J Intell Inf Syst (2008) 30:273–292
DOI 10.1007/s10844-007-0037-0

Consistency measures for feature selection

Antonio Arauzo-Azofra · Jose Manuel Benitez · Juan Luis Castro

Received: 9 November 2004 / Revised: 9 August 2006 /
Accepted: 2 January 2007 / Published online: 27 February 2007

Abstract The use of feature selection can improve accuracy, efficiency, applicability and understandability of a learning process. For this reason, many methods of automatic feature selection have been developed. Some of these methods are based on the search of the features that allows the data set to be considered consistent. In a search problem we usually evaluate the search states, in the case of feature selection we measure the possible feature sets. This paper reviews the state of the art of consistency based feature selection methods, identifying the measures used for feature sets. An in-deep study of these measures is conducted, including the definition of a new measure necessary for completeness. After that, we perform an empirical evaluation of the measures comparing them with the highly reputed wrapper approach. Consistency measures achieve similar results to those of the wrapper approach with much better efficiency.

Keywords Feature selection · Attribute evaluation · Consistency · Measures

A. Arauzo-Azofra (✉)
Department of Rural Engineering, University of Cordoba, Cordoba 14071, Spain
e-mail: arauzo@uco.es

J. M. Benitez · J. L. Castro
Department of Computer Science and Artificial Intelligence, University of Granada, Granada 18071, Spain

J. M. Benitez
e-mail: j.m.benitez@decsai.ugr.es

J. L. Castro
e-mail: castro@decsai.ugr.es

1 Introduction

Feature selection help us to focus the attention of an induction algorithm in those features that are the best to predict a target concept. Although theoretically, if the full statistical distribution were known, using more features could only improve results, in practical learning scenarios it may be better to use a reduced set of features (Kohavi & John, 1997). Sometimes a large number of features in the input of induction algorithms may turn them very inefficient as memory and time consumers, even turning them inapplicable. Besides, irrelevant data may confuse learning algorithms making them to reach false conclusions, leading them to get worse results.

Apart from increasing accuracy, efficiency and applicability of induction algorithms, the costs of data acquisition may also be reduced when a smaller number of features is selected, and the understandability of the results of induction algorithm improved.

All those advantages have made that feature selection had attracted much attention by the Machine Learning community, and many feature selection methods have been developed. In order to classify them, some categorizations (Dash & Liu, 1997; Jain & Zongker, 1997; Langley, 1994) have been proposed. These studies identify some different parts of feature selection algorithms. According to the different parts identified, we propose the modularization of the feature selection process to allow a better way of studying: the methods, their possible improvements, and the development of new ones. In this paper we focus our attention on one of the identified parts, the evaluation function of a given feature set.

The evaluation functions may be used with different purposes inside the feature selection process. We identify some of these uses of evaluation functions, and consider two of them as the most common and important: choosing the best feature set among those evaluated and guiding the search. An evaluation function that is able to choose the best set is not necessarily the best to guide the search.

Many different evaluation functions may be used. The aim of the search is to optimize the evaluation function, whether minimizing or maximizing its value. Usually evaluation functions are measures of some quality of the feature set regarding the data set. In this work, we make a review of those measures based on consistency that have been used in feature selection. The review also covers other consistency based feature selection methods not directly based on measures. In order to fill what we consider a natural gap in consistency measures, we formally define a measure that uses previous ideas. All these measures are evaluated and compared with the wrapper approach to feature selection.

This paper is structured as follows. In Section 2, we start describing the proposed modular decomposition of a feature selection algorithm, and the measures for feature sets. Section 3 studies the consistency measures and reviews the consistency based feature selection methods. After that, an empirical study of the measures is presented and explained in Section 4. And finally, conclusions and future work are described in Section 5.

2 Feature selection process

The problem of feature selection can be seen as a search problem on the powerset of the set of available features (Kohavi, 1994; Langley, 1994). The goal is finding a subset of features that allows us to improve, in some aspect, a learning activity.

In general, we can identify some parts of feature selection algorithms with different functionalities. Inside the process followed by feature selection methods we usually find:

– A search method through the feature sets space
– An evaluation function of a given set of features

The schema of Fig. 1 shows a modular decomposition of the whole feature selection process. It is based on the four issues (Langley, 1994) identified on feature selection methods. The divisions are also similar to those proposed by Dash and Liu (1997), with the addition of the starting point and the removal of the validation process. Although validation is highly recommended, it is not essential, and it is outside of the main feature selection algorithm, as it was already pointed out in the same work.

In the search process we may identify three issues: the choice of a starting point, the process of generating the next set to explore, and a stopping criterion. Instead of considering these as three independent issues, we have grouped them together because they define the search strategy, and there is a stronger relation among them than with the evaluation function.

The evaluation function, given a feature subset (S) and the training data set (T), returns a measure of the goodness of that feature set.

$$\text{Evaluation function} : S \times T \longrightarrow \mathbb{R} \tag{1}$$

There is a wide range of evaluation functions used in feature selection. Evaluation functions may be deterministic or non-deterministic, and sometimes they are probabilistic estimates of a theoretical measure. The functions may exhibit different properties, for example monotonicity. Their range will normally be in an interval like $[0, 1]$ or $[-1, 1]$, it may also be just a boolean value $\{0, 1\}$ indicating if the feature set is acceptable or not as a result.

At least three main uses of the evaluation functions may be identified. First, they are normally used as a criterion to choose, among all the explored feature subsets, which one is the best. In this case, the feature selection process will return the feature subset that optimizes the measure.

Another common use of the evaluation function is to guide the search process, as it is done for example in branch and bound (Somol & Pudil, 2004; Kudo & Sklansky, 2000), genetic algorithms (Brill, Brown, & Martin, 1992; Kudo & Sklansky, 2000), or the greedy search method explained below. Other methods use different

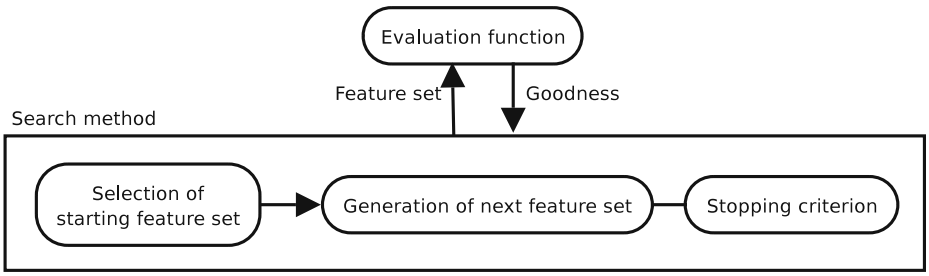

Fig. 1 Feature selection process

search strategies independent of the evaluation function. For example, exhaustive and random search explore feature sets ignoring the evaluation of previous set.

Finally, we can see methods like FOCUS2 (Almuallim & Dietterich, 1994). While having an independent (not based on an evaluation measure) search strategy built in its generation process, FOCUS2 uses a test of consistency, that they called sufficiency test, to decide when to stop the search. This consistency test can be seen as a binary evaluation function, that is used by the stopping criterion.

The modular view of the feature selection process presented allow us to develop a better understanding of feature selection methods, by getting an insight view of them. Besides, we can investigate different approaches to each of the modules independently of the others, as in this work we study some evaluation measures based on the consistency concept using a fixed fast greedy search process. In addition, using this model, it is possible to create a great variety of feature selection algorithms by combining different evaluation functions and search options. Finally, the parts could possibly be reused with other purposes than feature selection, for example, some evaluation functions are used in discretization.

Some feature selection methods do not have some of the modules identified in this schema, but they still fit on it. For example, Relief (Kira & Rendell, 1992) does not use a feature set evaluation function, and it does not even perform a search in the feature set space. It simply estimates the quality of features individually, like other feature weighting methods (Wettschereck, Aha, & Mohri, 1997), and then selects those with weight above a user given threshold. In this schema, Relief will only have a starting point strategy, there is no next set generation process, and the stopping criterion is just returning the starting set. Placing Relief in this schema reveals that it can be used as a starting point strategy for other methods.

Three different strategies to feature selection have been identified (Blum & Langley, 1997). The filter approach, where some features are selected before and independently of the learning algorithm. The wrapper approach, that uses the learning algorithm inside the feature selection process. And the embedded approach, in which the learning and feature selection are interlaced in one indivisible algorithm. All feature selection methods identify a subset of features to be used in the learning process. Learning algorithms may exhibit different grades of tolerance to irrelevant or redundant features, but if these algorithms do not identify which features to use they are not feature selectors. They should not be confused with embedded approaches to feature selection.

All the previously mentioned examples of modularized feature selection methods belong to the filter approach. The wrapper approach (John, Kohavi & Pfleger, 1994) also fits perfectly on the proposed schema. It aims at improving results by using the targeted learning algorithm in the evaluation function. The targeted learning algorithm is run with the candidate feature subset, and some quality measure of the results achieved is used as the evaluation measure. In this way the bias of the learning algorithms is taken into account by the feature selection.

While the wrapper approach has proven useful with very good results in some circumstances, it is still interesting to study other evaluation measures for several reasons that follows. First, an evaluation function may be more efficient in time or resources than the learning algorithm. Second, some learning algorithms can not be used with many features. In fact, this is one of the reasons to use feature selection. Such algorithms may render the wrapper approach inapplicable. And finally, some

evaluation measures may be better than the wrapper approach to guide the search process in some circumstances.

3 Consistency evaluation measures

Many different evaluation functions have been used in feature selection. A categorization of these functions according to their theoretical basis is proposed in Dash and Liu (1997). The categories identified are: distance measures, information measures, dependence measures, consistency measures and classifier error rate measures.

This work is centered on consistency measures. The idea behind these measures is that, in order to predict the concept or class value of its instances, a data set with the selected features alone must be consistent. That is, no two instances may have the same values on all predicting features if they have a different concept value. Therefore, the goal is equivalent to select those features that better allow to define consistent logical hypothesis about the training data set.

As the higher the number of features, the more consistent hypothesis that can be defined, the requisite, of a data set having consistency, is usually accompanied with the criterion of finding a small feature set. In any case, the search for small feature sets is the common goal of feature selection methods, so this is not a particularity of consistency based methods.

3.1 Basic consistency measure

The most basic of these measures is the one that simply guess if the training data set is consistent or not with the selected features. Its output is just a boolean value. This measure was first used in FOCUS (Almuallim & Dietterich, 1991), as what they called the sufficiency test. The search process of FOCUS, or the optimized version, FOCUS2 (Almuallim & Dietterich, 1994), uses this measure to stop the search in the first set of features that this measure evaluates to true. The algorithms perform the search in a way that guarantees finding a minimal set of features that make the training set consistent. This implements what they called the min-features-bias.

While good results had been achieved using the simple consistency measure, it has several limitations. First, consistency check can only be used directly with discrete features. Developing an extension of FOCUS algorithm to deal with these features is not straightforward and many approaches are possible. Some extensions are CFOCUS (Arauzo-Azofra, Benitez-Sanchez & Castro-Peña, 2003a) to handle continuous features, and F (Arauzo-Azofra, Benitez-Sanchez & Castro-Peña, 2003b) to include expert knowledge in the form of linguistic features. Second, FOCUS has low noise tolerance, just the change of a single value may turn the set inconsistent and force to add another feature, that may be redundant or even irrelevant. And third, the measure itself is not able to guide the search, it is necessary an additional strategy, like min-feature-bias, or any other that using the data may be able to direct the search in a profitable way. The consistency measures that are described in the following subsections aim at improving noise tolerance and providing a mean to guide the search by returning a degree of consistency.

All the consistency measures studied can emulate this measure by converting their output to a boolean value. When the data set is consistent the measures always return

a given value, usually 1, and a different value otherwise. In this way, it is possible to implement FOCUS with any consistency measure, but with some advantages, for example, being able to stop before reaching complete consistency to handle noise.

3.2 Liu's consistency measure

Liu, Motoda and Dash (Liu et al., 1998) proposed the first consistency measure defined independently of a search process in feature selection. More recently they have tested the measure with several search processes (Dash & Liu, 2003).

This measure uses an inconsistency rate that is computed by finding all examples (patterns) with the same values in all features (not considering the class feature), and counting all matching examples minus the largest number of examples of the same class for each group. The rate is computed dividing the sum of these counts by the number of examples in the data set.

Grouping the examples that match the same values for all the selected features, if we call inconsistent examples to those that do not belong to the majority class of their group, Liu's measure can be expressed with the (2), as the proportion of these inconsistent examples in the total number of examples.

$$Inconsistency = \frac{\text{number of inconsistent examples}}{\text{number of examples}} \tag{2}$$

The group of these measures is usually referred to as consistency measures, though what this measure—and the later described IEP—really measure is inconsistency. In order to compare measures and work with them indistinctly it is necessary to establish the relation between consistency and inconsistency. Since it seems reasonable to think of consistency degree as the opposite value of inconsistency, we define the consistency as:

$$Consistency = 1 - Inconsistency \tag{3}$$

Some search algorithms, like Branch & Bound, require the measure being monotonic to get optimal or better performance. The monotonic property requires that if S_i, S_j are feature sets and $S_i \subset S_j$, then $M(S_i, D) \leq M(S_j, D)$, where M is the measure and D a data set. As well as all the other consistency measures included in this paper, this measure presents the monotonic property.

We can find an intuitive meaning for this measure. It could be seen as the classification accuracy that a memory classifier (also known as table classifier, or RAM, these are classifiers that keep all patterns and classify with the most frequent class for each pattern) will achieve on the data set with the given features. In other words, the probability that an example of the training data set would be correctly classified.

The computation of this measure could be done very fast using hash tables. A process to compute the measure on an example data set is shown in Fig. 2. First, the data set is projected to use only the features to evaluate. After that, all examples are introduced on a hash table. The elements introduced are the class values using as index the values of selected features. In this way, all examples are grouped according to the value of their selected features. Finally, the number of examples that do not belong to the majority class of their group are counted. It is easy to see that the efficiency, in the average case, of this process is in $O(n)$.

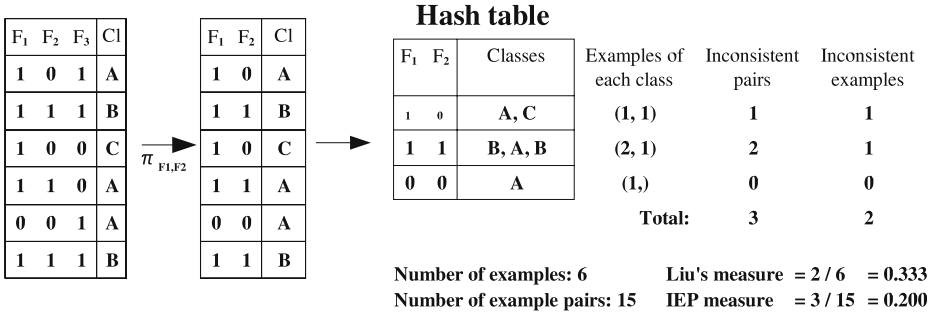

Fig. 2 Fast computation of inconsistent example Pairs and Liu's measures

Liu's measure is not defined for data sets with continuous features, but it could be used in combination with some discretization method, as was suggested by its authors. In a previous step the data set is discretized, and then the feature selection is applied. Once the features are selected, the learning algorithm may use the discretized features or their continuous version from the original data set. The rest of consistency measures also lack of a reasonable direct application on continuous data. Therefore, in the empirical study, we will use the related procedure to test the application of these measures on continuous data sets.

3.3 Rough set consistency measure

The following measure comes from the Rough Set Theory (Pawlak, 1991; Komorowski, Pawlak, Polkowski & Skowron, 1998; Polkowski & Skowron 1998), it is described in Pawlak (1991, Chapter 7.8). The measure has been used in discretization (Chmielewski & Grzymala-Busse, 1996), and it has even been compared with Liu's measure (Tay & Shen, 2002) in a discretization algorithm, but we have not found any previous work in which this measure had been used to guide a feature selection search.

We will just introduce the essential concepts of Rough Set Theory to describe the consistency measure. Let U denote the universe, i.e. the set of all examples from the data set. Let F denote the set of all features, and $S \subseteq F$ some selected features. The indiscernibility relation is defined as:

$$\mathrm{IND}(S) = \{(x, y) \in U \times U : \forall f \in S,\ f(x) = f(y)\} \tag{4}$$

This equivalence relation partitions U into equivalence classes, and the partition (set of equivalence classes) will be denoted $U/\mathrm{IND}(S)$.

For any subset of instances $X \subseteq U$, for example the set of examples belonging to a given class, the S lower approximation of X is defined by:

$$\underline{S}X = \bigcup \{Y \in U/\mathrm{IND}(S) : Y \subseteq X\} \tag{5}$$

If we take X as the set of examples of a class, $\underline{S}X$ represents those examples that could be consistently identified as members of that class using S features. We can repeat this for every class and define the positive region with the following equation,

where D denotes the set of dependent features, usually only one attribute identifying the class of the example.

$$POS_S(D) = \bigcup_{X \in U/\mathrm{IND}(D)} \underline{S}X \tag{6}$$

The degree of consistency is given by the proportion of these consistently classifiable examples in the total number of examples. The measure is shown in the following equation:

$$\gamma(S, D) = \frac{|POS_S(D)|}{|U|} = \sum_{X \in U/\mathrm{IND}(D)} \frac{|\underline{S}X|}{|U|} \tag{7}$$

The efficiency is the same as the one of Liu's measure. This measure can be computed with a similar process, but counting only those examples in groups where all examples belong to the same class.

While the other measures deal with what is left in a data set to be consistent, this measure looks to what is consistent. We can also think of these measure as more strict than Liu's, as those examples of the majority class in each group are not counted, if there is just one example from other class in the same group of indistinguishable examples.

As previously mentioned also this measure presents the monotonic property. It can be easily seen with the proof outlined in (8).

$$\forall c \in U/\mathrm{IND}(S \cup \{f\}) \exists \hat{c} \in U/\mathrm{IND}(S) : c \subseteq \hat{c} \longrightarrow$$
$$\forall X \in \mathcal{P}(F), \underline{S}X \subseteq \underline{S \cup \{f\}}X \longrightarrow \gamma(S, D) \leq \gamma(S \cup \{f\}, D) \tag{8}$$

3.4 Inconsistent example pairs measure

A consistent data set turns inconsistent when it happens to contain two examples with different class or concept value but the same values in all features. These two examples form an inconsistent example pair. In this way, a data set can be said to be more inconsistent, or that it shows a smaller degree of consistency, as more inconsistent example pairs appear in the data set. The measure we propose here uses the count of these pairs as an inconsistency measure.

The inconsistent example pairs have also been referred to as unsolved conflicts. In FOCUS terminology, a conflict is a pair of examples with different concept value. When the pair of examples that form a conflict have different values on some feature, the conflict is considered to be solved, and unsolved otherwise.

The unsolved conflict count has been used as search guide in Simple Greedy (Almuallim & Dietterich, 1994), and Set Cover (Dash, 1997), but to our knowledge it has never been defined as an independent measure, neither compared with other measures. We consider important to define a measure based on the count of inconsistent example pairs to fill a natural gap in consistency measures.

The count of inconsistent example pairs lies in a range between 0, when the data set is consistent, and the number of pairs of examples with different class, when no features are selected and so no pair may be distinguished. This makes the theoretical range of the measure to be the interval $[0, +\infty]$. Instead of using this count directly as the measure of inconsistency, it seems reasonable to make it proportional to the

Table 1 Bounds and interesting values of the inconsistency measures

Measure	General	Given a Data Set(DS)		Simplest DS	Hardest DS
		All feat.	$\emptyset$	$\emptyset$	$\emptyset$
Count of IEP	$[0, +\infty]$	$[IIP,$	$diffCl]$	0	*Pairs*
$\frac{Count\ of\ IEP}{No.pairsofdiff.Class}$	[0, 1]	$[\frac{IIP}{diffCl},$	1]	$\frac{0}{0}$ (Indet.)	1
$\frac{Count\ of\ IEP}{No.pairs}$	[0, 1]	$[\frac{IIP}{Pairs},$	$\frac{diffCl}{Pairs}]$	0	1
Liu's measure	[0, 1[	$[\frac{IIE}{N},$	$1 - Majority]$	0	$\frac{N-1}{N} \sim 1$
Rough Sets	[0, 1]	$[1 - \gamma,$	1 (0 if $\lvert Cl\rvert = 1$)]	0	1

DS Data Set, IIE No. Insolvable Inconsistent Ex., N $|DS|$ (No. examples), IIP No. Insolvable Inconsistent example Pairs, Cl Class feature, *Pairs* Total no. pairs in data set ($\frac{N(N-1)}{2}$), γ Rough Set Consistency, $diffCl$ = No. pairs of different class

data set, in order to make the measure comparable among data sets and bounded on a limited interval.

Table 1 shows some values of the measures. First row shows the count of inconsistent example pairs alone. The second one shows the proportion of inconsistent example pairs on the number of pairs with examples of different class. The third row corresponds to the proportion of inconsistent example pairs on the total number of pairs in the data set. The two final rows show the values for Liu's measure and rough set consistency measure as a mean of comparison.

In the general case the two options are bounded in the [0, 1] interval which is an advantage over the count alone. Since all the measures are monotonic, the range of a measure for a given data set will lie in the interval delimited by the values of the measure for the set of all features and the empty set. The specific minimum value is shown for each measure, but all of them agree to be 0 if the data set is consistent considering all features, what is not the case in presence of noise. The maximum value for the option dividing by the different class pairs is 1, using in this way the widest range possible in [0, 1] for all data sets. However the other option and Liu's measure provide a value that may be used as a measure of the a-priori (before selecting any feature) inconsistency or the inherent difficulty of a data set. In the case of Liu's consistency, this value is the well known Majority concept of a data set, i.e. the frequency of the most common class. Majority is commonly used as a minimum accuracy threshold acceptable for a classifier. In order to illustrate this, the values for two extreme cases of data sets are shown. The simplest data set is one with all instances belonging to the same class. It is consistent itself and there is no need to select any features, so it is reasonable to assign it a 0 as inconsistency degree. All measures satisfy this, except the one dividing by different class pairs that is undetermined and to be in accordance with its value for any given data set it should be defined to 1. On the other side, a data set in which every example belong to a different class will probably be more harder to find consistent hypothesis. This is the named hardest data set on the table, and all measures assign it the maximum value.

For a given data set, the difference between the three options is just a linear transformation that make the measures lie on the different identified intervals. Therefore the effects in guiding a search, or selecting a feature set, would be the same, but we consider the last option the most appropriate, as it allows the measure

to compare inconsistency degrees between different data sets. The measure is shown in (9).

$$Inconsistency = \frac{\text{number of inconsistent example pairs}}{\text{number of example pairs}} \tag{9}$$

The inconsistent example count measure is monotonic. It could be easily deduced from the following. An example pair that is consistent thanks to a feature in S_i will still be consistent with S_j as it is a superset. For this reason, the number of inconsistent example pairs could only decrease when features are added, so consistency measure will always be equal or greater.

Another interesting theoretical property of this measure was pointed out in Dash and Liu (1997). This measure, together with the simple greedy search algorithm that we will describe in the empirical study, resemble Johnson's approximation algorithm to Set Cover problem. In this way, it is guaranteed that a feature set with no more than $O(M \log N)$ features will be found, where N is the number of features in the data set and M is the size of the smallest consistent feature set.

An intuitive idea of this measure may be achieved thinking that it represents the probability that, on a given data set, with the selected features, we are able to distinguish two examples randomly chosen.

The fact that this measure works with the combination of all example pairs should not make us think that its computation is efficiently costly. In fact, its time and space efficiency in the average case can be as low as $O(n)$. The description of an algorithm using hash tables follows. An example of its application is shown in Fig. 2.

```
1.  #Algorithm to compute inconsistent example pairs measure
2.  ConsistencyMeasure(Dataset, SelectedFeatures)
3.      Hash = ∅
4.      For each Example in Dataset:
5.          Insert class(Example) into Hash at π_SelectedFeatures(Example)

6.      InconsistentExamplePairs = 0
7.      For each ClassList in Hash:
8.          InconsistentExamplePairs += number of all possible pairs
9.          of two different class values in ClassList

10.     n = |Dataset|
11.     return 1 − InconsistentExamplePairs / (n(n−1)/2)
```

Hash is a hash table in which every element included has a list (initially empty) of class values.

3.5 Other consistency based methods

We have aimed our study at those methods where measures can be separated from the search process. Nevertheless, in this section, we want to mention other consistency based feature selection methods that do not define independent measures. They rather define elaborated processes based on logic rules or heuristics, searching for a feature set that allows consistency. Anyway, we would like to point out that, although an extensive search has been performed, this is not an exhaustive list.

Since there are methods that could be used in feature selection, though they are not designed with feature selection in mind, we could have missed some of them.

Schlimmer (Schlimmer, 1993) describes an algorithm to induce logical determinations using the minimum possible number of features, that is in fact an embedded feature selection.

MIFES (Oliveira & Sangiovanni-Vicentelli, 1992) is an algorithm that can perform from feature selection, passing through construction of derived features, to constructive induction of the concept by creating a single feature that describes it. They present the concept of covering all the example pairs to achieve consistency with an intuitive matrix representation.

A recent approach (Boros et al., 2000) develop a logical analysis of data that include an embedded feature selection. It is based on the consistency concept and set covering, and it can handle with the proposed binarization discrete and numerical features, as well as imperfect data with missing values or errors.

There are some methods based on Rough Set Theory, like (Modrzejewski, 1993). A summary of the use of this theory to assess feature significance can be found in Chapter 7.1 of (Komorowski et al., 1998).

Zhong et al. (Zhong, Dong & Ohsuga, 2001) use the Rough Set Consistency measure multiplied by a factor to select features that generate simpler rules.

4 Empirical study

Our goal is to develop a rather wide empirical study, so we have considered classification problems as well as approximation problems. The type of values present in real problems are varied, discrete and continuous, so we evaluate the application of the measures in data sets with discrete features, continuous features and both mixed. The data sets chosen for the evaluation cover all the possible combinations between the problem and data types. To simplify the evaluation, the data sets are clustered into three groups: classification with discrete features, classification with continuous or mixed features, and regression with any type of features. In Table 2 the data sets used for each group are described. All data sets are available from the UCI machine learning repository (Hettich & Bay, 1999).

A discretization method is necessary to apply the consistency measures to continuous data and regression problems. Many discretization methods are available, but testing feature selection combined with all of them is outside the scope of this paper. Besides, we want to test feature selection without the interfering effect of elaborated discretization methods, that sometimes may even perform feature selection by themselves (Liu and Setiono, 1997). Therefore we will use a method that does not take into account feature interdependencies and behaves equal with all of them. The method used is three intervals equal frequency discretization, a practical and commonly used method that performs better that equi-distant interval discretization (Liu, Hussain,Tan & Dash, 2002). As a result of this, probably better results might be achieved using different numbers of intervals, or more elaborated discretization methods, specifically selected for each data set.

It should be pointed that discretization is used only in order to obtain the measure value and select features. It is not used in the learning algorithms, with the purpose of allowing them to get the most information possible from data.

Table 2 Data sets

Data set	No. examples	No. features	Prob. type	Features
house-votes84	435	17	Classification	Discrete
led24	1200	25	Classification	Discrete
lung-cancer	32	57	Classification	Discrete
lymphography	148	19	Classification	Discrete
mushrooms	8416	23	Classification	Discrete
promoters	106	59	Classification	Discrete
soybean	307	36	Classification	Discrete
splice	3190	62	Classification	Discrete
zoo	101	18	Classification	Discrete
anneal	898	39	Classification	Mixed
breast-cancer	286	10	Classification	Mixed
bupa	345	7	Classification	Continuous
credit	690	16	Classification	Mixed
ionosphere	351	33	Classification	Mixed
iris	150	5	Classification	Continuous
pima	768	9	Classification	Continuous
post-operative	90	9	Classification	Mixed
wdbc	569	21	Classification	Continuous
wine	178	14	Classification	Continuous
auto-mpg	398	9	Regression	Mixed
glass	214	10	Regression	Continuous
housing	506	14	Regression	Continuous
prostate	97	9	Regression	Continuous
servo	167	5	Regression	Discrete

The prediction algorithms we have used are the following three: the Naive Bayes classifier; an inducer of classification and regression trees, post-pruned using m-error estimate pruning method with parameter m set to 2.0, in order to achieve better generalization; and the kNN algorithm using 21 neighbours. We have used the implementations of these algorithms from the Orange data mining software (Demsar & Zupan, 2004). More details about the algorithms, as well as the source code, may be found on their documentation and web page.

4.1 Measures choosing a feature set

At first, we have studied how the measures behave in the selection of the best subset of features. This is one of the common uses we identified for the measures in the section describing the feature selection process. The idea is that high values of the measure, for a set of features, should correspond with high values of prediction accuracy.

The purpose of this experiment is to compare the values of the measures with the accuracy achieved using a learning method, using the same feature set. It is not possible to evaluate all the subsets of features, at least with most of the data sets we are using. This is because of the large number of possible combinations of features. For this reason, we have taken a sample of some feature sets from the whole powerset of all features. To have a representation of the whole space—as if we took the sets

randomly there would be a much higher probability of taking medium sized sets—we have taken a fixed number of random sets of every size. The fixed number of sets is chosen so that the total number of sets is over 100. Since there is only one feature set with size equal to all features, we have not taken this size into the total count to avoid including the same set multiple times, but we have always included the set with all features, because we think it is important to have it included in the comparative.

To have good estimations of the accuracy of the algorithms ten fold cross-validation has been used for every evaluated set. Result are summarized through the average of the ten folds. They are shown in Table 3. Accuracy is measured as the percentage of correct classification in the classification problems, and as the mean squared error(MSE) in the regression problems.

As an illustration of this experiment, in Fig. 3 there is a scatter plot of the evaluation measures and the classification accuracies, on the soybean classification problem. It can be seen how the relation among the accuracy of the different

Table 3 Correlation of each measure with the learning algorithms

Data set	LIU			IEP			RSC		
	NB	Tree	kNN	NB	Tree	kNN	NB	Tree	kNN
Classification with discrete features									
house-votes84	0.95	0.97	0.97	0.89	0.87	0.91	0.58	0.73	0.72
led24	0.85	0.82	0.81	0.49	0.48	0.49	0.86	0.83	0.82
lung-cancer	0.31	0.06	0.33	0.17	−0.01	0.22	0.36	0.16	0.37
lymphography	−0.25	−0.54	0.01	−0.22	−0.36	−0.07	−0.10	−0.42	0.16
mushrooms	0.90	0.99	0.99	0.72	0.82	0.86	0.88	0.93	0.93
promoters	0.56	0.34	0.63	0.49	0.33	0.61	0.48	0.39	0.50
soybean	0.98	0.98	0.97	0.76	0.76	0.75	0.96	0.96	0.96
splice	0.65	0.63	0.74	0.37	0.35	0.43	0.64	0.60	0.72
zoo	0.99	0.99	0.98	0.88	0.88	0.90	0.90	0.91	0.88
Average	0.66	0.58	0.71	0.51	0.46	0.57	0.62	0.57	0.67
Classification with continuous or mixed features									
adult	0.72	0.91	0.95	0.44	0.54	0.57	0.48	0.43	0.58
anneal	0.90	0.85	0.91	0.56	0.52	0.66	0.89	0.87	0.91
breast-cancer	0.61	−0.33	0.57	0.4	−0.15	0.43	0.40	−0.54	0.47
bupa	0.86	0.71	0.64	0.66	0.58	0.64	0.83	0.62	0.60
credit	0.92	0.86	0.87	0.72	0.63	0.67	0.78	0.65	0.72
ionosphere	0.82	0.85	0.24	0.63	0.7	0.42	0.87	0.78	0.29
iris	0.98	0.99	0.99	0.92	0.92	0.95	0.71	0.73	0.75
pima	0.80	0.62	0.82	0.65	0.27	0.65	0.71	0.63	0.72
wdbc	0.92	0.92	0.95	0.83	0.82	0.86	0.83	0.84	0.86
wine	0.97	0.96	0.96	0.78	0.77	0.77	0.91	0.91	0.89
Average	0.85	0.73	0.79	0.66	0.56	0.66	0.74	0.59	0.68
Regression									
auto-mpg	–	0.08	−0.50	–	0.06	−0.31	–	0.25	−0.5
glass	–	0.71	−0.71	–	−0.67	−0.80	–	−0.64	−0.59
housing	–	−0.94	−0.91	–	−0.71	−0.83	–	−0.72	−0.61
prostate	–	−0.03	−0.77	–	0.15	−0.71	–	0.21	−0.59
servo	–	−0.71	−0.63	–	−0.22	−0.12	–	−0.59	−0.44
Average	–	−0.46	−0.70	–	−0.28	−0.55	–	−0.30	−0.55

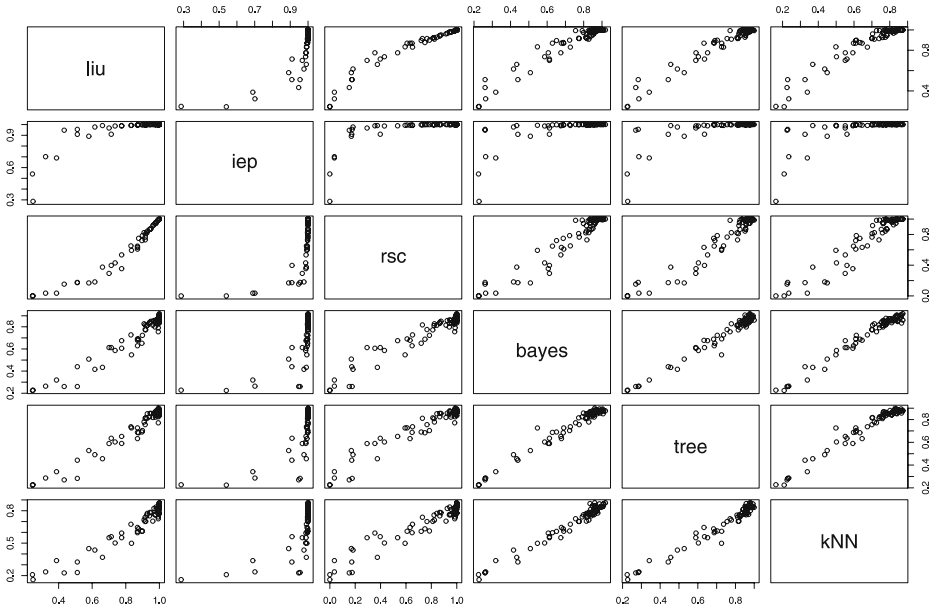

Fig. 3 Scatter plot of measures and classification accuracy for soybean data set

classifiers is mostly linear, having all of them a similar behaviour with each feature set given. The relation among Liu, RSC measure and the three classifiers accuracy is nearly linear, showing that these measures are good predictors for the accuracy of a given feature set in soybean problem. On the other side, IEP measure does not show a strong linear relation with accuracies, but there is a tendency to give high values on the feature sets that perform well on classification.

Table 3 shows the correlation factors between the measures and the learning methods accuracy. The results are shown for all the data sets considered in the three groups, as well as the mean of correlations for every group. The better values for correlation factor are those near 1 in the classification problems, as we expect positive correlation. On the other side, we expect negative correlation on regression problems, as the better values for MSE are the smaller ones.

We can see that there is generally high correlation in the classification problems, except lymphography. The correlation between measures and accuracy of learning models is high for regression problems when applying the kNN, while its rather low for regression trees.

With just a few exceptions, the correlation of Liu's measure with learners accuracy is a bit higher than the correlation between Rough Set Consistency measure (RSC) and learners. Inconsistent Example Pairs measure (IEP) shows a lower correlation, indicating the relation is less linear as we have seen in the scatter plot.

Obviously, the wrapper approach, which uses the accuracy of the learning method as measure, will get the best results in all cases with a correlation factor of 1.

However, in order to select a good feature set, it is not necessary to have linear correlation with accuracy. The condition that the measure should agree with is that for any two feature sets f_1, f_2, if their associated accuracies are $a_1 < a_2$ the measures

Table 4 Accuracy achieved with the different combinations of feature selectors and learners

Data set	Naive Bayes					Tree					kNN				
	No	Liu	IEP	RSC	Wr	No	Liu	IEP	RSC	Wr	No	Liu	IEP	RSC	Wr
house-votes84	90.1	91.5	92.9	93.6	95.4	96.3	96.1	96.3	96.6	94.7	93.8	94.7	94.0	94.9	95.9
led24	75.8	76.0	55.0	76.1	75.8	71.9	72.5	50.2	71.5	75.1	62.1	66.1	42.8	65.4	75.8
lung-cancer	55.8	50.0	49.2	58.3	46.7	38.3	65.8	44.2	55.8	48.3	46.7	55.8	39.2	58.3	46.7
lymphography	47.1	45.9	46.5	43.2	48.5	41.9	46.4	45.1	47.3	47.2	43.2	49.2	45.1	46.0	45.8
mushrooms	99.7	99.3	99.0	99.5	100	100	99.9	100	100	100	100	99.9	100	100	100
promoters	86.8	84.7	86.0	88.6	85.7	78.6	84.1	80.5	86.0	79.4	83.7	89.6	82.2	89.6	87.6
soybean	91.2	78.2	73.0	82.8	88.0	89.2	79.7	70.0	85.3	89.5	85.3	70.0	60.9	75.9	87.6
splice	95.6	94.3	67.7	85.7	95.8	93.8	94.1	66.8	85.2	94.1	83.5	86.9	65.6	79.6	89.1
zoo	92.0	97.0	95.0	96.0	96.0	96.0	95.0	94.0	95.0	95.0	94.1	83.1	86.1	85.1	92.1
anneal	95.9	94.4	92.2	90.9	96.3	96.4	96.3	93.7	94.0	97.1	90.7	92.8	91.1	89.1	97.6
breast-cancer	74.8	74.5	74.8	74.5	73.4	68.9	68.6	65.0	64.6	73.1	71.6	72.7	73.1	74.4	72.0
bupa	68.7	68.7	68.7	68.7	66.1	61.7	62.9	65.8	65.8	62.6	63.8	63.8	64.3	64.3	67.9
credit	86.2	84.5	86.4	85.5	83.9	84.1	84.4	85.4	85.7	83.9	86.7	84.9	85.2	84.9	86.2
ionosphere	90.9	91.5	91.5	89.5	92.0	93.7	90.0	93.5	89.5	92.3	82.4	87.2	85.2	87.2	89.5
iris	96.7	96.7	96.7	96.7	94.7	96.0	95.3	96.0	96.0	94.7	97.7	96.7	97.7	97.7	95.3
pima	76.2	76.2	76.2	76.2	76.8	71.2	71.2	71.2	71.2	67.1	74.7	74.7	74.7	74.7	74.0
post-operative	63.3	63.3	66.7	66.7	70.0	61.1	57.7	63.3	63.3	64.4	68.9	67.8	71.1	71.1	65.6
wdcb	95.4	95.1	94.6	94.0	96.7	94.2	93.0	93.3	92.5	94.0	97.2	96.0	96.3	96.1	96.5
wine	98.9	97.2	97.2	97.2	96.0	92.1	94.4	96.1	93.8	93.2	96.6	97.2	96.1	96.6	97.7
auto-mpg	–	–	–	–	–	49.0	48.5	48.5	48.5	15.0	10.5	17.6	17.6	17.6	10.3
glass	–	–	–	–	–	1.75	2.03	1.67	1.87	1.89	1.13	1.13	1.13	1.13	1.20
housing	–	–	–	–	–	22.7	22.2	21.5	22.0	21.2	23.7	22.4	22.0	22.7	13.3
prostate	–	–	–	–	–	1.32	1.33	1.33	1.39	1.42	0.92	0.93	0.87	0.87	0.80
servo	–	–	–	–	–	0.77	0.77	0.77	0.77	0.77	1.14	1.14	1.14	1.14	1.24

of feature sets should be $m_1 < m_2$, and we can imagine that having this condition strictly is only important in those feature sets with higher accuracies, as these are the sets that are going to be selected at the end of the search. Therefore this is complex to evaluate, and it seems reasonable to test the measure behavior in a complete application process to get a complete idea of its performance.

Besides, the correlation does not say anything about the capacity of the measure to guide the search. To overcome the limitations of just studying the measures alone, we have tested the measures in a complete environment, with a search process and classification with the feature set chosen.

4.2 Measures guiding search

We have chosen to utilize a greedy search process. This allows us to explore the potential use of the measures guiding the search process. The search process used is similar to Simple Greedy (Almuallim & Dietterich, 1994), Hill-climbing (Kohavi & John, 1997), and Set Cover based (Dash, 1997; Dash & Liu, 2003) already used in feature selection, and commonly used in statistics.

The starting point is the empty set. The idea is, given a feature set, to explore all the resulting sets of adding one of the available features, and continue with the one that gets best results on evaluation function. The stopping criterion is to stop when we reach the set with all features. At the end, the feature set visited with the best measure is returned.

The time efficiency of the search process is $O(n^2)$, where n is the total number of features. This is quite reasonable for most problems. It may also be speeded up with a more restrictive stopping criterion. For example, it can be stopped when, at a given step, no increase might be achieved on the evaluation function.

We have applied the feature selection process using each of the measures with the three learning algorithms. This process has been repeated ten times in order to apply ten fold cross-validation, with feature selection performed independently on each fold. The results shown are the averages of the accuracy achieved on the ten folds.

The wrapper measure uses internally another process of ten fold cross-validation to evaluate accuracy of a learner with the feature set in consideration. This is obviously performed on the training part of the current fold of the main process.

Table 4 shows the accuracy results grouped by the learning algorithms with which the feature selection is combined, and the different data set groups. As the Naive Bayes learner can not be applied to regression problems its cells are left empty. In Table 5 the number of features selected are shown. The first column indicates the number of features of the data set, that may result convenient to compare. As the consistency measures are independent of the learning algorithm, their number of features is shown in common for all learners, while the number of features selected by the wrapper approach is shown for every learner.

The wrapper approach obtains only slightly better accuracy results than consistency measures on average, around 1–2% higher accuracy in classification problems. Nevertheless, this is not a very significative difference, and it is also interesting to mention that consistency measures achieved better accuracy on some data sets. Therefore consistency measures are a reliable competitor of the wrapper approach. Both approaches, the wrapper, and filtering with consistency measures, have im-

Table 5 Average number of features used in each method

Data set	NB/Tree/kNN				NB	Tree	kNN
	No	LIU	IEP	RSC	Wr	Wr	Wr
house-votes84	16	10.6	9.3	10.3	3.1	8.3	4.2
led24	24	17.9	17.3	17.8	10.3	8.0	8.5
lung-cancer	56	4.3	4.1	5.1	14.0	4.9	12.9
lymphography	18	8.2	7.9	8.5	5.3	5.5	5.3
mushrooms	22	4.8	4.0	5.0	12.9	4.8	4.8
promoters	57	4.2	4.0	4.0	14.0	10.6	26.8
soybean	35	10.1	8.7	12.0	20.0	14.8	20.5
splice	60	10.6	9.6	10.3	35.5	16.1	6.8
zoo	16	4.9	4.9	5.1	7.2	5.2	11.0
anneal	38	23.1	13.4	15.7	27.6	19.0	15.7
breast-cancer	9	8.0	8.2	8.4	3.9	2.8	5.4
bupa	6	6.0	6.0	6.0	4.0	4.6	4.0
credit	15	11.4	10.6	11.1	9.1	5.9	7.0
ionosphere	32	9.7	8.9	9.0	10.9	15.5	4.3
iris	4	3.1	4.0	4.0	2.4	1.9	1.6
pima	8	8.0	8.0	8.0	4.3	3.6	5.6
post-operative	8	7.9	7.9	7.9	0.5	1.7	2.1
wdcb	20	9.1	9.6	9.2	9.9	7.1	10.8
wine	13	5.2	5.3	5.5	6.0	4.9	6.3
auto-mpg	8	2.9	2.9	2.9	–	5.4	5.1
glass	9	8.8	8.8	8.8	–	4.6	4.7
housing	13	12.0	11.8	12.6	–	8.8	6.8
prostate	8	7.2	7.2	7.3	–	3.3	5.3
servo	4	4.0	4.0	4.0	–	3.5	3.5

proved accuracy of learners on many data sets, confirming in this way the usefulness of feature selection.

Comparing Inconsistent Examples Pairs (IEP) measure with Liu's measure we can see that they yield very similar accuracy results, except in some cases like the splice data set, where IEP reduces the number of features by one more than Liu's measure and renders much worse accuracy.

The results are varied across the data sets. We can check that there are some data sets in which there are significative differences among measures. However taking into account all data sets no clear winner arises. Performing a paired t test on the differences for each pair of measures, renders no significative difference in accuracy among the learning algorithms.

Another important point in feature selection methods is the number of features they select. The consistency measures, and specially IEP, achieve considerably bigger reductions than the wrapper approach on the classification problems with discrete data. For example, in promoters and mushrooms data set, the number of features is reduced to less than a fourth of its original size, while accuracy is kept in a similar level to that achieved by the wrapper approach. On classification with continuous data problems, the differences are not so high, with the wrapper approach reducing more than the others. Finally, on regression data sets, the wrapper approach show the best results not only on feature reduction but also in accuracy.

The running time of the different algorithms has also been recorded, but we do not consider appropriate to use these times to strictly speak about differences among them because they have been obtained with different external factors. One of these factors is that the learning algorithms are implemented in C++—that is the wrapper measure is compiled—while the other measures are implemented in Python, which is an interpreted language. In this way, the implementation is supposed to give an advantage to wrapper measure. Nevertheless, in general, we can say that all measures perform quickly on small data sets. However, as expected by the theoretic efficiency, running time of discrete measures grows slowly with data set size, while the wrapper approach time grows much faster.

5 Conclusions

We have presented a survey on the use of data set consistency measures for feature selection. To begin with, the feature selection problem and their main applications are reviewed. After that, based on a previous work on categorization of feature selection methods, we have introduced a modular decomposition of the feature selection process illustrating its relation with some well known methods. We hope this modular view can provide new views for researching in feature selection, as well as a skeleton for possible new methods. Then, our study is centered on the evaluation function—one of the modules of the decomposition—and more precisely on those measures based on consistency.

The state of the art of consistency measures for feature selection is reviewed, describing the three identified measures: the monotonic consistency measure proposed by Liu et al. (1998) for feature selection, the generic consistency measure from Rough Set Theory, and one measure defined from the ideas of some previous consistency based feature selection methods, that we consider necessary to define as a measure to fill a natural gap in this field. All these measures are carefully analyzed and compared, considering their properties and interpretation. We have identified their limit values and their use comparing data sets, revealing the relation between Liu's measure and the majority concept. We have also presented a review of other feature selection methods based on consistency as they are the basis of measures. Finally, an empirical evaluation of these measures and the wrapper approach has been performed, paying special attention to accuracy and reduction of the number of features.

We have shown that consistency measures can be very useful in many feature selection problems for the following reasons. First, they can achieve similar accuracy results than the wrapper approach, while being much more efficient. Second, they can achieve a higher feature reduction. And finally, being independent of the classifier used, they may be more practical in some circumstances, for example using various algorithms on the same problem, or assessing experts. For these reasons, we can conclude that the use of the filter approach for feature selection is an interesting choice. When efficiency is a requirement, the wrapper approach is usually not suitable, but the filter approach with consistency measures is your choice. Moreover, even in situations where the wrapper approach could be used, the filter approach can render more accurate results.

The three consistency measures compared achieve pretty similar results, thus making a choice among them is difficult. In case we are interested in a high feature

reduction for a classification problem, we may choose Inconsistent Example Pairs measure, while if we are interested in maximal accuracy Liu's measure may be a better choice. As the three measures are very efficient, it also possible to apply all of them and to take the one which fits best to our problem, probably in the same time that it would take to run other measures.

The results suggest that consideration of continuous features and regression problems deserve a deeper study to improve accuracy, because while the consistency measures provide a much more efficient way of selecting features than the wrapper approach, the accuracy is slightly worse using the former approach. Finally, there is an open field of research in the combination of feature selection and discretization.

Acknowledgements This work was supported by the spanish Ministerio de Ciencia y Tecnología under projects TIC2003-04650 and TIN2004-07236.

References

Almuallim, H., & Dietterich, T. G. (1991). Learning with many irrelevant features. In *Proceedings of the ninth national conference on artificial intelligence* (*AAAI-91*), *Anaheim, CA, vol. 2* (pp. 547–552). Menlo Park, CA: AAAI Press.

Almuallim, H., & Dietterich, T. G. (1994). Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence, 69*(1, 2), 279–305.

Arauzo Azofra, A., Benitez, J. M., & Castro, J. L. (2003a). C-FOCUS: A continuous extension of FOCUS. In *Proceedings of the 7th online world conference on soft computing in industrial applications* (pp. 225–232).

Arauzo Azofra, A., Benitez-Sanchez, J. M., & Castro-Peña, J. L. (2003b). A feature selection algorithm with fuzzy information. In *Proceedings of the 10th IFSA world congress* (pp. 220–223).

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence, 97*, 245–271.

Boros, E., Hammer, P. L., Ibaraki, T., Kogan, A., Mayoraz, E., & Muchnik, I. (2000). An implementation of logical analysis of data. *IEEE Transactions on Knowledge Discovery and Data Engineering, 12*(2), 292–306.

Brill, F. Z., Brown, D. E., & Martin, W. N. (1992). Fast genetic selection of features for neural network classifiers. *IEEE Transactions on Neural Networks, 3*(2), 324–328.

Chmielewski, M. R., & Grzymala-Busse, J. W. (1996). Global discretization of constinuous attributes as preprocessing for machine learning. *International Journal of Approximate Reasoning, 15*(4), 319–331.

Dash, M. (1997). Feature selection via set cover. In *IEEE Knowledge and Data Engineering Exchange Workshop*.

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis, 1*(1–4), 131–156.

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

Demsar, J., & Zupan, B. (2004). *Orange: From experimental machine learning to interactive data mining*. (White paper) http://www.ailab.si/orange.

Hettich, S., & Bay, S. D. (1999). *The uci kdd archive*. http://kdd.ics.uci.edu/.

Jain, A., & Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. *IEEE transactions on pattern analysis and machine intelligence, 19*(2), 153–158.

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *International conference on machine learning*, (pp. 121–129). Journal version in AIJ, available at http://citeseer.nj.nec.com/13663.html.

Kira, K., & Rendell, L. A. (1992). A practical approach to feature selection. In *Proceedings of the ninth international workshop on machine learning* (pp. 249–256). San MAteo, CA, Morgan Kaufmann.

Kohavi, R. (1994). Feature subset selection as search with probabilistic estimates. In *AAAI fall symposium on relevance* (pp. 122–126).

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

Komorowski, J., Pawlak, Z., Polkowski, L., & Skowron, A. (1998). Rough sets: A tutorial. In S. K. Paland, & A. Skowron (Eds.) Rough-fuzzy hybridization: A new trend in decision-making (pp. 3–98). Singapore: Springer.

Kudo, M., & Sklansky, J. (2000). Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition, 33*(1), 25–41.

Langley, P. (1994). Selection of relevant features in machine learning. In *Procedings of the AAAI fall symposium on relevance*, New Orleans, LA. Menlo Park, CA: AAAI Press.

Liu, H., Hussain, F., Tan, C. L., & Dash, M. (2002). Discretization: An enabling technique. *Data Mining and Knowledge Discovery, 6*, 393–423.

Liu, H., Motoda, H., & Dash, M. (1998). A monotonic measure for optimal feature selection. In *European conference on machine learning* (pp. 101–106).

Liu, H., & Setiono, R. (1997). Feature selection via discretization. *Knowledge and Data Engineering, 9*(4), 642–645.

Modrzejewski, M. (1993). Feature selection using rough sets theory. In *Proceedings of the European conference on machine learning* (pp. 213–216).

Oliveira, A., & Sangiovanni-Vicentelli, A. (1992). Constructive induction using a non-greedy strategy for feature selection. In *Proceedings of ninth international conference on machine learning*, Aberdeen, Scotland (pp. 355–360). San Mateo, CA: Morgan Kaufmann.

Pawlak, Z. (1991). *Rough sets, theoretical aspects of reasoning about data*. Boston, MA: Kluwer.

Polkowski, L., & Skowron, A., (Eds.) (1998). *Rough sets in knowledge discovery*. Heidelberg: Physica Verlag.

Schlimmer, J. (1993). Efficiently inducing determinations: A complete and systematic search algorithm that uses optimal pruning. In *Proceedings of tenth international conference on machine learning* (pp. 289–290).

Somol, P. & Pudil, P. (2004). Fast branch & bound algorithms for optimal feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(7), 900–912.

Tay, F. E. H., & Shen, L. (2002). A modified chi2 algorithm for discretization. *Knowledge and Data Engineering, 14*(3), 666–670.

Wettschereck, D., Aha, D. W., & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review, 11*(1-5), 273–314.

Zhong, N., Dong, J., & Ohsuga, S. (2001). Using rough sets with heuristics for feature selection. *Journal of Intelligent Information Systems, 16*(3), 199–214.