# Improving Lazy Attribute Selection

Rafael B. Pereira[1], Alexandre Plastino[1], Bianca Zadrozny[2], Luiz H. de C. Merschmann[3], Alex A. Freitas[4]

[1] Universidade Federal Fluminense, Brazil
{rbarros, plastino}@ic.uff.br
[2] IBM Research, Brazil
biancaz@br.ibm.com
[3] Universidade Federal de Ouro Preto, Brazil
luizhenrique@iceb.ufop.br
[4] University of Kent, United Kingdom
A.A.Freitas@kent.ac.uk

**Abstract.** Attribute selection is a data preprocessing step which aims at identifying relevant attributes for a target data mining task – specifically in this article, the classification task. Previously, we have proposed a new attribute selection strategy – based on a lazy learning approach – which postpones the identification of relevant attributes until an instance is submitted for classification. Experimental results showed the effectiveness of the technique, as in most cases it improved the accuracy of classification, when compared with the analogous eager attribute selection approach performed as a data preprocessing step. However, in the previously proposed approach, the performance of the classifier depends on the number of attributes selected, which is a user-defined parameter. In practice, it may be difficult to select a proper value for this parameter, that is, the value that produces the best performance for the classification task. In this article, aiming to overcome this drawback, we propose two approaches to be used coupled with lazy attribute selection technique: one that tries to identify, in a wrapper-based manner, the appropriate number of attributes to be selected and another that combines, in a voting approach, different numbers of attributes. Experimental results show the effectiveness of the proposed techniques. The assessment of these approaches confirms that the lazy learning paradigm can be compatible with traditional methods and appropriate for a large number of applications.

Categories and Subject Descriptors: H. Information Systems [**H.2 Database Management**]: H.2.8 Database Applications—*Data mining*

Keywords: Attribute Selection, Classification, Lazy Learning

## 1. INTRODUCTION

According to [Guyon and Elisseeff 2006], attribute selection techniques are primarily employed to identify relevant and informative attributes in a dataset for the classification task. In general, besides this main goal, there are other important motivations: the improvement of a classifier's predictive accuracy, the reduction and simplification of the dataset, the acceleration of the classification task, the simplification of the generated classification model, and others. In this article, the main motivation of the attribute selection study is the improvement in classification accuracy.

The performance of a classification method is closely related to the inherent quality of the training data. Redundant and irrelevant attributes may not only decrease the classifier's accuracy but also make the process of building the model or the execution of the classification algorithm slower. In order to avoid these drawbacks, attribute selection techniques are usually applied for removing from the training set attributes that do not contribute to, or even decrease, the classification performance [Guyon et al. 2006; Liu and Motoda 2008a].

The development of this work was supported by CAPES, CNPq and FAPERJ research grants.

Attribute selection techniques can generally be categorized into three categories: embedded, wrapper or filter [Liu and Motoda 2008b]. Embedded strategies are directly incorporated into the algorithm responsible for the induction of a classification model. Decision tree induction algorithms can be viewed as having an embedded technique, since they internally select a subset of attributes that will label the nodes of the generated tree. Wrapper and filter strategies are performed in a preprocessing phase and they search for the most suitable attribute set to be used by the classification algorithm or by the classification model inducer. In wrapper selection, the adopted classification algorithm itself is used to evaluate the quality of candidate attribute subsets, while in filter selection, attribute quality is evaluated independently from the classification algorithm using a measure which takes into account the attribute and class label distributions. There are also hybrid strategies which try to combine both wrapper and filter approaches [Liu and Yu 2005].

Filter strategies are commonly divided into two categories. Techniques of the first category, as exemplified by Information Gain Attribute Ranking [Yang and Pedersen 1997] and Relief [Kira and Rendell 1992; Kononenko 1994], evaluate each attribute individually and select the best ones. Attributes that provide a good class separation will be ranked higher and therefore be chosen. The second category is characterized by techniques which evaluate subsets of attributes, searching, heuristically, for the best subset. Two well-known strategies of this group are Correlation-based Feature Selection [Hall 2000] and Consistency-based Feature Selection [Liu and Setiono 1996].

As described, traditionally, attribute selection techniques are executed as a data preprocessing step, making their results definitive from that point on. In [Pereira et al. 2011], we have proposed a new filter attribute selection strategy – based on a lazy learning approach – which postpones the identification of relevant attributes until an instance is submitted for classification. It relies on the hypothesis that taking into account the attribute values of an instance to be classified may contribute to identifying the best attributes for the correct classification of that particular instance. Therefore, for different instances to be classified, it is possible to select distinct subsets of attributes, each one customized for that particular instance. Experimental results using the well-known k-Nearest Neighbors ($k$-NN) and Naive Bayes classification techniques, over 40 different datasets from the UCI Machine Learning Repository [Asuncion and Newman 2007] and five large datasets from the NIPS 2003 feature selection challenge [Guyon et al. 2004], show the effectiveness of delaying attribute selection to classification time. The proposed lazy technique in most cases improves the accuracy of classification, when compared with the analogous attribute selection approach performed as a data preprocessing step.

This lazy attribute selection technique requires, as a user-defined parameter, the number of attributes to be selected. This is not a specific requirement of the lazy strategy, since traditional attribute selection techniques of the filter category generally also have this necessity [Duch 2006]. Nevertheless, choosing the proper number of attributes that will produce the best classification performance is not a simple task. Therefore, it is desirable to automate the choice of the number of attributes to be selected by the lazy selection technique previously proposed in [Pereira et al. 2011].

In this article, we propose two procedures to eliminate the need of manually choosing a proper number of attributes. The first one is a wrapper-based approach which employs the target classification technique and the training instances of a dataset with the purpose of identifying the best number of attributes for that dataset and classification technique. The second approach utilizes a voting mechanism that combines several classifiers, each one built using the target classification technique over the dataset with a different number of attributes selected by the lazy selection approach.

The remaining of this article is organized as follows. In Section 2, we describe the lazy attribute selection strategy, proposed in [Pereira et al. 2011], in detail. The wrapper-based technique is presented in Section 3, along with computations experiments over a large number of datasets. In Section 4, the voting strategy is proposed and its experimental results are reported. In Section 5, we conduct an analysis on the computational time used up by each of the proposed techniques. Finally, in Section 6, we make our concluding remarks and point to directions for future work.

## 2. LAZY ATTRIBUTE SELECTION

In conventional attribute selection strategies, attributes are selected in a preprocessing phase. The attributes which are not selected are discarded from the dataset and no longer participate in the classification process.

In [Pereira et al. 2011], we proposed a lazy attribute selection strategy based on the hypothesis that postponing the selection of attributes to the moment at which an instance is submitted for classification can contribute to identifying the best attributes for the correct classification of that particular instance. For each different instance to be classified, it is possible to select a distinct and more appropriate subset of attributes to classify it.

Below we give a toy example to illustrate the fact that the classification of certain instances could take advantage of attributes discarded by conventional attribute selection strategies. In addition, some of the attributes selected by conventional strategies may be irrelevant for the classification of other instances. In other words, the example illustrates that attributes may be useful or not depending on the attribute values of the instance to be classified. In Table I, the same data set, composed of three attributes – $X$, $Y$, and the class $C$ – is represented twice. The left occurrence is ordered by the values of $X$ and the right one is ordered by the values of $Y$. It can be observed in the left occurrence that the values of $X$ are strongly correlated with the class values making it a useful attribute. Only value 4 is not indicative of a unique class value.

Furthermore, as shown in the right occurrence, attribute $Y$ would probably be discarded since in general its values do not correlate well with the class values.

However, there is a strong correlation between the value 4 of attribute $Y$ and the class value $B$, which would be lost if this attribute were discarded. The classification of an element with value 4 in the $Y$ attribute would clearly take advantage of the presence of this attribute.

A conventional attribute selection strategy – which, from now on, we refer to as an "eager" selection strategy – is likely to lose crucial information that could be used for generating accurate predictions. In the example, an eager selection strategy is likely to select attribute $X$ in detriment of attribute $Y$, regardless of the instances that are submitted for classification.

Hence, the main motivation behind the proposed lazy attribute selection is the ability to assess the attribute values of the instance to be classified, and use this information to select attributes that better discriminate the classes for those particular values. As a result, for each instance we can select attributes that are useful for classifying that particular instance.

In [Pereira et al. 2011], we implemented a specific lazy attribute selection based on a filter strategy which evaluates each attribute individually and selects the best ones. Attributes that provide a good class separation are ranked higher and therefore chosen. The implemented version employed an

Table I. DataSet Example

| Dataset Sorted by X | | | Dataset Sorted by Y | | |
|---|---|---|---|---|---|
| – X – | – Y – | – C – | – X – | – Y – | – C – |
| 1 | 2 | B | 2 | 1 | A |
| 1 | 3 | B | 3 | 1 | B |
| 1 | 4 | B | 4 | 1 | A |
| 2 | 1 | A | 1 | 2 | B |
| 2 | 2 | A | 2 | 2 | A |
| 2 | 3 | A | 3 | 2 | B |
| 3 | 1 | B | 1 | 3 | B |
| 3 | 2 | B | 2 | 3 | A |
| 3 | 4 | B | 4 | 3 | B |
| 4 | 1 | A | 1 | 4 | B |
| 4 | 3 | B | 3 | 4 | B |
| 4 | 4 | B | 4 | 4 | B |

adaptation, for the lazy approach, of the entropy measure [Quinlan 1986] to determine the relevance of each attribute. The entropy concept is commonly used as measure of attribute relevance in eager and filter strategies that evaluate attributes individually [Yang and Pedersen 1997], and this method has the advantage of being fast. It also requires discrete attribute values.

The lazy attribute selection technique was defined as follows. Let $D(A_1, A_2, ..., A_n, C)$, $n \geq 1$, be a dataset with $n + 1$ discrete attributes, where $C$ is the class attribute. Let $\{c_1, c_2, ..., c_m\}$, $m \geq 2$, be the domain of the class attribute $C$. The entropy of the class distribution in $D$, represented by $Ent(D)$, is defined by

$$Ent(D) = -\sum_{i=1}^{m}[p_i * log_2(p_i)], \tag{1}$$

where $p_i$ is the probability that an arbitrary instance in $D$ belongs to class $c_i$.

Let $\{a_{j1}, a_{j2}, ..., a_{jk_j}\}$, $k_j \geq 1$, be the domain of the attribute $A_j$, $1 \leq j \leq n$. Let $D_{ji}$, $1 \leq j \leq n$ and $1 \leq i \leq k_j$, be the partition of $D$ composed of all instances whose value of $A_j$ is equal to $a_{ji}$. The entropy of the class distribution in $D$, restricted to the values of attribute $A_j$, $1 \leq j \leq n$, represented by $Ent(D, A_j)$, is defined by

$$Ent(D, A_j) = \sum_{i=1}^{k_j}[(\frac{|D_{ji}|}{|D|}) * Ent(D_{ji})]. \tag{2}$$

Thus we define the entropy of the class distribution in $D$, restricted to the value $a_{ji}$, $1 \leq i \leq k_j$, of attribute $A_j$, $1 \leq j \leq n$, represented by $Ent(D, A_j, a_{ji})$, as follows:

$$Ent(D, A_j, a_{ji}) = Ent(D_{ji}). \tag{3}$$

The concept defined in Equation 2 is used by the eager strategy known as Information Gain Attribute Ranking [Yang and Pedersen 1997] to measure the ability of an attribute to discriminate between class values. Equation 3 will be used in our proposed lazy selection strategy to measure the class discrimination ability of a specific value $a_{ji}$ of a particular attribute $A_j$. The closer the entropy $Ent(D, A_j, a_{ji})$ is to zero, the greater the chance that the value $a_{ji}$ of attribute $A_j$ is a good class discriminator.

The input parameters of the lazy strategy are: a dataset $D(A_1, A_2, ..., A_n, C)$, an instance to be classified with its attribute values $I[v_1, v_2, ..., v_n]$; and a number $r$, $1 \leq r < n$, which represents the number of attributes to be selected.

In order to select the $r$ best attributes to classify $I$, we proposed an evaluation of the $n$ attributes based on a lazy measure ($LazyEnt$), defined in Equation 4, which states that, for each attribute $A_j$, if the discrimination ability of the specific value $v_j$ of $A_j$ ($Ent(D, A_j, v_j)$) is better than (less than) the overall discrimination ability of attribute $A_j$ ($Ent(D, A_j)$) then the former will be considered for ranking $A_j$. The choice of considering the minimum value from both the entropy of the specific value and the overall entropy of the attribute was motivated by the fact that some instances may not have any relevant attributes considering their particular values. In this case, attributes with the best overall discrimination ability will be selected.

Then, the measure proposed to assess the quality of each attribute $A_j$ was defined by

$$LazyEnt(D, A_j, v_j) = min(Ent(D, A_j, v_j), Ent(D, A_j)), \tag{4}$$

where $min()$ returns the smallest of its arguments.

After calculating the value $LazyEnt(D, A_j, v_j)$ for each attribute $A_j$, the lazy strategy selects the $r$ attributes which present the $r$ lowest $LazyEnt$ values.

Table II. Accuracies for the Hypo-Thyroid dataset

| HYPO-THYROID | 1-NN | | 3-NN | | 5-NN | |
|---|---|---|---|---|---|---|
| Attributes Selected | **eager** | **lazy** | **eager** | **lazy** | **eager** | **lazy** |
| 10% (3) | 91.4 | 97.0 | 91.3 | 97.5 | 91.2 | 97.3 |
| 20% (6) | 94.6 | 96.1 | 94.8 | 95.4 | 94.6 | 95.1 |
| 30% (9) | 92.6 | 95.3 | 93.8 | 94.9 | 94.0 | 94.6 |
| 40% (12) | 92.4 | 94.8 | 93.7 | 94.6 | 93.8 | 94.4 |
| 50% (15) | 92.6 | 94.6 | 93.6 | 94.5 | 93.6 | 94.4 |
| 60% (17) | 92.5 | 94.6 | 93.7 | 94.6 | 93.7 | 94.4 |
| 70% (20) | 92.6 | 94.5 | 93.6 | 94.6 | 93.8 | 94.3 |
| 80% (23) | 92.7 | 94.0 | 93.6 | 94.1 | 93.7 | 94.1 |
| 90% (26) | 92.4 | 92.0 | 93.5 | 93.3 | 93.6 | 93.3 |
| 100% (29) | 91.5 | | 93.2 | | 93.3 | |

Table II shows the accuracies obtained by the $k$-NN algorithm, with the number of neighbors ($k$) parameter set to $k = 1$ (column 2), $k = 3$ (column 3), and $k = 5$ (column 4), after applying both eager and lazy attribute selection techniques based on the entropy measure. This table shows the results for just one example dataset. Results for many more datasets were reported in [Pereira et al. 2011], but this example dataset is enough for the purposes of our discussion here.

In the table, each row shows the results for a fixed percentage of attributes to be selected. We adopted percentages varying from 10% to 100%. The lazy approach was compared with the traditional eager selection strategy, considering that both strategies would select the same number of attributes. Each value of predictive accuracy was obtained by a 10-fold cross-validation procedure [Han and Kamber 2006].

In this particular dataset, the lazy strategy achieved a better result than the eager strategy. The best predictive accuracy was achieved with 10% of the attributes selected when using the lazy strategy, regardless of the $k$ parameter of the $k$-NN algorithm. With the eager strategy, the best percentage of attributes selected was 20%.

This given example shows a drawback of the original work ([Pereira et al. 2011]) on lazy attribute selection: for each dataset, the parameter value which represents the appropriate number of attributes to be selected is unknown a priori. In this work, we attempt to solve this problem by proposing and evaluating two different approaches: 1) estimating the best value of this parameter using a wrapper-based procedure and 2) combining the predictions obtained with several different values of this parameter using a voting-based procedure.

## 3. WRAPPER-BASED APPROACH

As stated before, the proper number of attributes to be selected by the previously proposed lazy attribute selection technique, that is, the value that produces the best classification performance, is unknown beforehand. Our first approach to overcome this issue is to employ the target classification method itself and the training dataset, in a wrapper-based approach, to estimate the best number of attributes.

The wrapper-based approach adopted in this work is based on a procedure used in conjunction with traditional eager attribute selection techniques that follow the filter strategy, as in [Ng 1998] and [Zhu et al. 2007]. The attributes are ranked by the filter using a measure such as entropy, and how many are selected may be determined using the classification technique in a wrapper-based procedure. This is not to be confused with a pure wrapper-based attribute selection procedure where the classification algorithm is used to evaluate the quality of a subset of attributes for all possible subsets. In particular, the approach of using the wrapper idea to select the number of attributes ranked by a filter is computationally less expensive than the pure wrapper approach because the classification algorithm is executed only for a few pre-selected attribute subsets [Guyon et al. 2006], since the ranking is considered.

The wrapper-based implementation proposed here works as follows. Let $D(A_1, A_2, ..., A_n, C)$, $n \geq 1$, be the training dataset with $n + 1$ attributes, where $C$ is the class attribute. And let $r$, $r \leq n$, be the percentage number of attributes to be selected by the lazy strategy described in Section 2. The following process of estimating an adequate value of $r$ is executed as a preprocessing phase, before the submission of any instance to be classified using the dataset $D$ and a given classification algorithm.

For each value of $r$, i.e., the percentage of attributes to be selected, varying from 10% to 100% with a regular increment of 10%, each instance of $D$ is classified with the classifier built from all the remaining training instances – using a leave-one-out procedure [Han and Kamber 2006]. For each of these values of $r$, the classification accuracy of the leave-one-out procedure is obtained. Then, the percentage $r$ that gives the best classification accuracy is chosen to be used by the lazy attribute selection technique at classification time. In case of ties, the smallest percentage value is chosen.

Then, any test instance $I$, submitted for classification considering the dataset $D$, will be classified after applying the lazy attribute selection strategy with the parameter of $r$% attributes, as described in Section 2.

Experiments with the $k$-NN algorithm

As in the original work ([Pereira et al. 2011]), the experiments are initially performed using the $k$-NN algorithm, available within the Weka tool (version 3.4.11) with the name of IBk [Witten and Frank 2005]. The $k$-NN algorithm assigns to a new instance to be classified the majority class among its $k$ closest instances, from a given training dataset [Cover and Hart 1967; Dasarathy 1991].

The same datasets from the UCI Machine Learning Repository [Asuncion and Newman 2007] adopted in the experiments conducted in [Pereira et al. 2011] were used to assess the techniques proposed in this work. A total of 40 datasets, which have a wide variation in size, complexity and application area, were chosen.

Their continuous attributes were also discretized by the same entropy minimization heuristic [Fayyad and Irani 1993] coupled with a minimum description length criterion [Rissanen 1986]. This was necessary since the entropy measure used to evaluate the quality of each attribute requires discrete attribute values.

Table III shows the accuracy obtained for each dataset with the $k$-NN algorithm, with $k$ equals to 1, after using our wrapper-based approach combined with the lazy attribute selection technique proposed in [Pereira et al. 2011]. In each row, there is the following information for each dataset: in the first column the dataset name, and their number of attributes, classes and instances. In the next columns it is presented the minimum and maximum predictive accuracy obtained in the original work ([Pereira et al. 2011]) considering the ten executions (i.e., with percentages of selected attributes varying from 10% to 100% with a regular increment of 10%). Note that, in practice, this maximum accuracy is not attainable, since this would require knowing the best percentage of attributes to be selected for the test instances. The column "no sel" indicates the accuracy obtained with no attribute selection, equivalent to 100% of attributes selected. The Wrapper Lazy column shows the predictive accuracy obtained after applying the wrapper-based technique in combination with the lazy attribute selection as described in this section, and in parenthesis the percentual distance to the maximum accuracy. The purpose of showing this percentual distance is to make explicit how far the wrapper procedure is from the optimum which would be attained if we knew the best percentage for each test instance. In the last column we show the average percentage of attributes selected by the wrapper-based technique. Note that for each cross-validation fold the technique can choose a different number of attributes, so this explains why the need to show an average percentage of attributes selected for each dataset. Bold-faced values indicate if the higher predictive accuracy was obtained by the wrapper-based approach or without attribute selection. Each value of predictive accuracy was obtained by a 10-fold cross-validation procedure [Han and Kamber 2006]. In addition, we employed a paired two-

Table III. Wrapper-based results with UCI datasets and the 1-NN algorithm

| **Dataset** | **Results from [Pereira et al. 2011]** | | **Wrapper Lazy** | | |
|---|---|---|---|---|---|
| (attributes, classes, instances) | minimum | maximum | no sel | accuracy | avg atrs selected |
| anneal (38, 5, 898) | 98.0 | 99.6 | **99.2** | 99.0 (0.6%) | 56% |
| audiology (69, 24, 226) | 72.6 | 77.0 | **76.1** | 75.2 (1.8%) | 60% |
| autos (25, 6, 205) | 78.0 | 87.3 | **85.9** | 85.4 (1.9%) | 69% |
| breast-cancer (9, 2, 286) | 69.9 | 74.1 | 69.9 | **71.0** (3.1%) | 65% |
| breast-w (9, 2, 699) | 96.1 | 97.1 | **97.1** | 96.6 (0.5%) | 49% |
| chess-Kr-vs-Kp (36, 2, 3196) | 93.5 | 96.8 | 96.6 | **96.8** (0.0%) | 49% |
| credit-a (15, 2, 690) | 81.9 | 85.5 | 82.3 | **84.5** (1.0%) | 23% |
| diabetes (8, 2, 768) | 72.8 | 78.0 | 76.4 | **77.2** (0.8%) | 41% |
| flags (29, 8, 194) | 52.6 | 60.8 | **59.8** | 58.2 (2.6%) | 50% |
| glass (9, 6, 214) | 66.4 | 77.1 | **77.1** | 76.2 (0.9%) | 78% |
| heart-cleveland (13, 2, 303) | 75.9 | 82.8 | 80.5 | **81.2** (1.6%) | 41% |
| heart-hungarian (13, 2, 294) | 78.6 | 81.6 | **80.3** | 79.9 (1.7%) | 42% |
| hepatitis (19, 2, 155) | 81.3 | 86.5 | 83.9 | 83.9 (2.6%) | 62% |
| horse-colic (27, 2, 368) | 76.1 | 83.7 | 78.5 | **80.7** (3.0%) | 35% |
| hypothyroid (29, 4, 3772) | 91.5 | 97.0 | 91.5 | **96.8** (0.2%) | 13% |
| ionosphere (34, 2, 351) | 92.6 | 94.6 | **92.6** | 92.3 (2.3%) | 52% |
| labor (16, 2, 57) | 93.0 | 100.0 | 96.5 | 96.5 (3.5%) | 40% |
| letter-recogn (16, 26, 20000) | 50.5 | 91.9 | **91.9** | 91.7 (0.2%) | 70% |
| lymph (18, 4, 148) | 72.3 | 85.1 | 82.4 | 82.4 (2.7%) | 74% |
| mol-bio-promoters (57, 2, 106) | 77.4 | 89.6 | 80.2 | **86.8** (2.8%) | 16% |
| mol-bio-splice (60, 3, 3190) | 73.3 | 90.7 | 73.3 | **90.7** (0.0%) | 10% |
| mushroom (22, 2, 8124) | 99.8 | 100.0 | 100.0 | 100.0 (0.0%) | 28% |
| optdigits (64, 10, 5620) | 82.0 | 94.8 | 94.3 | **94.7** (0.1%) | 65% |
| pendigits (16, 10, 10992) | 63.1 | 97.1 | **97.1** | 97 (0.1%) | 97% |
| postoperative (8, 3, 90) | 60.0 | 71.1 | 63.3 | **68.9** (2.2%) | 38% |
| primary-tumor (17, 21, 339) | 38.3 | 42.5 | 38.3 | **41.0** (1.5%) | 52% |
| solar-flare1 (12, 6, 323) | 60.4 | 71.5 | 65.9 | **68.4** (3.1%) | 44% |
| solar-flare2 (12, 6, 1066) | 71.8 | 76.3 | 73.5 | **74.6** (1.7%) | 32% |
| sonar (60, 2, 208) | 74.0 | 86.5 | **86.5** | 80.8 (5.7%) | 79% |
| soybean-large (35, 19, 683) | 88.0 | 93.4 | **92.2** | 91.9 (1.5%) | 77% |
| spambase (57, 2, 4601) | 92.7 | 93.7 | 93.0 | **93.4** (0.3%) | 66% |
| statlog-heart (13, 2, 270) | 71.5 | 85.2 | **84.1** | 82.2 (3.0%) | 57% |
| statlog-segment (19, 7, 2310) | 80.3 | 94.7 | **94.7** | 94.2 (0.5%) | 73% |
| statlog-vehicle (18, 4, 846) | 60.3 | 71.4 | **70.9** | 70.6 (0.8%) | 78% |
| thyroid-sick (29, 2, 3772) | 97.1 | 97.5 | **97.5** | 97.4 (0.1%) | 68% |
| vote (16, 2, 435) | 92.2 | 96.1 | 92.2 | **95.2** (0.9%) | 32% |
| vowel (13, 11, 990) | 30.5 | 89.8 | 89.8 | **90.0** (-0.2%) | 93% |
| waveform-5000 (40, 3, 5000) | 73.8 | 75.5 | 73.8 | **73.9** (1.6%) | 51% |
| wine (13, 3, 178) | 95.5 | 98.9 | **98.3** | 97.8 (1.1%) | 47% |
| zoo (17, 7, 101) | 82.2 | 97.0 | 96.0 | 96.0 (1.0%) | 64% |
| **Num. of wins** | | | 17 | **18** | |
| **(Num. of wins with stat. t-test)** | | | (1) | **(5)** | |

tailed Student's t-test (with a significance level of 0.05) with the goal of identifying which compared predictive accuracies are actually significantly different between the lazy wrapper-based approach and the classification without attribute selection. It is worth saying that experiments with the parameter $k$ equal to 3 and 5 will be reported in the next section.

The experimental results presented in Table III show that the wrapper-based approach combined with the lazy attribute selection technique achieves in five datasets a predictive accuracy better than that obtained without attribute selection, and the converse was true in just one dataset . The wrapper-based approach also deviates, on average, 1.47% from the maximum accuracy value obtained in the original work on lazy attribute selection.

These results indicate that the wrapper-based approach proposed here is a useful strategy to overcome the problem of parameter adjustment for the lazy attribute selection technique that was proposed in [Pereira et al. 2011]. However, the total number of attributes in the UCI datasets used in these experiments varies between 8 and 69, which is a small number for today's standards where most real data mining applications have attribute numbers on the order of hundreds or even thousands.

In order to evaluate if the proposed method is scalable and effective on larger datasets, additional experiments were performed with datasets from the NIPS 2003 challenge on feature selection [Guyon et al. 2004]. This competition took place in conjunction with the NIPS 2003 conference, and made

Table IV. Wrapper-based results with NIPS large datasets for the 1-NN classification algorithm

| **Dataset** | **Results from [Pereira et al. 2011]** | | | **Wrapper Lazy** | |
|---|---|---|---|---|---|
| (attributes, classes, instances) | minimum | maximum | no sel | accuracy | avg atrs selected |
| Arcene (10000, 2, 200) | 81.0 | 92.5 | **91.0** | 90.5 (1.5%) | 79% |
| Madelon (500, 2, 2600) | 61.2 | 73.0 | **73.0** | 72.8 (0.2%) | 95% |
| Gisette (5000, 2, 7000) | 96.4 | 97.1 | 96.8 | **97.2** (-0.1%) | 52% |
| Dexter (20000, 2, 600) | 88.3 | 93.7 | 88.3 | **92.7** (1.0%) | 25% |
| Dorothea (100000, 2, 1150) | 90.3 | 90.6 | **90.6** | 90.5 (0.1%) | 22% |

available five datasets to be used as benchmarks for attribute selection methods.

The original datasets were divided into three sets – training, validation and test – and the class label of each instance was provided only for the training and validation data. Thus, we ignored the test set and merged the training and validation sets into a single dataset. Then, we employed the same 10-fold cross-validation procedure adopted with the UCI datasets to assess the predictive accuracy on hold-out data.

The same procedure described earlier to discretize continuous attributes was adopted for these datasets, except for the Dorothea dataset, which has only binary attributes (0/1). Some irrelevant and random attributes, referred to as "probes" in [Guyon et al. 2004], are present in these datasets. In many cases, these attributes were discretized into a single bin by the discretization procedure. When this happened, we removed the attribute from the dataset.

Table IV summarizes the experimental results with these large datasets using the 1-NN algorithm. Again, we show the minimum and maximum obtained in the original work presented in [Pereira et al. 2011] considering the ten percentages of attributes selected. The Wrapper Lazy column shows the results of using the wrapper-based appproach combined with the lazy attribute selection technique and the percentual distance to the maximum value.

Here, the wrapper-based approach combined with lazy attribute selection was only able to surpass not doing any attribute selection in two out of the five datasets. However, in most cases it got very close to the maximum and in one case (Gisette dataset) it even surpassed the maximum. This is possible because of the cross-validation procedure which we use to obtain the accuracy, which allows the wrapper to select a different percentage value for each fold.

## 4. VOTING-BASED APPROACH

The second approach that we propose in this work to eliminate the need of choosing the number of attributes to be used by the lazy attribute selection technique is based on a voting mechanism. Voting ensemble methods rely on the hypothesis that a combination of differently trained classifiers can improve the prediction accuracy of the final decision, when compared to using a single classifier [Han and Kamber 2006].

The traditional voting method for combining classifiers works as follows: each classifier provides its final decision as a class label, and then the class label with the higher number of votes is selected as the final output of the system. This is useful especially when the performance of any individual classifier is not satisfactory due to overfitting or insufficient number of training instances in comparison with the number of attributes.

In this work, we have implemented a voting-based procedure to combine different classifiers, each one obtained by applying the lazy attribute selection with a different number of attributes selected. Thus, the voting ensemble method avoids the need to choose a particular number of attributes to be selected, and we can also expect a better overall predictive performance, as the decisions of several classifiers are combined.

The voting-based strategy proposed here works as follows. Let $D(A_1, A_2, ..., A_n, C)$, $n \geq 1$, be the

training dataset with $n+1$ attributes, where $C$ is the class attribute. Let $I$ be the test instance to be classified by a given classification algorithm. And let $r$, $r \leq n$, be the percentage number of attributes of $I$ to be selected by the lazy strategy described in Section 2. Differently from the wrapper-based estimation, the following voting process is not executed as a preprocessing phase. It is executed when an instance is submitted for classification.

For each value of $r$, varying from 10% to 100% with a regular increment of 10%, the instance $I$ is classified by the given classification algorithm, using the dataset $D$ and the attributes selected by the lazy strategy, as described in Section 2. Each one of these classification executions will result in a vote, i.e., in a class label. Then the class label with the higher number of votes will be selected as the result class. In case of ties, the result class is randomly selected from majority classes.

### 4.1 Experiments with the k-NN classification algorithm

The experiments were initally conducted for the same 40 datasets from UCI Machine Learning Repository [Asuncion and Newman 2007] previously mentioned. Again, the classifier used was the $k$-NN implemented within the Weka tool (called IBk) with parameter $k$ equal to 1, 3 and 5. Each predictive accuracy was calculated as the average of a 10-fold cross-validation procedure.

Table V shows the predictive accuracy results for each dataset, obtained by the $k$-NN classifiers performing the lazy attribute selection using both wrapper-based and voting-based approaches. For each dataset, the better result for a given classifier is marked in bold. Also, the better results considering the statistical significance test are underlined. The last two rows show the total number of wins for each method – lazy wrapper and voting – when considering or not the statistical significance test.

We observe that, regardless of the parameter $k$, the voting-based lazy method tends to achieve greater accuracies than the wrapper-based lazy method. For the 120 executions (40 datasets $\times$ 3 values of parameter $k$), the voting-based approach attained a better result in 77 cases and the wrapper-based in 31 cases. Ties arose in the remaining executions. Considering the statistical significance test, the voting-based approach obtained 21 times the best result and the wrapper-based just 8 times.

Given the satisfactory results for the voting-based approach applied to lazy attribute selection, a question arises as to what would happen if a similar procedure were applied for eager attribute selection. Aiming at a fair comparison, we applied the voting-based approach to the eager attribute selection technique most similar to our lazy strategy, called Information Gain Attribute Ranking [Yang and Pedersen 1997]. This technique is available within the Weka tool with the same name.

The predictive accuracy results for each dataset with $k$-NN classifiers obtained with the voting-based approach combined with both the eager and lazy attribute selection are shown in Table VI. As before, for each dataset and each $k$ parameter of the $k$-NN algorithm, bold-faced values indicate which method obtained the best result, and the best results achieved considering the statistical significance test are underlined. The last two rows show the total number of wins for each method – voting eager and voting lazy – with and without the statistical significance test.

Considering the 120 executions (40 datasets $\times$ 3 classification methods), the results reveal a major superiority of the voting-based lazy approach, as it achieved the best accuracy 63 times, whereas the voting-based eager approach achieved the best accuracy 37 times. Considering the statistical significance test, the voting-based lazy strategy obtained a better result in 20 cases and the wrapper-based eager in just 4 cases.

We also performed experiments to compare the voting-based lazy approach to the voting-based eager approach using the larger datasets from the NIPS 2003 challenge on feature selection [Guyon et al. 2004]. Table VII summarizes the results for the large datasets with the voting-based procedures implemented in this work (both eager and lazy). These experiments reveal that the voting-based lazy proposal is also competitive when handling larger datasets. This strategy presented a better

Table V. Wrapper-based and voting-based lazy attribute selection results with UCI datasets and $k$-NN algorithm

| Dataset (attributes, classes, instances) | 1-NN | | 3-NN | | 5-NN | |
|---|---|---|---|---|---|---|
| | **Wrapper** | **Voting** | **Wrapper** | **Voting** | **Wrapper** | **Voting** |
| anneal (38, 5, 898) | 99.0 | **99.4** | 98.0 | **98.7** | 96.9 | **98.0** |
| audiology (69, 24, 226) | 75.2 | **77.0** | 66.4 | **72.6** | 70.8 | **71.2** |
| autos (25, 6, 205) | 85.4 | **88.3** | **81.0** | 77.1 | 72.7 | **76.6** |
| breast-cancer (9, 2, 286) | 71.0 | **76.6** | 71.3 | **73.8** | 73.4 | 73.4 |
| breast-w (9, 2, 699) | 96.6 | **96.7** | 96.6 | **97.0** | 96.9 | **97.0** |
| chess-Kr-vs-Kp (36, 2, 3196) | 96.8 | **97.0** | 96.0 | **96.9** | 95.2 | **96.5** |
| credit-a (15, 2, 690) | 84.5 | **85.4** | 84.3 | **86.2** | 83.8 | **86.7** |
| diabetes (8, 2, 768) | 77.2 | **78.3** | 77.2 | **77.9** | 77.2 | **78.0** |
| flags (29, 8, 194) | **58.2** | 56.2 | **60.3** | 59.8 | 60.3 | **62.4** |
| glass (9, 6, 214) | 76.2 | **77.6** | 75.2 | 75.2 | 71.0 | **72.9** |
| heart-cleveland (13, 2, 303) | 81.2 | **81.5** | **82.5** | 81.8 | 81.5 | 81.5 |
| heart-hungarian (13, 2, 294) | **79.9** | 78.9 | 80.3 | **82.3** | 82.3 | **83.0** |
| hepatitis (19, 2, 155) | 83.9 | **85.8** | 82.6 | **83.2** | 83.2 | **83.9** |
| horse-colic (27, 2, 368) | **80.7** | 80.4 | 80.4 | **82.6** | 79.9 | **82.6** |
| hypothyroid (29, 4, 3772) | **96.8** | 94.9 | **97.6** | 95.2 | **97.3** | 94.7 |
| ionosphere (34, 2, 351) | 92.3 | **93.7** | 90.3 | **91.2** | 90.0 | **90.9** |
| labor (16, 2, 57) | 96.5 | **100.0** | **94.7** | 93.0 | 91.2 | 91.2 |
| letter-recogn (16, 26, 20000) | 91.7 | **92.9** | 90.3 | **91.6** | 89.8 | **90.4** |
| lymph (18, 4, 148) | 82.4 | 82.4 | **82.4** | 80.4 | **81.8** | 79.7 |
| mol-bio-promoters (57, 2, 106) | 86.8 | **89.6** | 85.8 | **86.8** | 85.8 | **87.7** |
| mol-bio-splice (60, 3, 3190) | **90.7** | 90.5 | 91.2 | **93.2** | 90.7 | **93.5** |
| mushroom (22, 2, 8124) | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| optdigits (64, 10, 5620) | 94.7 | **95.3** | 95.4 | **95.6** | **95.4** | 95.4 |
| pendigits (16, 10, 10992) | **97.0** | 96.8 | **96.6** | 96.2 | **96.3** | 95.7 |
| postoperative (8, 3, 90) | **68.9** | 66.7 | 70.0 | **71.1** | 70.0 | **71.1** |
| primary-tumor (17, 21, 339) | 41.0 | **42.5** | 43.1 | **44.8** | 45.7 | **46.3** |
| solar-flare1 (12, 6, 323) | 68.4 | **70.6** | 69.0 | **71.2** | 69.0 | **70.9** |
| solar-flare2 (12, 6, 1066) | **74.6** | 73.6 | **74.9** | 74.6 | 73.9 | **74.9** |
| sonar (60, 2, 208) | 80.8 | **82.2** | 76.0 | **78.8** | 74.5 | **75.0** |
| soybean-large (35, 19, 683) | 91.9 | **93.7** | 90.9 | **93.0** | 90.8 | **92.7** |
| spambase (57, 2, 4601) | 93.4 | **94.1** | 93.6 | **94.0** | 93.1 | **93.5** |
| statlog-heart (13, 2, 270) | 82.2 | **85.6** | **81.9** | 81.5 | **83.0** | 82.6 |
| statlog-segment (19, 7, 2310) | 94.2 | **95.3** | 93.3 | **93.9** | **92.8** | 92.6 |
| statlog-vehicle (18, 4, 846) | **70.6** | 69.7 | **72.1** | 69.4 | **71.3** | 68.8 |
| thyroid-sick (29, 2, 3772) | 97.4 | **97.6** | 97.1 | **97.4** | 97.3 | **97.4** |
| vote (16, 2, 435) | **95.2** | 94.3 | **95.2** | 94.9 | **94.9** | 94.7 |
| vowel (13, 11, 990) | **90.0** | 87.7 | **84.4** | 83.1 | 77.3 | 77.3 |
| waveform-5000 (40, 3, 5000) | 73.9 | **75.7** | 78.6 | **80.5** | 79.7 | **81.6** |
| wine (13, 3, 178) | 97.8 | **98.9** | 97.8 | **98.9** | 98.9 | 98.9 |
| zoo (17, 7, 101) | 96.0 | 96.0 | 90.1 | **94.1** | 89.1 | 89.1 |
| **Num. of wins** | **11** | **26** | **12** | **26** | **8** | **25** |
| **(Num. of wins with stat. t-test)** | **(2)** | **(7)** | **(3)** | **(8)** | **(3)** | **(6)** |

behaviour both with and without the statistical analysis. The lazy approach was significantly better than the eager approach in 4 cases and the eager version was not able to present a significantly better performance.

## 4.2 Experiments with the Naive Bayes algorithm

The experiments presented thus far adopted the $k$-NN classifier to evaluate the classification performance of the proposed wrapper-based and voting-based approaches. But, in principle, lazy attribute selection is a general strategy that can be applied to any lazy classification method or to any lazy version of an eager classification method (e.g.: lazy decision trees).

To show that the satisfactory results of the voting-based approach are valid not only for $k$-NN, in this section we are going to present the results of combining the voting-based approach with the Naive Bayes classification method.

Naive Bayes is a classification technique based on the Bayes theorem [Duda et al. 2001]. The classifier applies this theorem assuming that the attributes contribute in an independent manner to the likelihood of the value of the class – and although this premise is not always accurate, it usually yields good results in practice.

Table VI. Voting-based eager and voting-based lazy results with UCI datasets and the $k$-NN algorithm

| Dataset | 1-NN | | 3-NN | | 5-NN | |
|---|---|---|---|---|---|---|
| (attributes, classes, instances) | **Eager** | **Lazy** | **Eager** | **Lazy** | **Eager** | **Lazy** |
| anneal (38, 5, 898) | 99.3 | **99.4** | 98.1 | **98.7** | 97.7 | **98.0** |
| audiology (69, 24, 226) | 77.0 | 77.0 | 70.4 | **72.6** | **71.7** | 71.2 |
| autos (25, 6, 205) | 88.3 | 88.3 | **77.6** | 77.1 | 76.6 | 76.6 |
| breast-cancer (9, 2, 286) | 73.4 | **76.6** | 73.1 | **73.8** | **74.5** | 73.4 |
| breast-w (9, 2, 699) | 96.3 | **96.7** | 96.7 | **97.0** | **97.1** | 97.0 |
| chess-Kr-vs-Kp (36, 2, 3196) | 96.6 | **97.0** | 96.4 | **96.9** | 96.0 | **96.5** |
| credit-a (15, 2, 690) | 85.2 | **85.4** | **86.4** | 86.2 | 85.9 | **86.7** |
| diabetes (8, 2, 768) | **79.0** | 78.3 | **79.2** | 77.9 | **79.2** | 78.0 |
| flags (29, 8, 194) | 56.2 | 56.2 | **61.3** | 59.8 | 61.9 | **62.4** |
| glass (9, 6, 214) | 75.2 | **77.6** | 72.9 | **75.2** | 72.9 | 72.9 |
| heart-cleveland (13, 2, 303) | **82.2** | 81.5 | **82.8** | 81.8 | **82.2** | 81.5 |
| heart-hungarian (13, 2, 294) | **80.3** | 78.9 | **82.7** | 82.3 | 82.7 | **83.0** |
| hepatitis (19, 2, 155) | 83.9 | **85.8** | **86.5** | 83.2 | 83.9 | 83.9 |
| horse-colic (27, 2, 368) | 79.9 | **80.4** | **83.4** | 82.6 | **82.9** | 82.6 |
| hypothyroid (29, 4, 3772) | 93.2 | **94.9** | 94.4 | **95.2** | 94.0 | **94.7** |
| ionosphere (34, 2, 351) | 93.7 | 93.7 | 91.2 | 91.2 | 90.6 | **90.9** |
| labor (16, 2, 57) | 96.5 | **100.0** | **94.7** | 93.0 | 91.2 | 91.2 |
| letter-recogn (16, 26, 20000) | 92.5 | **92.9** | 91.2 | **91.6** | 89.9 | **90.4** |
| lymph (18, 4, 148) | **85.1** | 82.4 | **81.1** | 80.4 | 79.1 | **79.7** |
| mol-bio-promoters (57, 2, 106) | 86.8 | **89.6** | **87.7** | 86.8 | **88.7** | 87.7 |
| mol-bio-splice (60, 3, 3190) | 86.6 | **90.5** | 88.9 | **93.2** | 89.0 | **93.5** |
| mushroom (22, 2, 8124) | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| optdigits (64, 10, 5620) | 95.2 | **95.3** | **96.0** | 95.6 | **95.6** | 95.4 |
| pendigits (16, 10, 10992) | 95.5 | **96.8** | 94.6 | **96.2** | 93.8 | **95.7** |
| postoperative (8, 3, 90) | 62.2 | **66.7** | 71.1 | 71.1 | 71.1 | 71.1 |
| primary-tumor (17, 21, 339) | 42.5 | 42.5 | **46.3** | 44.8 | 46.0 | **46.3** |
| solar-flare1 (12, 6, 323) | **70.9** | 70.6 | 70.6 | **71.2** | 70.9 | 70.9 |
| solar-flare2 (12, 6, 1066) | **74.1** | 73.6 | 74.6 | 74.6 | **75.0** | 74.9 |
| sonar (60, 2, 208) | 81.7 | **82.2** | 77.4 | **78.8** | **75.5** | 75.0 |
| soybean-large (35, 19, 683) | 93.4 | **93.7** | 91.9 | **93.0** | 90.6 | **92.7** |
| spambase (57, 2, 4601) | 93.6 | **94.1** | 93.8 | **94.0** | 93.3 | **93.5** |
| statlog-heart (13, 2, 270) | 84.4 | **85.6** | **82.6** | 81.5 | **83.3** | 82.6 |
| statlog-segment (19, 7, 2310) | 94.7 | **95.3** | 93.9 | **93.9** | 92.1 | **92.6** |
| statlog-vehicle (18, 4, 846) | 69.7 | 69.7 | 68.6 | **69.4** | 68.1 | **68.8** |
| thyroid-sick (29, 2, 3772) | **97.9** | 97.6 | **97.7** | 97.4 | **97.6** | 97.4 |
| vote (16, 2, 435) | **94.7** | 94.3 | 94.7 | **94.9** | 94.3 | **94.7** |
| vowel (13, 11, 990) | 87.4 | **87.7** | 83.0 | **83.1** | **77.9** | 77.3 |
| waveform-5000 (40, 3, 5000) | 75.4 | **75.7** | 80.0 | **80.5** | 81.4 | **81.6** |
| wine (13, 3, 178) | 98.3 | **98.9** | 97.2 | **98.9** | 97.2 | **98.9** |
| zoo (17, 7, 101) | 96.0 | 96.0 | 94.1 | 94.1 | **90.1** | 89.1 |
| **Num. of wins** | **8** | **24** | **15** | **20** | **14** | **19** |
| **(Num. of wins with stat. t-test)** | **(1)** | **(7)** | **(2)** | **(7)** | **(1)** | **(6)** |

Table VII. Voting-based eager and voting-based lazy results with NIPS datasets and the $k$-NN algorithm

| Dataset | 1-NN | | 3-NN | | 5-NN | |
|---|---|---|---|---|---|---|
| (attributes, classes, instances) | **Eager** | **Lazy** | **Eager** | **Lazy** | **Eager** | **Lazy** |
| Arcene (10000, 2, 200) | **90.5** | 89.0 | 87.5 | **88.0** | 86.5 | 86.5 |
| Madelon (500, 2, 2600) | **68.8** | 68.4 | **69.0** | 68.6 | **68.9** | 68.5 |
| Gisette (5000, 2, 7000) | 96.8 | **97.3** | 96.8 | **97.4** | 96.6 | **96.9** |
| Dexter (20000, 2, 600) | 91.7 | **91.8** | 93.0 | **94.0** | 92.2 | **94.0** |
| Dorothea (100000, 2, 1150) | 88.4 | **90.3** | 89.0 | **90.3** | **90.6** | 90.3 |

The Naive Bayes algorithm can be used as an eager or lazy technique. If all conditional and a priori probabilities are previously calculated, before any instance is submitted for classification, it can be seen as an eager strategy. However, if we decide to compute the necessary probabilities for a particular instance only at classification time, it can be considered a lazy technique.

The voting-based procedure reported earlier was implemented for both the eager and lazy attribute selection techniques, and combined with the Naive Bayes classification method.

The predictive accuracy results for each dataset with the voting-based procedure obtained after the eager and lazy attribute selection and combined with the Naive Bayes are showed in Table VIII. As before, for each dataset, bold-faced values indicate which strategy – eager or lazy – obtained the best result, and the better results achieved considering the statistical significance test are underlined. The

Table VIII. Voting-based eager and voting-based lazy results with UCI datasets and Naive Bayes algorithm

| Dataset (attributes, classes, instances) | NaiveBayes Eager | NaiveBayes Lazy |
|---|---|---|
| anneal (38, 5, 898) | **95.0** | 94.9 |
| audiology (69, 24, 226) | 74.8 | **76.1** |
| autos (25, 6, 205) | **74.1** | 72.7 |
| breast-cancer (9, 2, 286) | **73.4** | 71.0 |
| breast-w (9, 2, 699) | **97.3** | 96.9 |
| chess-Kr-vs-Kp (36, 2, 3196) | **89.5** | 88.5 |
| credit-a (15, 2, 690) | 86.2 | **86.5** |
| diabetes (8, 2, 768) | 79.0 | 79.0 |
| flags (29, 8, 194) | **61.3** | 60.8 |
| glass (9, 6, 214) | 74.8 | 74.8 |
| heart-cleveland (13, 2, 303) | **84.2** | 83.8 |
| heart-hungarian (13, 2, 294) | **84.4** | 83.0 |
| hepatitis (19, 2, 155) | 85.8 | **86.5** |
| horse-colic (27, 2, 368) | **84.8** | 84.5 |
| hypothyroid (29, 4, 3772) | **95.3** | 94.8 |
| ionosphere (34, 2, 351) | 90.9 | **91.2** |
| labor (16, 2, 57) | 94.7 | **96.5** |
| letter-recogn (16, 26, 20000) | 74.0 | **74.6** |
| lymph (18, 4, 148) | 84.5 | 84.5 |
| mol-bio-promoters (57, 2, 106) | **93.4** | 92.5 |
| mol-bio-splice (60, 3, 3190) | **96.2** | 95.5 |
| mushroom (22, 2, 8124) | 96.0 | **96.0** |
| optdigits (64, 10, 5620) | 92.2 | **92.3** |
| pendigits (16, 10, 10992) | 84.5 | **87.0** |
| postoperative (8, 3, 90) | 70.0 | **71.1** |
| primary-tumor (17, 21, 339) | **48.1** | 47.5 |
| solar-flare1 (12, 6, 323) | **70.9** | 69.7 |
| solar-flare2 (12, 6, 1066) | **74.5** | 74.4 |
| sonar (60, 2, 208) | 71.2 | **71.6** |
| soybean-large (35, 19, 683) | 89.2 | **90.2** |
| spambase (57, 2, 4601) | 90.3 | **90.8** |
| statlog-heart (13, 2, 270) | **84.4** | 84.1 |
| statlog-segment (19, 7, 2310) | 89.6 | **90.2** |
| statlog-vehicle (18, 4, 846) | 62.2 | **62.4** |
| thyroid-sick (29, 2, 3772) | 97.1 | **97.2** |
| vote (16, 2, 435) | **91.3** | 89.9 |
| vowel (13, 11, 990) | **56.5** | 56.1 |
| waveform-5000 (40, 3, 5000) | 80.1 | **80.5** |
| wine (13, 3, 178) | 98.3 | **99.4** |
| zoo (17, 7, 101) | 93.1 | **95.0** |
| **Num. of wins** | **18** | **19** |
| **(Num. of wins with stat. t-test)** | **(3)** | **(5)** |

last two rows show the total number of wins for each method – eager voting and lazy voting – with and without the statistical significance test.

The results show that, for Naive Bayes, the voting-based procedure combined with lazy attribute selection is slightly better than the same procedure combined with eager attribute selection, since the former approach obtained significantly better results than the latter in five cases and the converse was true in only three cases.

The same experiments were conducted for the NIPS datasets, and the results are showed in Table IX. In this case, we can see that the voting-based procedure combined with the lazy attribute selection achieved higher predictive results than those obtained by the same procedure combined with eager attribute selection. This suggests that larger datasets may benefit more from the lazy attribute selection approach, given that there are more attributes to choose from.

### 4.3 Adjusting the number of ensemble classifiers

The experiments so far have shown that the voting-based technique implemented in this work gets the best results in terms of accuracy when combined with the lazy attribute selection strategy. So, we intend to evaluate the behavior of this technique when we vary the number of ensemble classifiers.

Table IX. Voting-based eager and voting-based lazy results with NIPS datasets and the Naive Bayes algorithm

| **Dataset** (attributes, classes, instances) | **Naive Bayes** **eager** | **lazy** |
|---|---|---|
| Arcene (10000, 2, 200) | 67.0 | **70.0** |
| Madelon (500, 2, 2600) | 63.4 | **64.1** |
| Gisette (5000, 2, 7000) | 89.8 | **90.1** |
| Dexter (20000, 2, 600) | 93.3 | **94.5** |
| Dorothea (100000, 2, 1150) | 90.4 | **90.5** |

Table X. Accuracies obtained with the 1-NN classifier and the voting-based technique, varying the number of ensemble classifiers

| Dataset | **1-NN Lazy 5** | **1-NN Lazy 10** | **1-NN Lazy 20** | **1-NN Lazy 50** |
|---|---|---|---|---|
| anneal | 99.4 | 99.4 | **99.6** | **99.6** |
| audiology | 76.5 | 77.0 | 76.5 | **77.4** |
| autos | 86.8 | **88.3** | 87.8 | 87.8 |
| breast-cancer | 74.8 | 76.6 | 76.9 | **78.0** |
| breast-w | **97.0** | 96.7 | 96.9 | 96.7 |
| chess-kr-vs-kp | **97.2** | 97.0 | 97.0 | 97.0 |
| credit-a | 84.9 | **85.4** | 85.1 | 85.1 |
| diabetes | **78.8** | 78.3 | 78.1 | 78.4 |
| flags | 55.7 | 56.2 | 54.6 | **56.7** |
| glass | 76.6 | **77.6** | **77.6** | **77.6** |
| heart-cleveland | **82.5** | 81.5 | 81.8 | 81.5 |
| heart-hungarian | **78.9** | **78.9** | 78.6 | 78.6 |
| hepatitis | 84.5 | **85.8** | 85.2 | 85.2 |
| horse-colic | 80.2 | **80.4** | 80.2 | 80.2 |
| hypothyroid | **94.9** | **94.9** | **94.9** | **94.9** |
| ionosphere | **94.0** | 93.7 | **94.0** | **94.0** |
| labor | 98.2 | **100.0** | **100.0** | **100.0** |
| letter-recognition | 92.7 | 92.9 | 92.9 | **93.0** |
| lymph | **83.1** | 82.4 | **83.1** | **83.1** |
| mol-bio-promoters | 85.8 | 89.6 | 89.6 | **90.6** |
| mol-bio-splice | 88.9 | 90.5 | 91.2 | **91.3** |
| mushroom | **100.0** | **100.0** | **100.0** | **100.0** |
| optdigits | 95.2 | 95.3 | **95.3** | 95.3 |
| pendigits | **96.9** | 96.8 | 96.6 | 96.6 |
| postoperative | **66.7** | **66.7** | **66.7** | **66.7** |
| primary-tumor | **42.5** | **42.5** | 41.9 | 41.9 |
| solar-flare1 | 70.0 | **70.6** | **70.6** | **70.6** |
| solar-flare2 | **73.9** | 73.6 | **73.9** | 73.6 |
| sonar | 83.2 | 82.2 | **83.7** | 83.2 |
| soybean-large | 93.9 | 93.7 | **94.3** | **94.3** |
| spambase | 94.0 | 94.1 | 94.2 | **94.3** |
| statlog-heart | **85.6** | **85.6** | 85.2 | **85.6** |
| statlog-segment | 95.1 | **95.3** | **95.3** | 95.2 |
| statlog-vehicle | **70.8** | 69.7 | 70.3 | 70.6 |
| thyroid-sick | 97.6 | 97.6 | 97.6 | **97.6** |
| vote | 93.8 | **94.3** | **94.3** | 94.0 |
| vowel | 87.6 | 87.7 | 87.8 | **88.6** |
| waveform-5000 | 75.3 | 75.7 | **76.0** | 75.9 |
| wine | **98.9** | **98.9** | **98.9** | **98.9** |
| zoo | **96.0** | **96.0** | **96.0** | **96.0** |
| **Num. Wins** | 17 | 17 | 18 | **22** |

The initial evaluation of the voting-based technique considered a fixed parameter of 10 classifiers, varying the percentage of attributes to be selected from 10% to 100% with a regular increment of 10%.

Additional experiments were carried out with the voting-based technique adjusting the number of classifiers to 5, 10, 20 and 50, as reported in Table X. All increments of the number of attributes to be selected were regular, i.e., the ensemble with 5 classifiers had an incremental variation per classifier of 20%, and the ensemble with 50 classifiers varied with an increment of 2%.

Table X shows the accuracy obtained using the 1-NN classifier, combining the lazy attribute selection with the voting-based technique and each of the assessed number of classifiers. Results in bold indicate when the combination achieved the best accuracy overall. The last row shows the number of times for the 40 UCI datasets in which the combination achieved the highest accuracy.

The results indicate that the increase in the number of classifiers does not imply in a significant difference in the overall result of the voting-based technique, even though there is a tendency to achieve a better result with a higher number of classifiers. Similar results were achieved with the 3-NN and the 5-NN classifiers.

## 5. COMPUTATIONAL TIME ANALYSIS

Both techniques proposed in this work incur in an extra computational time in order to avoid the manual selection of the number of attributes for the original lazy attribute selection. In this section we conduct a computational time analysis on these techniques, with the purpose of establishing if they are scalable.

For the wrapper-based approach, most of the work is done at the preprocessing step that runs a leave-one-out procedure to estimate the accuracy for all percentage values on the training data and choose the best percentage value. The actual time taken in this preprocessing step depends not only on the percentage intervals chosen, but also on the number of instances in the dataset. At test time, the number of attributes to be selected is already known and fixed, so there is no extra work and a regular lazy attribute selection step is executed.

On the other hand, for the voting-based approach, no preprocessing work is necessary. However, at test time, we need to obtain and combine $t$ classifiers (where $t$ depends on the percentage intervals chosen), which takes roughly $t$ times longer to execute than a single classification.

Table XI indicates, for each dataset, the computational time that each procedure – wrapper-based lazy (Column 3) and voting-based lazy (Column 5) – has taken overall, i.e. the total time of execution considering the preprocessing and the time to classify each instance, known as test time (in seconds). Also, the average time per instance is given in columns 4 and 6 for the wrapper-based and voting-based approaches, respectively (in milliseconds).

All executions reported were carried out on an Intel Core 2 Duo 4400 2 Ghz, 2 Gb RAM, with the 1-NN classification method of the Weka tool.

As stated before, the total time of execution of the wrapper-based approach is significantly variable (depending on the number of instances in the dataset) and longer than the total time of execution of the voting-based approach, due to its preprocessing time. Nonetheless, the average test times per instance of both techniques are similar, varying from less than one millisecond to 815 milliseconds.

This computational time analysis was also made for the large NIPS datasets employed in the experiments. Both the total time and the average time are, in general, higher than for the UCI datasets, due to the larger number of attributes of these datasets.

The same analysis was conducted for the 3-NN, 5-NN and Naive Bayes classification methods, and the results were similar.

It is worth observing that, in the context of data mining and machine learning tasks, the computational time is considered in general a much less important criterion than the predictive accuracy. This is due to the fact that the classification task is usually performed in an off-line, batch fashion.

## 6. CONCLUSION AND FUTURE WORK

In this article, we have proposed two procedures to avoid the need for manually adjusting the number of selected attributes of a lazy attribute selection technique.

The first procedure was a wrapper-based approach, and the results showed that the lazy selection could benefit from this technique to choose a suitable number of attributes.

Table XI. Computational time of the proposed techniques for the 1-NN classifier

| Data Set | | Wrapper Lazy | | Voting Lazy | |
|---|---|---|---|---|---|
| | | Total time (s) | Avg. test time/inst. (ms) | Total time (s) | Avg. test time/inst. (ms) |
| UCI | anneal | 391.3 | 46.0 | 46.2 | 51.4 |
| | audiology | 40.2 | 14.4 | 4.3 | 19.2 |
| | autos | 12.9 | 4.4 | 1.5 | 7.3 |
| | breast-cancer | 10.0 | 1.8 | 1.0 | 3.7 |
| | breast-w | 59.7 | 5.7 | 5.1 | 7.4 |
| | chess-kr-vs-kp | 4748.0 | 79.6 | 394.9 | 123.6 |
| | credit-a | 87.6 | 7.1 | 7.4 | 10.7 |
| | diabetes | 65.2 | 7.0 | 5.5 | 7.2 |
| | flags | 13.2 | 2.9 | 2.6 | 13.5 |
| | glass | 5.7 | 1.5 | 0.6 | 3.0 |
| | heart-cleveland | 15.1 | 4.1 | 1.5 | 5.0 |
| | heart-hungarian | 14.6 | 1.6 | 1.4 | 4.7 |
| | hepatitis | 5.8 | 2.4 | 0.8 | 5.4 |
| | horse-colic | 44.2 | 12.7 | 5.2 | 14.1 |
| | hypothyroid | 5660.4 | 84.1 | 400.3 | 106.1 |
| | ionosphere | 51.1 | 4.3 | 6.0 | 17.1 |
| | labor | 0.8 | 1.1 | 0.3 | 5.5 |
| | letter-recogn | 85648.1 | 338.7 | 6121.9 | 306.1 |
| | lymph | 5.1 | 2.1 | 0.7 | 4.8 |
| | mol-bio-prom | 7.9 | 4.8 | 2.3 | 21.2 |
| | mol-bio-splice | 7010.1 | 103.2 | 550.0 | 172.4 |
| | mushroom | 18956.1 | 163.7 | 1416.6 | 174.4 |
| | optdigits | 24310.3 | 815.2 | 1841.9 | 327.7 |
| | pendigits | 28263.4 | 199.2 | 1835.8 | 167.0 |
| | postoperative | 1.0 | 0.7 | 0.3 | 2.8 |
| | primary-tumor | 25.2 | 3.4 | 2.2 | 6.5 |
| | solar-flare1 | 16.8 | 4.7 | 1.6 | 4.9 |
| | solar-flare2 | 183.6 | 7.8 | 14.5 | 13.6 |
| | sonar | 30.2 | 1.6 | 3.2 | 15.3 |
| | soybean-large | 214.5 | 5.2 | 18.2 | 26.6 |
| | spambase | 14394.2 | 67.6 | 1102.4 | 239.6 |
| | statlog-heart | 12.2 | 2.6 | 1.2 | 4.6 |
| | statlog-segment | 1277.7 | 47.7 | 103.0 | 44.6 |
| | statlog-vehicle | 162.0 | 4.8 | 13.1 | 15.5 |
| | thyroid-sick | 4981.1 | 11.2 | 394.7 | 104.6 |
| | vote | 39.1 | 7.2 | 3.3 | 7.5 |
| | vowel | 168.2 | 11.9 | 13.4 | 13.5 |
| | waveform-5000 | 11698.2 | 133.8 | 912.7 | 182.5 |
| | wine | 5.3 | 0.9 | 0.7 | 3.9 |
| | zoo | 2.3 | 1.6 | 0.6 | 6.2 |
| NIPS 2003 | Arcene | 2284.0 | 504.2 | 120.6 | 602.9 |
| | Madelon | 1129.8 | 39.7 | 92.7 | 35.7 |
| | Gisette | 127542.2 | 6930.1 | 60075.6 | 8582.2 |
| | Dexter | 908.2 | 10.3 | 6.9 | 11.6 |
| | Dorothea | 689847.3 | 13434.7 | 11418.5 | 9929.1 |

The second approach utilized a voting-based mechanism to combine several lazy attribute selectors within the same classifier, each one with a different number of attributes. Experimental results indicated that this approach was superior to the first one. An analogous eager approach was implemented and compared with the proposed lazy technique, and the results favored the lazy approach.

These results can be considered a significant advance for lazy attribute selection, since this is the first work to overcome the limitation of having to manually choose an appropriate number of attributes to be selected for each dataset.

Although cross-validation, leave-one-out, wrapper and ensemble techniques are well known procedures used by many traditional classification and eager attribute selection methods to tune parameters and combine predictions from multiple parameter settings, we consider that, since eager and lazy learning are two distinct paradigms, we have to revisit these concepts when dealing with lazy attribute selection. Our main contribution is to evaluate these traditional concepts in the lazy selection scenario in order to build lazy selection strategies that are not dependent on a manually chosen parameter.

Even though the entropy is usually a good measure for assessing the relevance of an attribute, it has some drawbacks that could be avoided by employing other ranking measures for attribute

selection, such as the gain ratio, chi-square or gini index measures. Therefore, we plan to conduct experiments with other measures in the near future. We also plan as future work to extend the lazy attribute selection technique to filter strategies that evaluate subsets of attributes instead of evaluating them individually, like Correlation-based Feature Selection [Hall 2000] and Consistency-based Feature Selection [Liu and Setiono 1996].

## REFERENCES


Asuncion, A. and Newman, J. UCI machine learning repository. http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007. University of California, Irvine.

Cover, T. and Hart, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13 (1): 21–27, 1967.

Dasarathy, B. V. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, 1991.

Duch, W. Filter Methods. In I. Guyon, M. Nikravesh, S. Gunn, and L. Zadeh (Eds.), *Studies in Fuzziness and Soft Computing*. Vol. 207/2006. Springer-Verlag, pp. 89–117, 2006.

Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification*. John Wiley & Sons, 2001.

Fayyad, U. M. and Irani, K. B. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*. Chambery, France, pp. 1022–1029, 1993.

Guyon, I. and Elisseeff, A. An Introduction to Feature Extraction. In I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh (Eds.), *Feature Extraction, Foundations and Applications*. Springer, pp. 1–25, 2006.

Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L., editors. *Feature Extraction, Foundations and Applications*. Springer, 2006.

Guyon, I., Hur, A. B., Gunn, S., and Dror, G. Result analysis of the nips 2003 feature selection challenge. In *Advances in Neural Information Processing Systems 17*. pp. 545–552, 2004.

Hall, M. A. A correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the International Conference on Machine Learning*. Stanford, CA, USA, 2000.

Han, J. and Kamber, M. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2006.

Kira, K. and Rendell, L. A practical approach to feature selection. In *Proceedings of the International Conference on Machine Learning*. Aberdeen, Scotland, pp. 249–256, 1992.

Kononenko, I. Estimating attributes: Analysis and extensions of relief. In *Proceedings of the European Conference on Machine Learning*. Catania, Italy, pp. 171–182, 1994.

Liu, H. and Motoda, H. *Computational Methods of Feature Selection*. Chapman & Hall/CRC, 2008a.

Liu, H. and Motoda, H. Less is More. In H. Liu and H. Motoda (Eds.), *Computational Methods of Feature Selection*. Chapman & Hall/CRC, pp. 3–17, 2008b.

Liu, H. and Setiono, R. A probabilistic approach to feature selection: A filter solution. In *Proceedings of the International Conference on Machine Learning*. Bari, Italy, pp. 319–327, 1996.

Liu, H. and Yu, L. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering* 17 (4): 491–502, 2005.

Ng, A. Y. On feature selection: learning with exponentially many irrelevant features as training examples. In *Proceedings of the International Conference on Machine Learning*. Morgan Kaufmann, Madison, WI, USA, pp. 404–412, 1998.

Pereira, R., Plastino, A., Zadrozny, B., Merschmann, L., and Freitas, A. Lazy attribute selection – choosing attributes at classification time. *Intelligent Data Analysis* 15 (5), 2011.

Quinlan, J. R. Induction of decision trees. *Machine Learning* 1 (1): 81–106, 1986.

Rissanen, J. Stochastic complexity and modeling. *Annals of Statistic* 14 (3): 1080–1100, 1986.

Witten, I. H. and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

Yang, Y. and Pedersen, J. O. A comparative study on feature selection in text categorization. In *Proceedings of the International Conference on Machine Learning*. Nashville, TN, USA, pp. 412–420, 1997.

Zhu, Z.-X., Ong, Y.-S., and Dash, M. Wrapper–filter feature selection algorithm using a memetic framework. *IEEE Transactions on Systems Man and Cybernetics, Part B* 37 (1): 70–76, 2007.