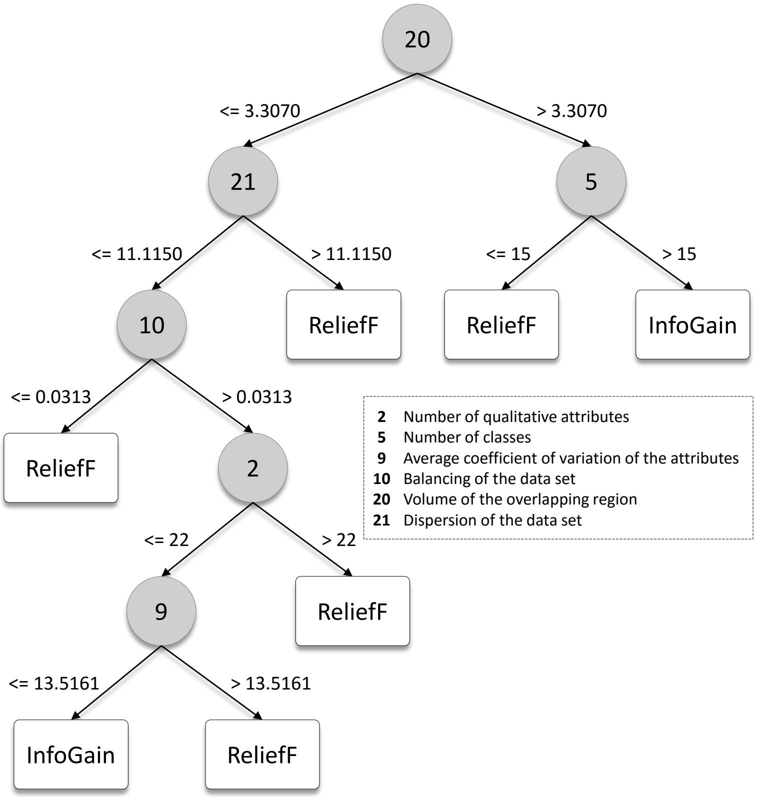

20
<= 3.3070
> 3.3070
21
5
<= 11.1150
> 11.1150
<= 15
> 15
10
ReliefF
ReliefF
InfoGain
<= 0.0313
> 0.0313
ReliefF
2
2 Number of qualitative attributes
5 Number of classes
9 Average coefficient of variation of the attributes
10 Balancing of the data set
20 Volume of the overlapping region
21 Dispersion of the data set
<= 22
> 22
9
ReliefF
<= 13.5161
> 13.5161
InfoGain
ReliefF

Figure 13: Symbolic metamodel of recommendation of FS algorithms based on information and distance measures

It is also observed in ReliefF that the ponderation of the contribution of close neighbors of different classes, is made according to the *a priori* probability $P(c)$ corresponding to the class c and the probabilistic treatment of incomplete data. The difference calculation, $diff$, is similar to the original Relief method, however, it considers the probability of two examples having different values for a certain feature.

Such as ReliefF, the InfoGain method also selects the features through the individual evaluation. Thus, it was necessary to establish a threshold in order to define the subset of features returned as selected by these algorithms. Besides, according to the information organized in Table 9, the recommendation model of FS algorithms based on information and distance measures had a predictive performance average of 78.00%, with standard deviation of 8.34%, for a EMC of 38.67%.

Table 9: Statistics of the predictive performance of the metamodel induced over the MB2

Statistics	**Predictive Performance (%)**
Minimum	66.67
Maximum	93.33
Average	78.00
Standard Deviation	8.34

Figure 14 exhibits the recommendation model for FSS algorithms induced over MB3. In this figure, the metamodel included 14 classification rules. Out of a total of 21 characteristics, ten ("number of attributes", "number of quantitative attributes", "number of examples", "average asymmetry of the attributes", "average correlation between attributes", "balancing of the data set", "signal/noise ratio", "equivalent number of attributes", "Fisher's discriminant" and "volume of the overlapped region") were elected as decision nodes of this model. Adopting over again as the importance criterion the number of times the characteristics appear in the induced rules, the most important features of this metamodel were: "number of attributes" and "signal/noise ratio".

Still, as visualized in Table 10, the model of recommendation of FS algorithms based on measures of dependence and consistency reached a predictive performance average of 76.00%, with standard deviation of 8.43%, for a EMC of 48.67%.

Table 10: Statistics of predictive performance of the metamodel induced over MB3

Statistics	**Predictive Performance (%)**
Minimum	66.67
Maximum	93.33
Average	76.00
Standard Deviation	8.43

After analyzing the metamodels employing all meta-features, we considered knowledge obtained from some of our previous work (Parmezan et al., 2012a,b) to orientate groups of meta-features to be used.

In Parmezan et al. (2012a) it was verified that the characterization of data through statistic measures has greater influence on determining the performance of FS algorithms than through measures of the simple category. Analogously, in Parmezan et al. (2012b), it was verified, besides the

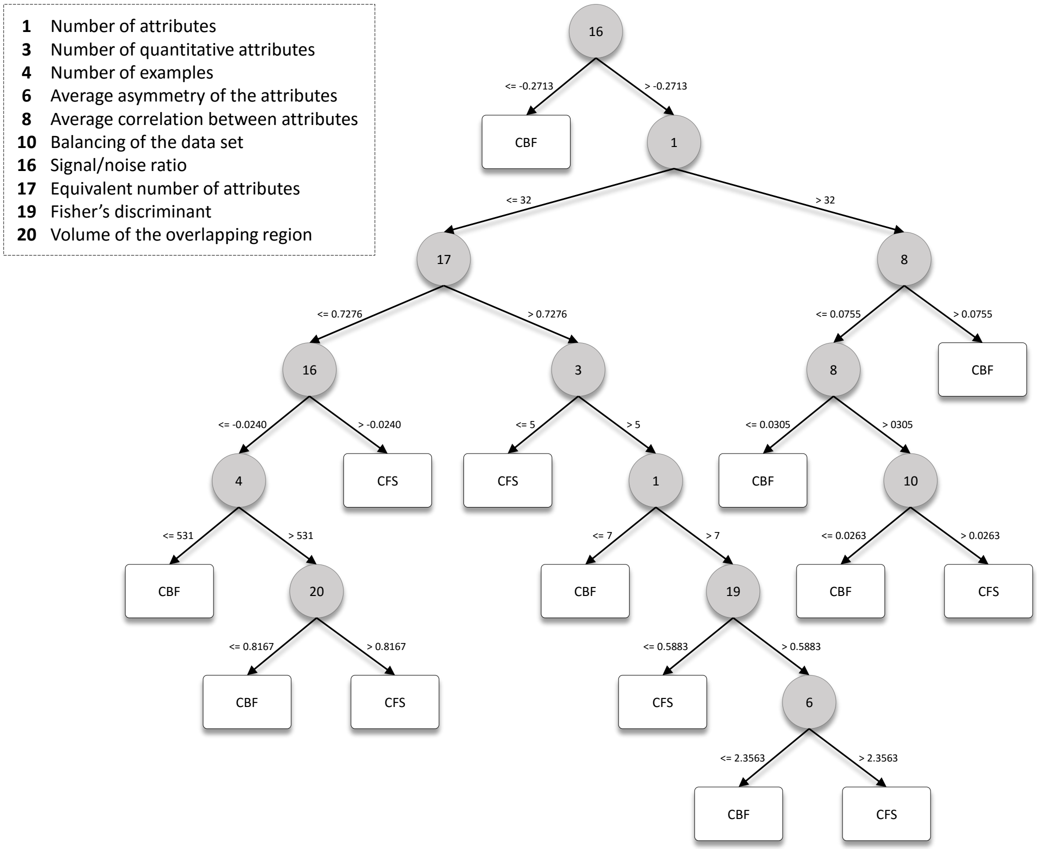


Figure 14: Symbolic metamodel of recommendation of FS algorithms based on dependence and consistency measures

importance of the use of statistic measures, the potentiality of the use of information measures in the data characterization. Thus, in this work also other six metamodels were generated, which were induced from the constructed metabases, but considering only the characteristics derived from the following two scenarios:

1. Statistics and information;
2. Statistics, information and complexity.

Figure 15 presents rules of the symbolic recommendation models of FS algorithms constructed considering these two scenarios. On one hand, the metamodels constructed over the two first metabases (MB1 and MB2) for Scenario 1 included five classification rules each. On the other hand, the model of recommendation generated in this scenario from MB3 presented 16 rules classification. The metamodels induced according to Scenario 2 comprehended four (MB1), seven (MB2) and fifteen (MB3) classification rules. Considering Scenario 1, the most frequent characteristic in the models of recommendation was "average mutual information between classes and attributes", while in Scenario 2 "dispersion of the data set" was the most frequent one.

```
average_entropy_attributes <= 1.5675: FSS
average_entropy_attributes > 1.5675
|   class_entropy <= 2.5850
|   |   average_mutual_information <= 2.1498
|   |   |   average_asymmetry <= 0.9728: FSS
|   |   |   average_asymmetry > 0.9728: Ranking
|   |   average_mutual_information > 2.1498: FSS
|   class_entropy > 2.5850: Ranking
```

(a) MB1 in Scenario 1

```
dispersion_dataset <= 37.5: FSS
dispersion_dataset > 37.5
|   majority_class_error <= 62.0281
|   |   average_conditional_entropy <= 0.8304: Ranking
|   |   average_conditional_entropy > 0.8304: FSS
|   majority_class_error > 62.0281: Ranking
```

(b) MB1 in Scenario 2

```
average_mutual_information <= 1.5569
|   balancing_dataset <= 0.0313
|   |   majority_class_error <= 70.9239: ReliefF
|   |   majority_class_error > 70.9239
|   |   |   average_correlation <= 0.0551: ReliefF
|   |   |   average_correlation > 0.0551: InfoGain
|   balancing_dataset > 0.0313: InfoGain
average_mutual_information > 1.5569: ReliefF
```

(c) MB2 in Scenario 1

```
volume_overlapping_region <= 3.3070
|   dispersion_dataset <= 11.1150
|   |   balancing_dataset <= 0.0313: ReliefF
|   |   balancing_dataset > 0.0313: InfoGain
|   dispersion_dataset > 11.1150: ReliefF
volume_overlapping_region > 3.3070
|   average_asymmetry <= 0.3536
|   |   average_kurtosis <= 2.6826: ReliefF
|   |   average_kurtosis > 2.6826
|   |   |   average_correlation <= 0.1378: ReliefF
|   |   |   average_correlation > 0.1378: InfoGain
|   average_asymmetry > 0.3536: ReliefF
```

(d) MB2 in Scenario 2

```
signal_noise_ratio <= -0.2713: CBF
signal_noise_ratio > -0.2713
|   signal_noise_ratio <= -0.0239
|   |   average_entropy_attributes <= 0.3810
|   |   |   class_entropy <= 1.2277: CBF
|   |   |   class_entropy > 1.2277
|   |   |   |   balancing_dataset <= 0.3235: CBF
|   |   |   |   balancing_dataset > 0.3235: CFS
|   |   average_entropy_attributes > 0.3810
|   |   |   balancing_dataset <= 0.9149
|   |   |   |   equivalente_number_attributes <= 0.7274
|   |   |   |   |   equivalente_number_attributes <= 0.6323: CFS
|   |   |   |   |   equivalente_number_attributes > 0.6323: CBF
|   |   |   |   equivalente_number_attributes > 0.7274
|   |   |   |   |   equivalente_number_attributes <= 1.0798: CFS
|   |   |   |   |   equivalente_number_attributes > 1.0798
|   |   |   |   |   |   average_entropy_attributes <= 0.7243
|   |   |   |   |   |   |   average_mutual_information <= 0.6202: CFS
|   |   |   |   |   |   |   average_mutual_information > 0.6202: CBF
|   |   |   |   |   |   average_entropy_attributes > 0.7243
|   |   |   |   |   |   |   average_kurtosis <= 6.4282
|   |   |   |   |   |   |   |   average_correlation <= 0.0841: CBF
|   |   |   |   |   |   |   |   average_correlation > 0.0841: CFS
|   |   |   |   |   |   |   average_kurtosis > 6.4282: CFS
|   |   |   balancing_dataset > 0.9149
|   |   |   |   average_asymmetry <= 1.3000: CBF
|   |   |   |   average_asymmetry > 1.3000
|   |   |   |   |   average_asymmetry <= 3.5181: CFS
|   |   |   |   |   average_asymmetry > 3.5181: CBF
|   signal_noise_ratio > -0.0239: CFS
```

(e) MB3 in Scenario 1

```
signal_noise_ratio <= -0.2713: CBF
signal_noise_ratio > -0.2713
|   dispersion_dataset <= 0.0644: CBF
|   dispersion_dataset > 0.0644
|   |   signal_noise_ratio <= -0.0239
|   |   |   fractal_dimension_dataset <= 14.2448
|   |   |   |   equivalente_number_attributes <= 0.7137
|   |   |   |   |   average_asymmetry <= 2.9330: CBF
|   |   |   |   |   average_asymmetry > 2.9330
|   |   |   |   |   |   average_asymmetry <= 16.2175: CFS
|   |   |   |   |   |   average_asymmetry > 16.2175: CBF
|   |   |   |   equivalente_number_attributes > 0.7137
|   |   |   |   |   average_entropy_attributes <= 0.7243
|   |   |   |   |   |   volume_overlapping_region <= 0.9129: CBF
|   |   |   |   |   |   volume_overlapping_region > 0.9129
|   |   |   |   |   |   |   average_entropy_attributes <= 0.7027: CFS
|   |   |   |   |   |   |   average_entropy_attributes > 0.7027: CBF
|   |   |   |   |   average_entropy_attributes > 0.7243
|   |   |   |   |   |   volume_overlapping_region <= 19.2429: CFS
|   |   |   |   |   |   volume_overlapping_region > 19.2429
|   |   |   |   |   |   |   fisher_discriminant <= 0.4968: CFS
|   |   |   |   |   |   |   fisher_discriminant > 0.4968: CBF
|   |   |   fractal_dimension_dataset > 14.2448
|   |   |   |   dispersion_dataset <= 70.1429: CBF
|   |   |   |   dispersion_dataset > 70.1429
|   |   |   |   |   average_coefficient_variation <= 3.2999: CFS
|   |   |   |   |   average_coefficient_variation > 3.2999: CBF
|   |   signal_noise_ratio > -0.0239: CFS
```

(f) MB3 in Scenario 2

Figure 15: Rules of symbolic metamodels of algorithms recommendation induced for Scenarios 1 and 2

Table 11 compares statistics regarding the predictive performance of the recommendation models designed from the examined scenarios in comparison with the models induced over the original metabases (all the features). In this table, the metamodel generated over MB1 considering only the 12 characteristics, bounded to statistics and information, presented degradation of the predictive performance when compared to the model of recommendation constructed using the original MB1 (21 characteristics). In contrast, the prediction quality of these metamodels was surpassed by the recommendation model constructed from the same 12 characteristics, plus four meta-features of the complexity category, totalling 16 morphologic characteristics.

Table 11: Statistics of predictive performance of the induced metamodels for the Scenarios 1 and 2

	Predictive Performance (%)								
	All the Features			**Scenarios 1**			**Scenarios 2**		
Statistics	**MB1**	**MB2**	**MB3**	**MB1**	**MB2**	**MB3**	**MB1**	**MB2**	**MB3**
Minimum	66.67	66.67	66.67	60.00	53.33	60.00	66.67	60.00	66.67
Maximum	100.00	93.33	93.33	86.67	73.33	93.33	100.00	86.67	100.00
Average	82.67	78.00	76.00	72.00	63.33	72.00	87.33	72.00	90.63
Standard Deviation	10.52	8.34	8.43	8.78	6.48	10.33	11.09	9.33	7.17

The recommendation models induced over MB2, using only meta-features covered by the examined scenarios, did not exhibit improvement in the predictive performance when confronted with the metamodel generated from the original MB2. In the same way, the recommendation model constructed over MB3 considering the first scenario did not present improvement in the predictive performance regarding the metamodel induced from the original MB3. However, when analyzing the second scenario, the recommendation model constructed using characteristics derived from the statistics, information and complexity categories surpassed the performance of the metamodels induced over the original MB3 and considering the first scenario.

In order to verify the existence of statistically significant differences among the performances of the constructed metamodels, we applied the nonparametric statistical test Kruskal-Wallis for unpaired groups, with significance level of 5%, followed by the Dunn post-hoc test in the cases in which significant difference was found. As a result, it was verified that the metamodels generated from Scenario 1 did not exhibit a statistically significant performance improvement when compared to the metamodels induced over the respective original metabases. The same occured when comparing Scenario 2 and the original metabases. However, when confronting the models resulting from Scenario 1 with the ones from Scenario 2, it was possible to note that the recommendation model constructed over MB3 considering Scenario 2 presented a performance improvement compared to the metamodel generated from the same metabase, but considering Scenario 1 ($p < 0.01$).

The results obtained applying the statistical significance test lead to the conclusion that, for the problem of recommendation of FS algorithms, the characterization of data through complexity measures can, together with measures of statistics and information, enable a good indication of the FS algorithms (predictive performance of 90.63% with standard deviation of 7.17%).

To demonstrate why the decision trees are good candidates for metalearners, we face J48, used in this paper, with six other ML algorithms. Most of the methods selected for the comparison are popular in the Metalearning theme (Lemke et al., 2015), being ADTree and CART of symbolic paradigm, kNN of instance-based paradigm, MLP of connectionist paradigm, and Naïve Bayes and SVM of statistical paradigm. The metalearners were applied to the three built metabases (MB1, MB2, and MB3) considering both the original data (all the features) and the other two scenarios discussed above. The induced metamodels were evaluated using 10x10 fold-cross-validation and their predictive performance averages, with the respective standard deviations in parentheses, are listed in Table 12.

The Kruskal-Wallis statistical test and Dunn's posthoc test were again employed to verify the existence of statistically significant differences among the average performances of the metamodels induced by the seven metalearners in Table 12. Individually analyzing the ML algorithms, CART to MB3 in Scenario 2 showed significant degradation of predictive performance against the original metadata (all the features). In contrast, as we discussed in Table 11, J48 to MB3 in Scenario 2 presented statistically significant performance improvement compared to the symbolic metamodel generated from the same metabase in Scenario 1.

Considering all metalearners in the table and their respective metamodels built on the original metabases, the Naïve Bayes for MB1 showed significant predictive performance degradation against the algorithms ADTree, CART, J48, and kNN. Looking at MB2, J48 outperformed algorithms kNN and Naïve Bayes with statistically significant difference. Concerning the recom-

Table 12: Predictive performance averages, and standard deviations in parentheses, of metamodels generated by different metalearners

	Predictive Performance (%)		
	All the Features		
Metalearner	**MB1**	**MB2**	**MB3**
ADTree	80.67 (10.63)	68.67 (9.45)	72.67 (9.14)
CART	80.67 (13.13)	73.33 (4.44)	73.33 (14.74)
J48	82.67 (10.52)	78.00 (8.34)	76.00 (8.43)
*k*NN	75.33 (8.34)	60.00 (9.94)	70.67 (10.52)
MLP	71.33 (10.45)	62.67 (7.83)	74.67 (12.88)
Naïve Bayes	40.67 (11.72)	59.33 (9.17)	71.33 (8.46)
SVM	73.33 (3.14)	61.33 (4.22)	72.00 (12.09)
	Scenarios 1		
Metalearner	**MB1**	**MB2**	**MB3**
ADTree	66.67 (12.17)	54.00 (15.54)	72.00 (11.24)
CART	70.00 (7.86)	60.67 (4.92)	68.00 (15.01)
J48	72.00 (8.78)	63.33 (6.48)	72.00 (10.33)
*k*NN	71.33 (8.34)	60.00 (7.70)	64.67 (12.19)
MLP	68.67 (10.45)	62.00 (14.07)	69.33 (11.84)
Naïve Bayes	34.67 (7.18)	54.00 (10.09)	62.67 (13.40)
SVM	72.67 (3.78)	61.33 (2.81)	69.33 (11.84)
	Scenarios 2		
Metalearner	**MB1**	**MB2**	**MB3**
ADTree	80.00 (9.43)	66.00 (10.63)	83.75 (19.59)
CART	82.00 (12.98)	69.33 (8.43)	48.75 (3.95)
J48	87.33 (11.09)	72.00 (9.33)	90.63 (7.17)
*k*NN	77.33 (8.43)	62.00 (10.91)	55.00 (14.67)
MLP	67.33 (12.75)	63.33 (8.46)	61.25 (14.97)
Naïve Bayes	42.00 (13.68)	58.00 (7.33)	57.50 (13.92)
SVM	72.67 (3.78)	61.33 (2.81)	60.00 (9.86)

mendation models generated in Scenarios 1, Naïve Bayes for MB1 showed significantly worse performance when compared with the algorithms CART, J48, and *k*NN. Analyzing the induced metamodels in Scenarios 2, Naïve Bayes for MB1 obtained the worst result, with statistically significant differences, faced to the algorithms ADTree, CART, J48, and *k*NN. In this same point of view, ADTree for MB3 showed significant performance improvement when compared with the algorithms CART, *k*NN and Naïve Bayes, the same happened with J48 concerning the algorithms CART, kNN, MLP, Naïve Bayes, and SVM. In summary, the best results have been achieved through decision trees (J48 followed by ADTree model), while the worst performers were provided by the statistical learning algorithm Naïve Bayes.

As for the properties of the symbolic recommendation models of FSS algorithms constructed in this paper and in previous works, Table 13 synthesizes the main ones.

Table 13: Comparison between symbolic recommendation models of FSS algorithms constructed in this paper and the ones obtained in previous studies

Model	**#E**	**#A**	**Measures**	**#R**	**Frequent Features**	**EMC (%)**	**Predictive Performance (%)**
MRA	30	10	Simple and statistics	5	"Average asymmetry of the attributes"	43.33	56.33(27.09)
MRB	17	10	Statistics and information	3	"Average entropy of the attributes"	47.06	59.34
MR1	150	21	Simple, statistics, information and complexity	14	"Number of attributes" and "signal/noise ratio"	48.67	76.00(8.43)
MR2	150	12	Statistics and information	16	"Equivalent number of attributes"	48.67	72.00(10.33)
MR3	150	16	Statistics, information and complexity	15	"Average asymmetry of the attributes", "signal/noise ratio", "volume of the overlapping region" and "dispersion of the data set"	48.67	90.63(7.17)

The information presented in Table 13 is organized as follows: in the first column the metamodels are named, so that the MRA corresponds to the metamodel analyzed in Parmezan et al. (2012a), MRB represents the recommendation model evaluated in Parmezan et al. (2012b), MR1 indicates the metamodel induced from the 21 meta-features of MB3, MR2 includes the recommendation model generated over MB3 considering Scenario 1 and MR3 expresses the metamodel induced over the same metabase, but regarding Scenario 2. The second and third columns show, respectively, the number of data sets (#E) and of attributes (#A) used in the construction of the training metabases. The fourth column shows the categories whose measures were used in the step of data sets characterization. The fifth and sixth columns expose, in this order, the number of rules (#R) contemplated by the induced metamodels and the most important meta-feature (or meta-features in the case of a tie) according to the frequency with which it appears as decision nodes in the induced rules. At last, the EMC is presented in the seventh column, while the eighth column shows the average of the predictive performance followed, between parenthesis, by the respective standard deviation.

In Table 13, out of the five considered metabases, four included meta-features belonging to the categories of statistics and information measures. Among all the constructed models, three included over 14 classification rules. In this sense, the most important meta-features amongst all the models were the ones extracted from "average asymmetry of the attributes" and "signal/noise ratio".

The predictive performances of these recommendation models were estimated using stratified 10-fold cross-validation, except for the MRB in which was used the holdout validation.

The average and the standard deviation of the predictive performance of the metamodel MR3 was 90.63% and 7.17%, respectively. In other words, the MR3 exhibited an approximate error of 9.37% for a EMC of 48.67%, proving to be the best algorithms recommendation model for the selection of subset of features, considering all the data sets and studied algorithms. Differently, the MRA presented a predictive performance average of 56.33% with standard deviation of 27.09%, *i.e.*, it was obtained an approximate error of 43.66%, close to the one of the majority class (43.33%). It means that, besides the fact that the metamodel MRA presented the worst predictive performance, its use is not preferable regarding the trivial or ingenuous classification, which always classifies a new example as belonging to the majority class.

As mentioned in Section 4, the objective of this paper overlaps the one reported in Filchenkov and Pendryak (2015), Shilbayeh and Vadera (2014), and Wang et al. (2013). All these pieces of work aim at effectively recommending algorithms for the FS task. In order to do so, they use an architecture of recommendation algorithms based on the use of Metalearning. Observing the intersections of these works, it is possible to highlight the following advantages, drawbacks and limitations:

- In Wang et al. (2013) it was developed a lazy method which identifies the k databases more similar to a new data set and estimates, using a multicriteria metric, the performance of every FS algorithm. Afterwards, the r FS algorithms which are considered the most appropriate are ordered according to their respective performances. The main drawback of the similarity-based method is its setup, which requires two parameters, k and r. Besides being dependent on each problem, the identification of k's value can present a high computational cost;

- An intuitive Metalearning framework was proposed in Shilbayeh and Vadera (2014) with the intention to learn about which FS algorithms work best for a given data set. The framework was implemented using the Weka toolkit and involves the same steps that a trivial recommender system on meta-level. A limitation of this approach is that it requires the user to know beforehand which ML method will be used to extract patterns from the subset of features selected by the recommended algorithm;

- Filchenkov and Pendryak (2015) presented an approach for engineering meta-features in the field of FS algorithms recommendation. Beyond the usual meta-features in Metalearning, the authors investigated the use of characterization measures based on models. The experimental protocol was formulated in agreement with the work of Wang et al. (2013). In their results, a set of meta-features shows to be useful in predicting FS algorithms. However, the high number of meta-features used in the paper is the main drawback of the proposal since it is possible to achieve similar or better results from a reduced, and yet of quality, set of characterization measures.

In this work, we propose an architecture of recommendation of FS algorithms based on the idea of providing rankings as a method to construct suggestions.

For that, several decision trees were induced over three training metabases, aiming at understanding in which circumstances an FS algorithm surpasses the other and which aspects of the data have greater influence in determining the performance of these algorithms. This approach, besides resulting in interpretable and robust to overfitting models of recommendation, is less computationally costly than any recently proposed method.

Although we have used decision trees on our experiments for the mentioned reasons, our arquiteture is very flexible as it is not tied to any model, as some other proposals are (Shilbayeh and Vadera, 2014; Wang et al., 2013). If the user wishes, he or she may use any learning algorithm to construct the recommendation model.

Despite these advantages, our architecture is limited to the binary classification problem. While other frameworks only work with one training metabase, our architecture decomposes the original problem into subproblems, each one with two classes, which are organized in different metabases. On one hand, these derivations facilitate meta-knowledge management. On the other hand, it ends up increasing the set of investigated FS algorithms.

Another important issue to be highlighted refers to the proposed multicriteria performance measure. This is an interesting and powerful tool as it combines any three measures in a single one allowing more complete analysis. In our experimental evaluation, we combined three of the most important performance issues in FS – error rate, percentage of selected features by the FS algorithms and learning time – in order to recommend the FS algorithms.

8. Conclusion

In the process of Data Mining, the adequate choice of Feature Selection algorithms may potentialize the quality of the data supplied as entry for classification algorithms. Considering the experimental aspect of the area, on one hand, the most common procedure for guiding this choice is, given a data set, applying several algorithms and criteria to select, with each of them, a subset of features in order to verify with which of these subsets it is possible to induce the best classifier. On the other hand, this choice may also be conducted based on the knowledge of the expert, normally detained by professionals of the application domain and computational areas. In both cases, it is known that the process of choosing Feature Selection algorithms, besides demanding an extensive technical knowledge, may consume a long period of time. In virtue of that, it becomes necessary to propose and use mechanisms in order to facilitate this choice and simultaneously ensure the effectiveness of the selected algorithm.

The Metalearning field is still little explored in the Feature Selection theme, but it may aid in the recommendation of suitable algorithms for this task. It involves research on the proposition and application of methods capable of supplying the user with an automatic and systematic aid for choosing algorithms.

In this paper, based on concepts of Metalearning, a new architecture for recommendation of Feature Selection algorithms was developed and evaluated. Such framework associates morphological properties of the data sets, extracted through characterization measures, with the performance of the Feature Selection algorithms, estimated from the induction of computational models using the selected features. A major advantage of our architecture compared to other proposals in the literature is its flexibility, which allows the components of each step to be modified according to the user's local requirements. In addition, it combines distinct concepts of intelligent and expert systems to promote a solution that, with the aid of leading-edge technologies, facilitates the integration of our architecture into any information system.

A novel Multicriteria Performance Measure was also proposed in the present paper. This measure combines any three measurements on a single one, creating an interesting and powerful tool to evaluate not only Feature Selection algorithms but in any context where is necessary a combination, whether to maximize the measure or minimize it. We have used the Multicriteria Performance Measure in our empirical evaluation to combine the error rate, the learning time and the percentage of selected features by the FS algorithm, which are the most important individual measures, often considered when treating FS problems.

The present study aimed at recommending algorithms based on information, distance, dependence and consistency measures to aid in the selection of subsets of features. In order to do so, a metabase was constructed using 21 measures extracted from characteristics of 150 data sets.

The symbolic metamodels induced over the metabases indicate that the data characterization through statistics, information and complexity measures is promising for the representation of the problem in question. It was also verified that, for most constructed models of recommendation, the metafeatures derived from "ratio signal/noise", "dispersion of the data set" and "average mutual information between classes and attributes" were selected as decision nodes in the metamodels. This means that the referred characterization measures have a high discriminant power over the others, being considered good candidates in the representation of data sets for the automatic recommendation problem of Feature Selection algorithms.

The proposed architecture, besides resulting in understandable and robust to overfitting models, is less computationally costly than approaches recently conceived in literature and linked to other Machine Learning paradigms.

The overview of results showed that it is possible to obtain good results in the recommendation of algorithms for the Feature Selection, with accuracy higher than 90%, using symbolic classifiers. The approach of choosing J48 as the inducer for our recommendation model was also quantitatively superior, *i.e.*, in terms of predictive performance, it was equivalent or better, in all comparisons, than all the other popular metalearners.

Despite all the significant advantages, our architecture is limited to the binary classification problem. Thus, the framework needs to decompose the original problem into subproblems, each one with two classes, which are organized in different metabases. On one hand, these derivations facilitate meta-knowledge management. On the other hand, the derivations end up increasing the set of investigated FS algorithms, while other frameworks avoid this problem by working only with one training metabase.

Future works include the evaluation of other methods for the construction of suggestions, such as ranking generation, and for the induction of metamodels, as well as the analysis of other multicriteria measures for the evaluation of the performance of Feature Selection algorithms. Simultaneously, it is desirable to supply the user with a visual aid in the recommendation of these algorithms using, for example, a model proposed in Lee (2005) which categorizes the Feature Selection algorithms regarding the predictive performance and the percentage of selected features.

9. Acknowledgements

Authors would like to thank for the support of the Brazilian National Counsel of Technological and Scientific Development and the Araucaria's Foundation for the Support of scientific and Technological Development of Paraná.

References

Alpaydin, E., 2004. Introduction to machine learning. The MIT Press, Cambridge, England.

Bache, K., Lichman, M., 2013. UCI machine learning repository.
URL https://goo.gl/nasfSC

Bensusan, H., Giraud-Carrier, C., 2000. Discovering task neighbourhoods through landmark learning performances. In: Principles of Data Mining and Knowledge Discovery. Vol. 1910 of Lecture Notes in Computer Science. Springer, pp. 325–330.

Bhatt, N., Thakkar, A., Ganatra, A., 2012. A survey & current research challenges in meta learning approaches based on dataset characteristics. International Journal of soft computing and Engineering 2 (10), 234–247.

Bhatt, N., Thakkar, A., Ganatra, A., Bhatt, N., 2013. Ranking of classifiers based on dataset characteristics using active meta learning. International Journal of Computer Applications 69 (20).

Brazdil, P. B., Giraud-Carrier, C., Soares, C., Vilalta, R., 2009. Metalearning: applications to data mining. Springer-Verlag, Berlin, Germany.

Breiman, L., Friedman, J., Stone, C., Olshen, R., 1984. Classification and regression trees. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, California, United States of America.

Covões, T. F., 2010. Seleção de atributos via agrupamento. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil.
URL https://goo.gl/y68g7X

Cui, C., Hu, M., Weir, J. D., Wu, T., 2016. A recommendation system for meta-modeling: a meta-learning based approach. Expert Systems with Applications 46, 33–44.

Das, S., 2001. Filters, wrappers and a boosting based hybrid for feature selection. In: International Conference on Machine Learning. Williams College, pp. 74–81.

Dash, M., Liu, H., 2003. Consistency-based search in feature selection. Artificial Intelligence 151 (1-2), 155–176.

Deitel, P. J., Deitel, H. M., 2014. Java how to program, 10th Edition. Pearson, Cambridge, England.

Fayyad, U., Piatetsky-shapiro, G., Smyth, P., 1996. From data mining to knowledge discovery in databases. AI Magazine 17, 37–54.

Fayyad, U. M., Irani, K. B., 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In: Thirteenth International Joint Conference on Articial Intelligence. Vol. 2. Morgan Kaufmann Publishers, pp. 1022–1027.

Filchenkov, A., Pendryak, A., 2015. Datasets meta-feature description for recommending feature selection algorithm. AINL-ISMW FRUCT, 11–18.

Fix, E., Hodges, J. L., 1951. Discriminatory analysis, nonparametric discrimination: consistency properties. US Air Force School of Aviation Medicine Technical Report 4 (3), 477.

Freund, Y., Mason, L., 1999. The alternating decision tree learning algorithm. In: Proceedings of the Sixteenth International Conference on Machine Learning. Morgan Kaufmann, San Francisco, United States of America, pp. 124–133.

Giraud-Carrier, C., 2005. The data mining advisor: meta-learning at the service of practitioners. In: International Conference on Machine Learning and Applications. IEEE Computer Society, Washington, United States of America, pp. 113–119.

Gomes, T. A., Prudêncio, R. B., Soares, C., Rossi, A. L., Carvalho, A., 2012. Combining meta-learning and search techniques to select parameters for support vector machines. Neurocomputing 75 (1), 3–13.

Hall, M., 1999. Correlation-based feature subset selection for machine learning. Ph.D. thesis, Department of Computer Science, University of Waikato, Waikato, New Zealand.

Hall, M. A., 2000. Correlation-based feature selection for discrete and numeric class machine learning. In: International Conference on Machine Learning. pp. 359–366.

Han, J., Kamber, M., Pei, J., 2011. Data mining: concepts and techniques, 3rd Edition. Morgan Kaufmann, California, United States of America.

Haykin, S. S., 2009. Neural networks and learning machines, 3rd Edition. Prentice Hall, Upper Saddle River, United States of America.

Ho, T., Basu, M., Law, M., 2006. Measures of geometrical complexity in classification problems. In: Data complexity in pattern recognition. Advanced Information and Knowledge Processing. Springer-Verlag, London, pp. 1–23.

Kira, K., Rendell, L. A., 1992. A practical approach to feature selection. In: Proceedings of the Ninth International Workshop on Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, United States of America, pp. 249–256.

Kohavi, R., John, G. H., 1997. Wrappers for feature subset selection. Artificial Intelligence 97 (1-2), 273–324.

Kononenko, I., 1994. Estimating attributes: analysis and extention of Relief. In: European Conference on Machine Learning. Springer-Verlag, Amsterdam, Kingdom of the Netherlands, pp. 171–182.

Kück, M., Crone, S. F., Freitag, M., 2016. Meta-learning with neural networks and landmarking for forecasting model selection - an empirical evaluation of different feature sets applied to industry data. In: International Joint Conference on Neural Networks. IEEE, pp. 1–8.

Lee, H. D., 2005. Seleção de atributos importantes para a extração de conhecimento de bases de dados, Tese (Doutorado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil.
URL https://goo.gl/kFyPOg

Lee, H. D., Monard, M. C., 2003. Seleção de atributos para algoritmos de aprendizado de máquina supervisionado utilizando como filtro a dimensão fractal. Revista de La Sociedad Chilena de Ciencia de La Computación 4 (1), 1–8.
URL https://goo.gl/TpXMii

Lee, H. D., Monard, M. C., Wu, F. C., 2006. A fractal dimension based filter algorithm to select features for supervised learning. Lecture Notes in Computer Science 4140, 278–288.

Lee, J.-S., Olafsson, S., 2013. A meta-learning approach for determining the number of clusters with consideration of nearest neighbors. Information Sciences 232, 208–224.

Lemke, C., Budka, M., Gabrys, B., 2015. Metalearning: a survey of trends and technologies. Artificial Intelligence Review 44 (1), 117–130.

Lemke, C., Gabrys, B., 2010. Meta-learning for time series forecasting and forecast combination. Neurocomputing 73 (1012), 2006–2016.

Lindner, G., Studer, R., 1999. AST: support for algorithm selection with a CBR approach. In: Principles of Data Mining and Knowledge Discovery. Vol. 1704 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 418–423.

Liu, H., Motoda, H., 2008. Computational methods of feature selection. Chapman & Hall/CRC data mining and knowledge discovery, Minnesota, United States of America.

Liu, H., Motoda, H., 2013. Feature selection for knowledge discovery and data mining. The Springer International Series in Engineering and Computer Science. Springer, United States of America.

Liu, H., Motoda, H., Yu, L., 2004. A selective sampling approach to active feature selection. Artificial Intelligence 159 (12), 49–74.

Luo, G., 2016. PredicT-ML: A tool for automating machine learning model building with big clinical data. Health Information Science and Systems 4 (5), 1–16.

Maudsley, D., 1979. A theory of meta-learning and principles of facilitation: an organismic perspective. Thesis (Ed.D.) – University of Toronto.

Michie, D., Spiegelhalter, D. J., Taylor, C. C., Campbell, J., 1994. Machine learning, neural and statistical classification. Ellis Horwood, Upper Saddle River, United States of America.

Mitra, P., Murthy, C. A., Pal, S. K., 2002. Unsupervised feature selection using feature similarity. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (3), 301–312.

Molina, L., Belanche, L., Nebot, A., 2002. Feature selection algorithms: a survey and experimental evaluation. In: International Conference on Data Mining. pp. 306–313.

Muñoz, L. A. B., González-Navarro, F. F., 2011. Review and evaluation of feature selection algorithms in synthetic problems. CoRR.

Nogueira, B. M., 2009. Avaliação de métodos não-supervisionados de seleção de atributos para mineração de textos. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil.
URL https://goo.gl/7QjO2M

Parmezan, A. R. S., Lee, H. D., Wu, F. C., 2011a. Redução da dimensionalidade em bases de dados naturais através de métodos de filtro para seleção de atributos importantes. In: XIX Simpósio Internacional de Iniciação Científica da Universidade de São Paulo, São Carlos, Brasil. pp. 1–1.
URL https://goo.gl/Kj0En4

Parmezan, A. R. S., Lee, H. D., Wu, F. C., 2012a. Estudo preliminar da construção de um modelo de recomendação de algoritmos de seleção de atributos utilizando meta-aprendizado. In: XX Simpósio Internacional de Iniciação Científica da Universidade de São Paulo, São Paulo, Brasil. pp. 1–1.
URL https://goo.gl/gY0LnY

Parmezan, A. R. S., Wu, F. C., Lee, H. D., 2011b. Estudo de medidas de importância e algoritmos para seleção de atributos para mineração de dados. In: XX Encontro Anual de Iniciação Científica, Ponta Grossa, Brasil. pp. 1–4.
URL https://goo.gl/C6Krfr

Parmezan, A. R. S., Wu, F. C., Lee, H. D., 2012b. Meta-aprendizado no auxílio à seleção de atributos: um estudo para medidas de correlação e consistência. In: XXI Encontro Anual de Iniciação Científica, Maringá, Brasil. pp. 1–4.
URL https://goo.gl/R0Ds5I

Peng, Y., Flach, P. A., Soares, C., Brazdil, P., 2002. Improved dataset characterisation for meta-learning. In: Discovery Science. Vol. 2534 of Lecture Notes in Computer Science. Springer, pp. 141–152.

Pfahringer, B., Bensusan, H., Giraud-Carrier, C., 2000. Meta-learning by landmarking various learning algorithms. In: International Conference on Machine Learning. Morgan Kaufmann, pp. 743–750.

Press, W. H., Flannery, B. P., Teukolsky, S. A., Vetterling, W. T., 1992. Numerical recipes in C: the art of scientific computing, 2nd Edition. Cambridge University Press, Cambridge, England.

Prudencio, R., Ludermir, T., 2004. Using machine learning techniques to combine forecasting methods. In: Lecture Notes in Artificial Intelligence. pp. 1122–1127.

Purwar, A., Singh, S. K., 2015. Hybrid prediction model with missing value imputation for medical data. Expert Systems with Applications 42 (13), 5621–5631.

Pyle, D., 1999. Data preparation for data mining. Morgan Kaufmann, California, United States of America.

Reif, M., Shafait, F., Goldstein, M., Breuel, T., Dengel, A., 2014. Automatic classifier selection for non-experts. Pattern Analysis and Applications 17 (1), 83–96.

Rice, J. R., 1976. The algorithm selection problem. Advances in Computers 15, 65–118.

Shilbayeh, S., Vadera, S., 2014. Feature selection in meta learning framework. In: Science and Information Conference. IEEE, pp. 269–275.

Smith-Miles, K. A., 2009. Cross-disciplinary perspectives on meta-learning for algorithm selection. ACM Computing Surveys 41 (1), 1–25.

Sohn, S. Y., 1999. Meta analysis of classification algorithms for pattern recognition. Pattern Analysis and Machine Intelligence 21 (11), 1137–1144.

Spolaôr, N., Cherman, E. A., Monard, M. C., 2011. Uso do ReliefF para seleção de atributos em dados multirrótulo. In: Conferência Latinoamericana de Informática. pp. 960–975.
URL https://goo.gl/Llo19y

Stiglic, G., Bajgot, M., Kokol, P., 2010. Gene set enrichment meta-learning analysis: next-generation sequencing versus microarrays. BMC Bioinformatics 11 (1), 176–185.

Traina, C., Traina, A. J. M., Faloutsos, C., 2003. MDE – measure distance exponent manual. (Internal Document).

Vapnik, V. N., 1999. The nature of statistical learning theory, 2nd Edition. Information Science and Statistics. Springer Science & Business Media, New York, United States of America.

Vilalta, R., Drissi, Y., 2002. A perspective view and survey of meta-learning. Artificial Intelligence Review 18 (2), 77–95.

Vukicevic, M., Radovanovic, S., Delibasic, B., Suknovic, M., 2016. Extending meta-learning framework for clustering gene expression data with component-based algorithm design and internal evaluation measures. International Journal of Data Mining and Bioinformatics 14 (2), 101–119.

Wang, G., Song, Q., Sun, H., Zhang, X., Xu, B., Zhou, Y., 2013. A feature subset selection algorithm automatic recommendation method. Journal of Artificial Intelligence Research 47, 1–34.

Wang, X., Smith-Miles, K., Hyndman, R., 2009. Rule induction for forecasting method selection: meta-learning the characteristics of univariate time series. Neurocomputing 72 (1012), 2581–2594.

Witten, I. H., Frank, E., Hall, M. A., 2011. Data mining: practical machine learning tools and techniques, 3rd Edition. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, Amsterdam, Kingdom of the Netherlands.

Yu, L., Liu, H., 2004. Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research 5, 1205–1224.

Zabashta, A., Smetannikov, I., Filchenkov, A., 2015. Study on meta-learning approach application in rank aggregation algorithm selection. In: International Workshop on Meta-Learning and Algorithm Selection co-located with ECMLPKDD 2015. pp. 115–116.

Zabashta, A., Smetannikov, I., Filchenkov, A., 2016. Rank aggregation algorithm selection meets feature selection. Springer International Publishing, Cham, Switzerland, pp. 740–755.

Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., Liu, H., 2010. Advancing feature selection research – ASU feature selection repository. Tech. rep., School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, United States of America.