PERGAMON

Pattern Recognition 35 (2002) 835–846

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

A hybrid filter/wrapper approach of feature selection using information theory

Marc Sebban[a], Richard Nock[b,*]

[a] *Département de Sciences Juridiques et Economiques, Université Antilles-Guyane, Campus de Fouillole, 97159 Pointe à Pitre, Guadeloupe, France*

[b] *Département Scientifique Interfacultaire, Université Antilles-Guyane, Campus de Schoelcher, 97233 Schoelcher, Martinique, France*

Received 4 November 1999; accepted 12 April 2001

Abstract

We focus on a hybrid approach of feature selection. We begin our analysis with a *filter model*, exploiting the geometrical information contained in the minimum spanning tree (MST) built on the learning set. This model exploits a statistical test of *relative certainty gain*, used in a forward selection algorithm. In the second part of the paper, we show that the MST can be replaced by the 1 nearest-neighbor graph without challenging the statistical framework. This leads to a feature selection algorithm belonging to a new category of *hybrid models* (*filter-wrapper*). Experimental results on readily available synthetic and natural domains are presented and discussed.

Keywords: Machine learning; Data mining; Information theory; Feature selection; Wrapper models; Filter models

1. Introduction

The problem of feature (attribute, or variable) selection, i.e. the selection of relevant description variables in the data, has historically been a prerogative of statistical research. It is only recently that this problem has received growing attention in computer science. One of the main reasons for this trend in machine learning and data mining has been to handle the rapidly growing quantities of data, more or less noisy, collected thanks are due to new acquisition technologies such as the world wide web [1–3].

Feature selection can be of great help to handle such a problem, primarily from an algorithmic and complexity-theoretic point of view. Indeed, exploiting and mining large data requires the help of powerful machine learning and data mining algorithms, which can be highly time or space consuming [1]. Provided feature selection is done in low complexity and reduces significantly the size of the data, it may provide an efficient preprocessing stage to reduce the time or space required practically by these machine learning and data mining stages. In contrast, from a pure classification standpoint, the selection of a good feature subset appears to be of little interest at first glance. Indeed, a Bayesian classifier, i.e. realizing Bayes optimal error, is monotonic. This means that adding features cannot decrease the model's performance. Theoretically speaking, in the feature selection framework, this statement postulates that removing features can be of no help in improving the model's performance. However, the monotonicity assumption rarely holds in practice [4]. The reason is that most practical machine learning and data mining algorithms are not ideal, and irrelevant or weakly relevant features may damage the accuracy of the model built. As an illustration, a study

* Corresponding author. Tel.: +33-596-72-73-64; fax: +33-596-72-73-62.

E-mail addresses: msebban@univ-ag.fr (M. Sebban), rnock@martinique.univ-ag.fr (R. Nock).

0031-3203/02/$22.00
PII: S0031-3203(01)00084-X

in Ref. [5] shows that with the decision-tree induction algorithm C4.5 [6], the non-deletion of weakly relevant features generates deeper decision trees with lower performances than those obtained without these features. In Ref. [7], the author shows that the storage of the IB3 algorithm increases exponentially with the number of irrelevant features. Similar conclusions are presented in Ref. [8].

Finally, feature selection can be of great help simply as a preprocessing step to induction algorithms, for the objective to reduce the size of the formulas found. Sebban et al. [9] report results concerning techniques removing examples (and not features), known as prototype selection algorithms. They show for example that these algorithms can reduce by more than 25% the size of the trees found by C4.5 when they are run before C4.5, without challenging the accuracy of the test. In many applications where comprehensibility and visualization are crucial issues, such a size reduction would be well worth the run of a data reduction technique before any further induction algorithm.

To summarize, scientists have been encouraged to elaborate sophisticated feature selection methods to tackle three problems:

- Reduce classifiers cost and complexity.
- Improve model accuracy.
- Improve the visualization and comprehensibility of induced concepts.

The difference between the features kept and those left by a feature selection algorithm can be characterized by a notion of relevance, a word we have already used, yet we have not provided a formal definition of what it is. Actually, there are many definitions of relevance, each of which addresses from a particular point of view the (relevant) question "relevant to what?" [10,11]. It is not the purpose of this paper to present the many answers which can be found. The reader may find general issues about this problem in the two aforementioned papers, and more specific computational issues about these definitions in Ref. [12]. In all that follows, the focus of our paper shall be on the resolution of the three problems cited before, better than addressing the way our algorithm copes with the selection of features relevant to some particular theoretical definition(s).

According to the terminology proposed in Ref. [11], two generic approaches are available in feature selection: wrapper and filter models. The principle of filter models is to evaluate, using statistical techniques over the data, the accuracy of the future, induced classifier. Therefore, the method "filters out" irrelevant features before the induction process. In wrapper models, we search for a good subset of features using the induction algorithm itself. The principle of wrapper models is generally based on the optimization of the accuracy rate, estimated by one of the following methods: holdout, cross-validation [13], or bootstrap [14].

In this article, we begin our analysis with a new filter approach to find relevant features. We exploit the characteristics of a neighborhood graph built on the learning set, to compute a new estimation criterion based on a quadratic entropy. The distribution of this criterion satisfies convenient normal properties, allowing the construction of a test to evaluate the quality of a feature subset. We use this statistical test (more precisely the critical threshold) in a forward selection algorithm. In order to reduce the computational costs of the neighborhood graph's construction, we propose a more general framework exploiting the 1-nearest-neighbor (1NN) graph. We show that this geometrical structure is less expensive to compute and leads to the construction of an original hybrid model of feature selection, presenting characteristics of both filter and wrapper approaches. Finally, we present some experimental results on benchmarks of the UCI database repository, or on tailor-made synthetic domains, comparing the performances of the feature subsets selected by our algorithms with those obtained in the original spaces, or with conventional approaches of feature selection.

2. Feature selection and hybrid models of feature selection

This part aims at presenting the general issues of feature selection, the principal problems that are raised, the usual solutions proned in filter models, as well as how our solution can be situated and motivated with respect to the other ones. We first begin with some elementary notations. We are given a p-dimensional representation space, where p is the number of features characterizing a set S of $|S| = n$ learning instances (or examples), where $|.|$ denotes the cardinality. Each instance ω_i is represented by a p-dimensional input vector $X(\omega_i) = (x_{i1}, x_{i2}, \ldots, x_{ip})$, and by a label (or class) $Y(\omega_i) \in Y$, $Y = \{y_1, y_2, \ldots, y_k\}$, where $k = \mathrm{card}(Y)$ is the number of classes.

Three problems participate in complicating the feature selection problem. First, elementary combinatorics show that feature selection should require the testing of 2^p different subsets to find the optimal one, which is a sufficiently large exponential to prevent the practical feasibility of the procedure, even for low-dimensional data.

Even worse, from a statistical point of view, even if we could guarantee the testing of all combinations, the quality of the feature subsets could practically only be estimated, on the sole basis of the potentially small set of instances available. Indeed, the learning instances do not cover the entire set of all possible examples, a set to which we refer as the whole domain. It can even be the case that the available examples scarcely cover a tiny portion

of it. Therefore, we cannot guarantee to be optimal in the sense of some definitions evaluating relevance with respect to this whole domain.

Finally, to complete the picture, from a complexity theoretic point of view, feature selection can be proven to be NP-hard for usual definitions of relevance [12]. Even worse, it can be proven that approximating the minimal relevant subset is hard up to very large factors [12]. Moreover, worst-case bounds of Ref. [12] establish that the performances of feature selection algorithms (even non necessarily polynomial time) can be almost as poor as the results obtained without feature selection! All these remarks show the necessity to build heuristics to address feature selection.

According to the paper of Ref. [15], four basic issues determine the nature of the heuristic search process:

- The starting point in the search space: with an empty feature set (forward selection) or with all the features (backward selection).
- The organization of the search: addition or deletion of an attribute at each stage, never reconsidering the previous choice.
- The strategy used to evaluate alternative subsets of attributes (filter or wrapper model).
- The criterion for halting search through the space of feature subsets. The simplest solution consists in fixing the size of the feature subset *ad hoc*.

With respect to these criteria, we basically consider in that paper a hybrid filter/wrapper approach using a statistical criterion for halting search, adding features one at a time, and starting from the empty feature subset.

If we drill down the concepts used for our approach and consider it more in depth, the primary idea of our approach was to use, for theoretical statistical reasons, a convenient topology over the examples to evaluate relevance. This topology is, we prove, similar to the one built using nearest-neighbor (NN) algorithms. Replacing this topology by the one of the 1-NN, we gain the benefit of fast computation while giving a wrapper flavor to our algorithm. When judged from a more practical point of view, this choice might appear quite disputable. On one hand, when used as a preprocessing step for a particular type of induction algorithm, a wrapper approach optimizing the accuracy during feature selection while using the same kind of formulas as the induction algorithm may be very convenient to improve its results [11]. On the other hand, the type of feature subset selected depends highly on the concept used during the wrapper algorithm. This is clearly not an advantage if an emphasis is made on the explanation of the features obtained: in that case, an approach less dependent on a specific classifier's accuracy is desirable.

With respect to these observations, it is important to note that the wrapper flavor in our algorithm is restricted with respect to basic wrapper approaches, in which most computation time is used to induce a concept representation (decision trees in many cases). Besides, the criterion which we optimize is not the accuracy. In the light of the numerous definitions of relevance [10], this is certainly an advantage: indeed, they establish that relevance is an intrinsic property of the concept represented by the attributes, thus imperfectly estimable by a particular formula's accuracy, subject to the representational bias of some induction algorithm. This gives the filter behavior of our algorithm, and makes it an original alternative to traditional feature selection algorithms, falling in either the filter, or the wrapper category.

To conclude this part on the general issues of feature selection, we now present some of the criteria used to estimate the quality of feature subsets in filter models. Five principal categories of measures can be found throughout the literature to evaluate the feature's relevance in feature weighting or selection algorithms (feature selection algorithms are weighting algorithms, where irrelevant or weakly relevant features have zero weight. For more details about feature weighting see Ref. [16]).

- *Interinstance distance*: this criterion is used in Kira and Rendell's RELIEF [17]. This method selects a random training case ω_j, a similar positive case ω_a, and a similar negative case ω_b. It then updates the feature weight, weight_i, using

$$\text{weight}_i = \text{weight}_i - \text{diff}(x_{ji}, x_{ai}) + \text{diff}(x_{ji}, x_{bi}), \qquad (1)$$

  where $\text{diff}(.,.)$ is a given metric. Based on this principle, Kononenko proposes an extension of RELIEF in Ref. [18].
- *Interclass distance*: the average distance between instances belonging to different classes is a good criterion to measure the relevance of a given feature space. However, the use of this criterion is restricted to problems without mutual class overlaps.
- *Probabilistic distance*: in order to correctly treat class overlaps, a better approach consists in measuring distances between probability density functions. This method of proceeding often leads to the construction of homogeneity tests [19].
- *Class projection*: this approach assigns weights using conditional probabilities on features that can be indiscriminately nominal, discrete or continuous [20].
- *Entropy*: feature selection can be understood in terms of information theory. One can then assign feature weights using Shannon's mutual information [21]; see also Ref. [22] where the cross-entropy measure is used. This approach is certainly the closest to ours.

We propose in the next section a new method of evaluating the feature's relevance. We assume that a classifier's ability to correctly label instances depends on the existence in the feature space of wide geometrical

structures of points identically labeled. We first characterize these structures using the information contained in a Minimum Spanning Tree. This information is used to apply a statistical test measuring what we call a *relative certainty gain*.

3. The test of relative certainty gain

3.1. Geometrical concepts

Our approach relies in searching characteristics of the learning sample in a neighborhood graph. More precisely, we use a minimum spanning tree over the learning sample, which is a simple structure to build, and has interesting geometrical properties. The construction of this neighborhood graph allows one to exploit local and global informations about the concept to learn. For the sake of completeness, we first review some basic definitions about graphs. They shall be completed in a next subsection to introduce our information theory material.

Definition 1. A tree is a connected graph without cycles.

Definition 2. A subgraph that spans all vertices of a graph is called a spanning subgraph.

Definition 3. A subgraph that is a tree and that spans all vertices of the original graph is called a spanning tree.

The following definition addresses *weighted* graphs, in which each edge is given a real weight.

Definition 4. Among all the spanning trees of a weighted and connected graph, the one(s) with the least total weight is(are) called the minimum spanning tree(s), abbreviated MST(s) for short.

Suppose we are given a metric over the p-dimensional representation space; we can easily build an MST by considering the weight of an edge as the distance between its two vertices. The MST therefore describes a tree with the lowest weight over the complete graph.

3.2. Metrics

Examples can be described using various types of features. While nearest neighbor techniques can handle continuous and discrete features well (using e.g. the Euclidean distance), they are not suited to handle nominal attributes, that is, symbolic attributes whose values do not reflect any linear order, such as red, black, green, white for color values [23]. The problem becomes difficult when all these attribute types coexist. Our approach to this problem is to use specific metrics adapted to the nature of each feature. We sketch it here for completeness. The main tool for adapting distances to nominal attributes is the value difference metric (VDM) [20] which defines the distance between two values x and x' of an attribute X_a as follows:

$$\mathrm{VDM}_{X_a}(x,x') = \sum_{c=1}^{k} \left| \frac{n_{X_a,x,c}}{n_{X_a,x}} - \frac{n_{X_a,x',c}}{n_{X_a,x'}} \right|^q = \sum_{c=1}^{k} |P_{X_a,x,c} - P_{X_a,x',c}|^q ,$$

where

- $n_{X_a,x}$ is the number of instances in the training set S that have a value x for attribute X_a,
- $n_{X_a,x,c}$ is the number of instances in S that have a value x for attribute X_a and output class c,
- k is the number of output classes,
- q is a constant, usually 1 or 2,
- $P_{X_a,x,c}$ is the conditional probability that the output class is c given that attribute X_a has the value x, i.e., $P(c|X_a = x)$.

VDM cannot be used for any type of feature, even if it provides a convenient solution for nominal attributes. The main problem of VDM is that it largely ignores continuous attributes, and requires discretization to map these continuous values into nominal values. But such continuous attributes are typically handled by the usual Euclidean metric. The fact that VDM and the Euclidean metric are complementary metrics has led to the creation of the heterogeneous value difference metric (HVDM) [23]. This metric mixes the usual Euclidean distance for linear (i.e. continuous or discrete) attributes, and VDM on nominal attributes. The distance between two instances ω_i and ω_j then is:

$$\mathrm{HVDM}(\omega_i,\omega_j) = \sqrt{\sum_{a=1}^{p} d_{X_a}^2(x_{ia},x_{ja})},$$

where $d_{X_a}(x, x')$ denotes the distance between the two values x and x' for attribute X_a, and is defined as

$$d_{X_a}(x,x') = \begin{cases} 1 & \text{if } x \text{ or } x' \text{ is unknown,} \\ & \text{otherwise,} \\ \mathrm{VDM}^*_{X_a}(x,x') & \text{if attribute } X_a \text{ is nominal,} \\ \mathrm{Euclid}^*_{X_a}(x,x') & \text{if attribute } X_a \text{ is continuous} \\ & \text{or discrete.} \end{cases}$$

Here, $\mathrm{Euclid}^*_{X_a}(x,x')$ represents a normalized Euclidean distance between x and x', and $\mathrm{VDM}^*_{X_a}(x,x')$ represents a normalized version of the VDM seen before for attribute

X_a. We refer the reader to Ref. [23] for further considerations about this metric, not needed here. Paper [23] also provides other heterogeneous distance functions for alternatives to HVDM in particular situations, i.e. when one needs to optimize the distances on a specific problem. We refer the reader to Ref. [23] for further details. These heterogeneous distance functions are called the interpolated value difference metric (IVDM) and the windowed value difference metric (WVDM). The point is that most of the distance functions in Ref. [23] properly handle nominal and continuous input attributes, and allow the construction of an MST in mixed spaces. Therefore, they can naturally be used as they are in our algorithms.

3.3. Entropy notions

Definition 5. Suppose we are given

$$\mathscr{S}_k = \left\{ (\gamma_1, \ldots, \gamma_j, \ldots, \gamma_k) \in \mathbb{R}^k : (\forall j \in \{1, 2, \ldots, k\}, \gamma_j \geqslant 0) \wedge \sum_{j=1}^{k} \gamma_j = 1 \right\}, \quad (2)$$

the k-dimensional simplex, where k is a positive integer. An entropy measure is an application from $\mathscr{S}_k$ to $\mathbb{R}_+$, with the following properties (for more details see Ref. [24]): Symmetry, Minimality, Maximality, Continuity and Concavity.

Definition 6. The quadratic entropy is a function QE from $[0,1]^k$ to $[0,1]$,

$$(\gamma_1, \ldots, \gamma_k) \to \mathrm{QE}((\gamma_1, \ldots, \gamma_k)) = \sum_{j=1}^{k} \gamma_j (1 - \gamma_j). \quad (3)$$

3.4. Local and total uncertainties in the MST

Given the previous definitions, we use the quadratic entropy concept to measure local and total uncertainties in the MST built on the learning set.

Definition 7. We define the neighborhood $N(\omega_i)$ of a given instance ω_i belonging to S as follows:

$$N(\omega_i) = \{\omega_j \in S : \omega_i \text{ is linked by an edge to } \omega_j \text{ in the MST}\} \cup \{\omega_i\}. \quad (4)$$

Definition 8. The local uncertainty $U_{loc}(\omega_i)$ for a given instance ω_i belonging to S is defined as follows:

$$U_{loc}(\omega_i) = \mathrm{QE}\left(\frac{n_{i1}}{n_{i.}}, \frac{n_{i2}}{n_{i.}}, \ldots, \frac{n_{ik}}{n_{i.}}\right) = \sum_{j=1}^{k} \frac{n_{ij}}{n_{i.}} \left(1 - \frac{n_{ij}}{n_{i.}}\right), \quad (5)$$

where $n_{i.} = card(N(\omega_i))$ and $n_{ij} = card(\{\omega_l \in N(\omega_i) \mid Y(\omega_l) = y_j\})$.

Definition 9. The total uncertainty U_{tot} in S is defined as follows:

$$U_{tot} = \sum_{i=1}^{n} U_{loc}(\omega_i) = \sum_{i=1}^{n} \frac{n_{i.}}{n_{..}} \sum_{j=1}^{k} \frac{n_{ij}}{n_{i.}} \left(1 - \frac{n_{ij}}{n_{i.}}\right), \quad (6)$$

where $n_{..} = \sum_{i=1}^{n} n_{i.} = n + 2(n-1) = 3n - 2$.

The reason for the use of U_{tot} and the quadratic entropy in order to evaluate the quality of some feature subset may not appear clear at first glance; however, U_{tot} is a natural measure of the impurity observed on the MST, that is, a measure of the overlaps between classes at the level of each instance's neighborhood. Intuitively, the better the feature subset, the smallest the overlaps, and the smallest U_{tot}. Furthermore, at each instance's level, the quadratic entropy becomes equivalent to Gini's impurity criterion, used in the decision tree induction to evaluate the quality of a tree node in the well known CART^{TM} package [24]. It is worthwhile to remark that Gini's criterion, as well as more recent criteria such as Schapire–Singer's Z criterion [25] have been rigorously proven to be very efficient measures to grow decision trees, in particular more accurate than the accuracy itself [26]. Furthermore, in our case, a convenient statistical test, which we now describe, allows to estimate with confidence whether a feature subset can be preferred to another one in our algorithm.

3.5. The statistical test

The previous criterion U_{tot} allows one to estimate the information level of the learning sample in a given feature space. The statistical test proposed is based on the following observation. In order to correctly estimate feature relevance, a convenient approach consists in measuring the class overlap degree in the probability density functions, and compare this one with the degree obtained with a total overlap. The way to proceed consists in applying what is called in inferential statistics a *homogeneity test*, with the following null hypothesis H_0:

$$H_0: \quad F_1(x) = F_2(x) = \cdots = F_k(x) = F(x),$$

where $F_i(x)$ is the repartition function of the class i. To be able to apply this test, we must know the law of the statistic used in the test (here, the total uncertainty U_{tot}) under the null hypothesis. Works proposed in Ref. [27] show that the distribution of the relative quadratic entropy gain is a χ^2 with $(n-1)(k-1)$ degrees of freedom.

Rather than considering directly U_{tot}, we use the following relative certainty gain,

$$\mathrm{RCG} = \frac{U_0 - U_{tot}}{U_0}, \tag{7}$$

where U_0 is the uncertainty of the learning set before the construction of the MST:

$$U_0 = \mathrm{QE}\left(\frac{n_1}{n}, \frac{n_2}{n}, \ldots, \frac{n_k}{n}\right) = \sum_{j=1}^{k} \frac{n_j}{n}\left(1 - \frac{n_j}{n}\right), \tag{8}$$

where $n_j = \mathrm{card}(\{\omega_i \mid Y(\omega_i) = y_j\})$. According to Ref. [27],

$$n_{..}\mathrm{RCG} \equiv \chi^2_{(n-1)(k-1)},$$

$$E(n_{..}\mathrm{RCG}) = (n-1)(k-1),$$

$$V(n_{..}\mathrm{RCG}) = 2(n-1)(k-1).$$

For reasonably large learning sets ($n > 30$), the distribution of $n_{..}\mathrm{RCG}$ is approximately normal with expectation $(n-1)(k-1)$ and variance $2(n-1)(k-1)$:

$$n_{..}\mathrm{RCG} \approx N((n-1)(k-1), 2(n-1)(k-1)). \tag{9}$$

The null hypothesis will then be rejected (up to risk α) if and only if:

$$\frac{n_{..}\mathrm{RCG} - (n-1)(k-1)}{\sqrt{2(n-1)(k-1)}} > U_\alpha \tag{10}$$

or equivalently, whenever

$$n_{..}\mathrm{RCG} > (n-1)(k-1) + U_\alpha\sqrt{2(n-1)(k-1)}, \tag{11}$$

where U_α is the value of the repartition function of the normal law $N(0,1)$ having probability α to be exceeded. Instead of fixing the α risk in advance (generally 5%), we can calculate the α_c critical threshold necessary for rejecting H_0. Then, we can optimize α_c as an estimation criterion to search for the feature subset which allows to be the farthest from the H_0 hypothesis. Actually, the smaller this risk is, the further from the H_0 hypothesis we are. Then, we use this risk α_c in the following feature selection algorithm.

4. The feature selection algorithm

The heuristic search, shown below, relies on a greedy forward selection algorithm, optimizing the critical threshold of the test at each time.

$\alpha_0 \leftarrow 1$;
$E \leftarrow \emptyset$;
$X \leftarrow \{X_1, X_2, \ldots, X_p\}$;
$stop \leftarrow false$;
do
  for each $X_i \in X$ **do**
    compute α_{ci}, the critical threshold
    in the $E \cup X_i$ feature space;
  $X_{\min} \leftarrow \arg\min_i \alpha_{ci}$;
  if $\alpha_{\min} < \alpha_0$ **then**
    $X \leftarrow X - \{X_{\min}\}$;
    $E \leftarrow E \cup \{X_{\min}\}$;
    $\alpha_0 \leftarrow \alpha_{\min}$;
  else
    $stop \leftarrow true$;
while $stop = false$;
return E;

5. Replacing the MST by the 1-NN Graph

In this section, we study how the MST can be replaced by the 1-NN graph, thus shifting the behavior of our algorithm toward wrapper approaches of feature selection. Comparing experimentally the two approaches is the subject of the next sections.

5.1. 1-NN graph is equivalent to the MST

Now, we show that the 1-NN graph is an adequate candidate to replace the MST. In most cases, all connex parts of the 1-NN graph are in fact small MSTs; in the remaining cases, they bear very close relationships with MSTs.

Definition 10. Let "$nn(X(\omega), X(\omega'))$" denote the relationship "$X(\omega)$ is the nearest neighbor of $X(\omega')$".

In the case where one example might have more than one nearest neighbor, the relationship is replaced by "$X(\omega)$ is one nearest neighbor of $X(\omega')$". The 1-NN graph may be represented using an oriented graph $G = (S, A)$ where S is the training set, and A is the nearest neighbor relationship. An arc comes from some $X(\omega) \in S$ to some $X(\omega') \in S$ whenever $nn(X(\omega), X(\omega'))$ holds, which additionally means that $X(\omega)$ would vote for $X(\omega')$ in the 1-NN algorithm. Before beginning the study of some properties of G, it is convenient to note that A mainly depends on the current subset of features chosen.

We denote as $G^* = (S, E)$ the simple non-oriented graph built from G by replacing each arc by an edge, and merging multiple edges between each couple of vertices $X(\omega)$ and $X(\omega')$, a situation which occurs only when both $nn(X(\omega), X(\omega'))$ and $nn(X(\omega'), X(\omega))$ hold.

In order to state the following lemma, we make the simplifying hypothesis (which shall be relaxed later) that for all examples in S their nearest neighbor is unique, which precludes the random choice of one of the nearest neighbors to vote.

Lemma 1. *G^* is cycle-free. In other words, G^* defines a forest.*

Proof. We first prove that G^* is cycle-free iff G is circuit-free. Fix

$$\forall X(\omega') \in S, d_i(X(\omega')) = |\{X(\omega) \in S: nn(X(\omega), X(\omega')) \text{ holds}\}| \, \forall X(\omega) \in S,$$

we have $d_i(X(\omega)) \leqslant 1$. Indeed, each example has exactly one nearest neighbor. Fix as $\{X(\omega_1), X(\omega_2), \ldots, X(\omega_{s'})\}$ a subset of S defining a cycle in G^*. That means that $\forall j \in \{1, 2, \ldots, s'-1\}$, $nn(X(\omega_j), X(\omega_{j+1})) \vee nn(X(\omega_{j+1}), X(\omega_j))$, and $nn(X(\omega_1), X(\omega_{s'})) \vee nn(X(\omega_{s'}), X(\omega_1))$. Since no vertex $X(\omega) \in \{X(\omega_1), X(\omega_2), \ldots, X(\omega_{s'})\}$ can satisfy $d_i(X(\omega)) > 1$, we obtain that the cycle of G^* necessarily satisfies exactly one of the following properties:

$$\forall j \in \{1, 2, \ldots, s'-1\}, nn(X(\omega_j), X(\omega_{j+1})) \wedge nn(X(\omega_{s'}), X(\omega_1)) \quad (12)$$

or

$$\forall j \in \{1, 2, \ldots, s'-1\}, nn(X(\omega_{j+1}), X(\omega_j)) \wedge nn(X(\omega_1), X(\omega_{s'})). \quad (13)$$

In other words, if G^* contains a cycle, G contains a circuit. The proof that G^* contains a cycle if G contains a circuit comes from the construction of G^*.

We now prove that G does not contain any circuit by contradiction. Fix as $\{X(\omega_1), X(\omega_2), \ldots, X(\omega_{s'})\}$ a subset of S defining a circuit in G. Without loss of generality, we suppose that the following property is satisfied: $\forall j \in \{1, 2, \ldots, s'-1\}, nn(X(\omega_j), X(\omega_{j+1})) \wedge nn(X(\omega_{s'}), X(\omega_1))$. If we note as $D(X(\omega), X(\omega'))$ the distance between $X(\omega)$ and $X(\omega')$ measured using the currently selected features, we get that $D(X(\omega_j), X(\omega_{j+1})) \leqslant D(X(\omega_{j+1}), X(\omega_{j+2}))$, $\forall j \in \{1, 2, \ldots, s'-2\}$, and

$$D(X(\omega_{s'-1}), X(\omega_{s'})) \leqslant D(X(\omega_{s'}), X(\omega_1)),$$

$$D(X(\omega_{s'}), X(\omega_1)) \leqslant D(X(\omega_1), X(\omega_2)).$$

Our simplifying hypothesis implies that at least one of the inequalities is strict, and we get a contradiction, as claimed. □

Lemma 2. *Any tree in the forest G^* is a minimum spanning tree.*

Proof. Again, the proof is obtained by contradiction. Fix some subset of S, $\{X(\omega_1), X(\omega_2), \ldots, X(\omega_{s'})\}$, defining a tree T in G^*. If it is not an MST, define T^{opt} as one possible MST having smaller weight. Since T and T^{opt} are trees, they contain exactly $s' - 1$ vertices and the addition of one edge in them breaks their tree structure by adding a cycle.

Summing up, because T and T^{opt} are necessary different, that means that there exists a subset of vertices $\{X(\omega_{j_1}), X(\omega_{j_2}), \ldots, X(\omega_{j_{s''}})\} \subseteq \{X(\omega_1), X(\omega_2), \ldots, X(\omega_{s'})\}$ such that

(1) $\{X(\omega_{j_1}), X(\omega_{j_2}), \ldots, X(\omega_{j_{s''}}\}$ defines a chain in T,
(2) $(X(\omega_{j_1}), X(\omega_{j_{s''}})$ is in T^{opt} but not in T,
(3) $\exists i \in \{1, 2, \ldots, s'' - 1\}$ such that $(X(\omega_{j_i}), X(\omega_{j_{i+1}}))$ is in T but not in T^{opt},
(4) $D(X(\omega_{j_1}), X(\omega_{j_{s''}})) < D(X(\omega_{j_i}), X(\omega_{j_{i+1}}))$.

We show that (1–3) render (4) impossible. Suppose without loss of generality that $nn(X(\omega_{j_i}), X(\omega_{j_{i+1}}))$ holds.

Since no vertex $X(\omega) \in \{X(\omega_{j_1}), X(\omega_{j_2}), \ldots, X(\omega_{j_{s''}})\}$ can satisfy $d_i(X(\omega)) > 1$, we have

$$\forall l \in \{j_i, \ldots, j_{s''-1}\}, nn(X(\omega_{j_l}), X(\omega_{j_{l+1}})) \text{ holds.}$$

and we necessarily have the following chain of inequalities:

$$D(X(\omega_{j_i}), X(\omega_{j_{i+1}})) \leqslant D(X(\omega_{j_{i+1}}), X(\omega_{j_{i+2}})) \leqslant \cdots \leqslant D(X(\omega_{j_{s''-1}}), X(\omega_{j_{s''}})).$$

Because of (2) however, and the fact that $nn(X(\omega_{j_{s''-1}}), X(\omega_{j_{s''}}))$ holds, we also have

$$D(X(\omega_{j_{s''-1}}), X(\omega_{j_{s''}})) < D(X(\omega_{j_1}), X(\omega_{j_{s''}})).$$

The chain of inequalities and the latter give $D(X(\omega_{j_i}), X(\omega_{j_{i+1}})) < D(X(\omega_{j_1}), X(\omega_{j_{s''}}))$, a contradiction with Eq. (4), as claimed. This ends the proof of Lemma 2. □

For any graph (oriented or simple and non-oriented) $G = (V, A)$, and any subset $A' \subseteq A$, the graph $G = (V, A')$ is called the partial subgraph of G induced by $A' \subseteq A$. A connex component of a simple non-oriented graph $G = (V, A)$ is a non-empty subgraph $G' = (V', A')$ of G that satisfies the following properties: (i) $V' \subseteq V$, (ii) $\forall v \in V \setminus V'$, $\forall v' \in V'$, $(v, v') \notin A$, (iii) $\forall (v, v') \in V'^2$, $(v, v') \in A \Rightarrow (v, v') \in A'$. In the general case where the unicity of the nearest neighbor is not ensured, Lemma 2 can be easily generalized:

Lemma 3. *The minimum spanning tree $T = (S', A')$ defined over the subset of S containing the vertices S' of some connex component of G^* is a partial subgraph of this connex component (thus, induced by A').*

Proof. The proof of Lemma 2 states that any MST defined over a subset of vertices consisting of a connex component of G^* cannot contain vertices that are not in that connex component. □

5.2. Complexity of computing the MST vs the 1-NN graph

In order to compare the building of the MST with that of the 1-NN graph, we make the hypothesis that the distance matrix between examples is pre-computed. That requires $\mathcal{O}(|S|^2K)$ where K is the cost of computing the distance between two examples, a function essentially depending on the number of features.

5.2.1. Building the MST using Kruskal or Prim's algorithms

Both the algorithms are made faster by the precomputation of the distance matrix between examples, in order to sort them by an increasing order. Their overall complexity is of order $\mathcal{O}(|S|^2 \log |S|)$.

5.2.2. Building the 1-NN graph

The computation of the 1-NN graph can be done without sorting the edges. For any vertex, we only need to find its nearest neighbor, which requires $\mathcal{O}(|S|)$ steps. The overall complexity is therefore $\mathcal{O}(|S|^2)$ steps, and precludes the $\mathcal{O}(|S|^2 \log |S|)$ steps for sorting the edges in the MST algorithms. This represents a clear advocacy for the use of the 1-NN graph instead of the MST.

5.2.3. Reaching linear complexity for building the 1-NN graph

The MST could not be intuitively computed in less than $|S|(|S|-1)/2$ steps since it needs to explore in the worst case all edges, even if the precomputation of the distance matrix also orders the distances, modulo a penalizing $\log |S|$ factor for its complexity. However, the time complexity for computing the 1-NN graph can easily be dropped down without additional costs. In order to achieve it, we include in the precomputation of the distance matrix a test of constant-time for each couple of instances, thus without increasing the overall complexity in computing the matrix. We keep for each instance its current nearest neighbor found, which can easily be done by checking it along the two currently explored instances whose distance is computed. Eventually, multiple nearest neighbors are collected into a current list. The remaining task for the 1-NN graph is simply to merge all lists or singletons for each instances, which can be done in at most $|S|$ steps.

5.3. Toward a hybrid approach to feature selection

The choice to build the 1-NN graph instead of the MST has a very important side effect regarding the classification of feature selection algorithms. Our algorithm becomes the optimization over a precise topology which is that of the final classifier, of a criterion being not the accuracy. While it keeps the filter behavior, the new approach shifts its behavior a little toward wrapper algorithms, even though it does not suffer the drawback of time-consuming concepts induction. Rather than keeping the filter's term for the algorithm, we now relate to it as a hybrid approach. Two experimental sections now follow, studying respectively the MST and the 1-NN graph. The first one on the MST evaluates the interest of our approach of feature selection versus no feature selection stage. In the following experimental section studying the 1-NN graph, we provide some comparisons between (i) this hybrid approach, (ii) the first one using the MST, and (iii) a conventional wrapper approach in which the RCG is replaced by the accuracy.

6. Experimental results on the MST

In order to show the applicability of a new approach, an experimental study should satisfy two criteria: relevance and insight [28]. Relevance measures the implications of the technique for problems on which the technique may be used in practice, that is why this criterion is best satisfied by performing experiments on real world problems. Insight is aimed at testing explicit hypotheses on the technique, by performing experiments on tailor-made data; thus, this criterion is best satisfied by performing experiments on synthetic problems. In this section, we present some experimental results on the two types of problems. Some experiments concern synthetic domains. In that case, we know a priori the number of relevant and irrelevant features. There are three synthetic domains:

- *Synt*1: 10 features, among which seven have various degrees of relevance, and three are irrelevant (X_8, X_9, X_{10}).
- *Synt*2: 10 feat ures, among which three are redundant features (X_1, X_2, X_3), and seven are irrelevant (X_4 through X_{10}).
- *Synt*3: 100 features, including seven identically distributed relevant features (X_1–X_7), and 93 irrelevant (X_8–X_{100}).

Irrelevant features in all these synthetic domains are i.i.d. $N(0,1)$ random variables.

The second type of problems concern natural domains. We test our algorithm on 10 data sets, among which seven belong to the UCI database repository. [1]

Results of Table 1 show that performances of our feature selection algorithm are interesting, eliminating both irrelevant and redundant features. In the majority of cases, the accuracy estimates obtained with a 5-fold-cross-validation using a 10-NN classifier are better in the selected feature subspace than with all the attributes. Statistically speaking, a sign test gives in addition a threshold probability $p_t \approx 10,94\%$ for

[1] http://www.ics.uci.edu/ ∼mlearn/MLRepository.html.

Table 1
Results on synthetic and natural domains: Acc_1 corresponds to the accuracy estimates with all the original features (recall that p is their cardinality) and Acc_2 presents results with the selected feature subset (p' is the subspace size). Best results are indicated using bold faces

Dataset	p	Acc_1	p'	Acc_2
Audiology	69	70.2	21	**70.3**
Breast	13	66.2	3	**82.7**
Echocardiogram	6	**65.9**	1	64.9
Glass2	9	**64.6**	8	63.6
Hepatitis	19	78.5	9	**80.2**
Iris	4	82.3	2	**93.5**
Synt1	10	85.0	4	**87.4**
Synt2	10	72.4	5	**73.0**
Synt3	100	75.3	2	**77.4**
White house	16	91.5	1	**95.7**

testing the hypothesis "the accuracy gives results at least as good as those of the RCG criterion", while the RCG uses on average almost 80% less features than the accuracy. Given the relatively small number of datasets to carry out this test, this represents an additional advocacy for the use of the RCG criterion. Concerning the White House domain, it is well known [29] that there exists one attribute (physician-fee-freeze) which gives more than 95% accuracy on the test. This attribute is exactly the one selected by our algorithm. Now, we investigate the replacement of the MST by the 1-NN graph.

7. Experimental results on the 1-NN

In order to analyze the performances of our new approach, our experiments cope with two objectives:

(1) Check that our criterion built from the 1-NN graph allows one to select a good subset of features, as relevant as the one selected with the MST.
(2) Compare performances of our model with the wrapper model optimizing the accuracy of the 1-NN. That amounts to comparing our hybrid model with the equivalent wrapper model, that is, the wrapper model having access to approximately the same "information" on the neighborhoods.

According to these objectives, our algorithm was run on 23 databases, most of which come from the UCI database repository. Dataset LED is the classical LED recognition problem [24], but to which the original ten classes are reduced to two: even or odd. LED24 is LED to which 17 irrelevant attributes are added. Hard is a hard problem consisting of two classes and 10 features per instance. There are five irrelevant features. The class is given by the XOR of the five relevant features. Finally, each feature has 10% noise. The Xd6 problem was previously used by Ref. [30]: it is composed of 10 attributes, one of which is irrelevant. The target concept is a disjunctive normal form over the nine other attributes. There is also classification noise. Other problems were used as they appeared in the UCI repository in the 1998 distribution [29]. For each database, we used the following experimental setup:

(1) A first feature subset is selected optimizing the information criterion based on the 1-NN graph.
(2) A second feature subset is selected optimizing the accuracy of the 1-NN rule at each step of the algorithm.
(3) In order to compare the relevance of the selected subsets, we use a posteriori a 10-NN classifier in a 5-fold-cross-validation procedure. We applied this strategy not only on the two selected subsets (RCG and accuracy), but also on the whole set of features (all attributes). The results are presented in Table 2.

A way to analyze the results consists in comparing the performances of RCG vs. accuracy, RCG vs. all attributes, and accuracy vs. all attributes. We can note that:

(1) Overall, our criterion built on the 1-NN graph allows one and to obtain results similar to the MST. Among nine common databases (Tables 1 and 2) treated by the two geometrical structures, four give the same selected feature subset, three are better for the MST, and two are better for the 1-NN.
(2) In the majority of cases, RCG presents better results than those obtained by optimizing the accuracy. Among 23 databases, RCG allows 10 times better accuracy, is identical 8 times, and has only 5 times smaller accuracy. Globally, the mean gain of RCG is about $+1.6\%$. A sign test gives a threshold probability $p_t \approx 0.05$, which is significant.
(3) The advantage of RCG is confirmed by analyzing the results of all attributes. Actually, for 17 benchmarks, RCG allows a better accuracy (on average $+3.0\%$), with less features. A sign test gives a very low threshold probability: $p_t \approx 0.0053$, which is highly significant.
(4) The advantage of accuracy against all attributes appears to be less significant: a sign test now gives a greater threshold probability $p_t \approx 0.11$.

The preceding experimental results show that the accuracy is not an accurate criterion to be optimized, since it is outperformed by the RCG. Such results were previously observed and theoretically explained in decision-tree induction. In Ref. [26], a formal proof is given which explains why the Gini criterion and the entropy should be optimized instead of the accuracy when a top-down induction algorithm is used to grow a decision-tree. Their

Table 2
Results on 23 databases: the three last columns contain the a posteriori accuracy by cross-validation in the three different feature spaces (All stands for all Attributes, and Acc for accuracy). Bold faces indicate the best result(s)

Dataset	p	$\|S\|$	RCG	All	Acc
Audiology	69	226	69.3	70.2	**77.1**
Australian	14	690	**84.6**	76.4	**84.6**
BigPole	4	3481	**62.9**	62.2	**62.9**
Breast cancer	9	699	93.3	**95.8**	95.0
Echocardiogram	6	131	**66.8**	65.9	60.1
German	24	1000	69.2	**71.4**	69.9
Glass2	9	163	**65.6**	64.6	63.1
Hard	10	256	**52.4**	49.0	48.8
Heart	13	270	74.6	**77.6**	74.7
Hepatitis	19	155	79.1	78.5	**79.2**
Horse	22	368	**75.7**	66.5	**75.7**
Iris	4	150	**93.5**	83.6	**93.5**
LED	7	500	87.0	**88.0**	77.9
LED24	24	200	**84.7**	70.3	79.1
Monks 1	6	432	**83.6**	75.6	**83.6**
Pima	8	768	**70.1**	68.7	**70.1**
Synt1	10	300	**87.4**	85.0	86.0
Synt2	10	300	**73.0**	72.4	**73.0**
Synt3	100	300	**77.4**	75.3	**77.4**
Vehicle	18	846	**72.2**	68.4	70.5
Waves	21	501	**79.3**	78.1	79.1
White house	16	435	**95.5**	89.1	94.3
Xd6	10	600	**74.4**	**74.4**	61.8

theoretical results support the claim according to which maximization the accuracy should be done directly by maximizing the accuracy's increasing using a highly concave criterion, like Gini's or the entropy. In addition, Ref. [26] provides an optimal criterion which should give the maximal increase of the accuracy. This criterion was later used in the AdaBoost boosting algorithm of Ref. [25], and we refer to it as Schapire–Singer's Z criterion. It is a function from $[0,1]^k$ to $[0,1]$:

$$(\gamma_1,\ldots,\gamma_k) \rightarrow Z(\gamma_1,\ldots,\gamma_k) = \sum_{j=1}^{k} \sqrt{\gamma_j(1-\gamma_j)}. \qquad (14)$$

The results of Ref. [26], along with our results on the comparsion of the RCG's and the accuracy's optimization on the 1-NN graph (that of the final classifier), are an advocacy for testing the optimization of Schapire–Singer's Z criterion itself. In the experiments of Table 3, we give a comparison between its optimization and that of the quadratic-entropy. We can note that among 15 databases, the feature subsets are 11 times similar. If we except the audiology data set, the optimization of Z does not bring advantages in feature selection. Nevertheless, it is important to note that we do not dispose of a convergence in law's result for the Z criterion. This surely makes a stopping rule for the growing of the selected feature's set more hazardous.

Table 3
Comparisons between RCG and the Schapire–Singer's Z criterion on 15 databases. When single, the best result is indicated using bold faces

Dataset	RCG	Z
Audiology	69.3	**72.2**
BigPole	62.9	62.9
Echocardiogram	**66.8**	62.7
Glass2	65.6	65.6
Hard	52.4	52.4
Heart	74.6	74.6
Hepatitis	79.1	79.1
Horse	75.7	75.7
Iris	93.5	93.5
LED	87.0	87.0
LED24	84.7	84.7
Pima	70.1	70.1
Vehicle	72.2	72.2
White house	**95.5**	94.5
Xd6	**74.4**	73.8

8. Conclusion

Algorithms allowing to improve the reliability and interpretability of concept construction in machine learning and data mining have become a central issue of these fields, with the development of new data acquisition techniques, and the increase in the size of databases.

Feature selection algorithms are potential candidates to address efficiently these problems. We have presented in this paper a feature selection model based both on information theory and statistical tests. A feature is selected if and only if the information given by this attribute allows to statistically reduce class overlaps. Results on synthetic and natural domains show that our tool is suited to treat irrelevant and redundant features, even in very large feature spaces. In our approach, two parameters were optimized. The first one concerns the geometrical structure to apply on the learning set, on which our criterion is built. The analysis of the paper shows that the 1-NN graph presents a framework similar to that of the MST, and allows in reducing the complexity of our algorithm. Second, we analyzed which criterion to optimize in our algorithm. Our study shows that the quadratic entropy (which has already shown its advantages in the decision tree field) not only seems to be significantly better than the accuracy, but also, and more surprisingly, better than the Z criterion of Ref. [25]. The analysis and explanation of this phenomenon shall be the subject of future works.

References

[1] R. Nock, M. Sebban, Advances in adaptive prototype weighting and selection, Artif. Intell. Tools 10 (1–2) (2001), to appear.

[2] M. Sebban, R. Nock, Contribution of boosting in wrapper models, Proceedings of the 3rd European Conf. on Principles and Practice of KDD, 1999, pp. 214–222.

[3] M. Sebban, R. Nock, Prototype selection as an information-preserving problem, Proceedings of the 17th Int. Conf. on Machine Learning, 2000, pp. 855–862.

[4] R. Kohavi, Feature subset selection as search with probabilistic estimates, AAAI Fall Symp. on Relevance, 1994.

[5] S.B. Thrun, J. Bala, E. Bloedorn, I. Bratko, B. Cestnik, J. Cheng, K. De Jong, S. Dzeroski, S.E. Fahlman, D. Fisher, R. Hamann, K. Kaufman, S. Keller, I. Kononenko, J. Kreuziger, R.S. Michalski, T. Mitchell, P. Pachowicz, Y. Reich, H. Vafaie, W. Van de Welde, W. Wenzel, J. Wnek, J. Zhang, The MONK's problems: a performance comparison of different learning algorithms, Technical Report CMU-CS-91-197, Carnegie Mellon University, 1991.

[6] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, Los Altos, CA, 1994.

[7] D. Aha, Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms, Int. J. Man Mach. Studies 36 (1992) 267–287.

[8] P. Langley, W. Iba, Average case analysis of a nearest-neighbor algorithm, Proceedings of the 13th Int. Joint Conf. on Artificial Intelligence, 1993, pp. 889–894.

[9] M. Sebban, R. Nock, J.-H. Chauchat, R. Rakotomalala, Impact of learning set quality and size on decision tree performances, Int. J. Comput. Systems Signals 1 (2001) 85–105.

[10] A. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artif. Intell. 97 (1997) 245–272.

[11] George H. John, Ron Kohavi, Karl Pfleger, Irrelevant features and the subset selection problem, Proceedings of the 11th Int. Conf. on Mach. Learning, 1994, pp. 121–129.

[12] R. Nock, M. Sebban, Sharper bounds for the hardness of prototype and feature selection, Proceedings of the 11th Int. Conf. on Algorithmic Learning Theory, 2000, pp. 224–237.

[13] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, Proceedings of the 15th Int. Joint Conf. on Artif. Intell. 1995, pp. 1137–1143.

[14] B. Efron, R. Tibshirani, An introduction to the bootstrap, Chapman & Hall, New York, 1993.

[15] P. Langley, Selection of relevant features in machine learning, AAAI Fall Symp. on Relevance, 1994.

[16] D. Wettschereck, D. Aha, Weighting feature, Proceedings of the 1st Int. Conf. on Case-Based Reasoning, 1995, pp. 347–358.

[17] K. Kira, L. Rendell, A practical approach to feature selection, Proceedings of the 9th Int. Conf. on Machine Learning, 1992, pp. 249–256.

[18] I. Kononenko, Estimating attributes; analysis and extension of RELIEF, Proceedings of the 10th European Conf. on Machine Learning, 1995, pp. 171–182.

[19] C. Rao, Linear Statistical Inference and Its Application, Wiley, New York, 1965.

[20] C. Stanfill, D. Waltz, Towards memory-based reasoning, Commun. ACM (1986) 1213–1228.

[21] D. Wettschereck, T. Dietterich, An experimental comparison of the nearest neighbor and nearest hyperrectangle algorithms, Mach. Learning 19 (1995) 5–28.

[22] D. Koller, R.M. Sahami, Toward optimal feature selection, Proceedings of the 13th Int. Conf. on Machine Learning 13 (1996) 284–292.

[23] D. Wilson, T. Martinez, Improved heterogeneous distance functions, J. Arti. Intell. Res. (1997) 1–34.

[24] L. Breiman, J.H. Freidman, R.A. Olshen, C.J. Stone, Classification and Regression Trees., Wadsworth, Belmont, CA, 1984.

[25] R.E. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Proceedings of the 11th Int. Conf. on Computational Learning Theory, 1998, pp. 80–91.

[26] M.J. Kearns, Y. Mansour, On the boosting ability of top-down decision tree learning algorithms, Proceedings of the 28th Ann. ACM Symp. on the Theory of Computing, 1996, pp. 459–468.

[27] R. Light, B. Margolin, An analysis of variance for categorical data, J. Amer. Stat. Assoc. 66 (1971) 534–544.

[28] P. Langley, Relevance and insight in experimental studies, IEEE Expert 11 (1996) 11–12.

[29] C.L. Blake, C.J. Merz, UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/ ∼ mlearn/MLRepository.html.

[30] W. Buntine, T. Niblett, A further comparison of splitting rules for decision-tree induction, Mach. Learning 8 (1992) 75–85.

About the Author—MARC SEBBAN was born in Lyon, France in 1969. He received his Ph.D. in Computer Science from the University of Lyon 2 in 1996. Since 1997, he has been Assistant Professor within the Department of Mathematics and Computer Science at the "Université Antilles-Guyane", located in the Guadeloupe campus of the University. His current research interests include: machine learning, feature and prototype selection, knowledge discovery, data mining and computational geometry.

About the Author—RICHARD NOCK was born in Vienne, France in 1970. He received his Ph.D. in Computer Science from the University of Montpellier II, France, in 1998. He also holds an Agronomical Engineering degree from the Ecole Nationale Supérieure Agronomique de Montpellier (ENSA.M), France. Since 1998, he has been Assistant Professor within the Interfaculty Department at the "Université Antilles-Guyane", located in the Martinique campus of the University. His current research interests include: machine learning, data mining, computational complexity and image processing.