Lazy Attribute Selection – Choosing Attributes at Classification Time

Rafael B. Pereira[*], Alexandre Plastino[*], Bianca Zadrozny[*]
Luiz Henrique de C. Merschmann[†], Alex A. Freitas[‡]

July 3, 2010

Abstract

Attribute selection is a data preprocessing step which aims at identifying relevant attributes for the target machine learning task – namely classification in this paper. In this paper, we propose 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. Our strategy 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. Experimental results using the k-NN and Naive Bayes classifiers, over 40 different data sets from the UCI Machine Learning Repository and five large data sets from the NIPS 2003 feature selection challenge, 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. We also propose a metric to estimate when a specific data set can benefit from the lazy attribute selection approach.

Keywords: Attribute selection, classification, lazy learning

[*]Fluminense Federal University - Brazil
[†]Ouro Preto Federal University - Brazil
[‡]University of Kent - U.K.

1 Introduction

Given a training set where each instance is described by a vector of attributes and by a class label, the classification task can be stated as the process of correctly predicting the class label of a new instance or a set of new instances described by their attribute values. One of the main research issues in classification is to design accurate and efficient classification algorithms and models that work for data sets that are large both in terms of the number of instances as well as the number of attributes.

Classification techniques are traditionally categorized as eager or lazy. Eager strategies work on a training set to build an explicit classification model that maps unlabeled instances to class labels. At classification time, they simply use the model to make class predictions. Well-known eager techniques include decision trees (Quinlan, 1986, 1993), neural networks (Ripley, 1996), associative classifiers (Liu et al, 1998) and SVMs (Support Vector Machines) (Burges, 1998).

Lazy strategies, on the other hand, do not construct explicit models and delay most processing of the training set until classification time, when the instance to be classified is known. The most well-known lazy technique is k-NN (k-Nearest Neighbors) (Cover and Hart, 1967; Dasarathy, 1991), in which the class label of an instance is estimated based on the class labels of neighboring instances. The Naive Bayes algorithm (Duda et al, 2001) 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 during classification time, it can be considered a lazy technique. There are also lazy versions of traditional eager techniques such as lazy decision trees (Friedman et al, 1996), lazy rule induction (Gora and Wojna, 2002) and lazy associative classification (Veloso et al, 2006). In these approaches, when an instance is presented for classification only the part of the model needed to classify the particular instance is constructed.

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). These techniques are traditionally executed as a data preprocessing step, making the attribute selection definitive from that point on. The classification process itself is executed over the reduced training set.

In this paper, we propose a new general attribute selection strategy, whose main characteristic is to postpone the selection of relevant attributes – in a lazy fashion – to the moment when an instance is submitted for classification, instead of eagerly selecting the most important attributes. Our hypothesis is that knowing the attribute values of an instance will allow the identification and selection of the best attributes for the correct classification of that specific instance. Therefore, for different instances to be classified, it will be possible to select distinct subsets of attributes, each one customized for that particular instance. We expect that based on the proposed lazy selection technique the adopted classifier will be able to achieve better predictive accuracy than it is possible with an eager attribute selection strategy.

This paper is organized as follows. In Section 2, we describe the attribute selection task in more detail and review related work. In Section 3, we motivate and propose our general lazy attribute selection strategy. We also present a specific instantiation of this strategy that uses an entropy-based criterion to rank attributes. Experimental results using the proposed lazy strategy in conjunction with the k-NN and Naive Bayes classifiers are presented in Section 4. Finally, in Section 5, we make our concluding remarks and point to directions for future work.

2 Attribute Selection

According to (Guyon and Elisseeff, 2006), attribute selection techniques are primarily employed to identify relevant and informative attributes. 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 data set, the acceleration of the classification task, the simplification of the generated classification model, and others. The main motivation of the attribute selection strategy proposed here is improvement in classification accuracy.

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. In general, wrapper techniques achieve higher predictive accuracy than filter strategies since they evaluate candidate attributes subsets using the same algorithm that will be used in the classification (testing) phase. However, since wrapper strategies require several executions of the classification algorithm, their computational costs tend to be much higher than the cost of filter strategies. There are also hybrid strategies which try to combine both approaches (Liu and Yu, 2005).

In this paper we chose to concentrate on developing a lazy attribute selection that follows the filter strategy. The motivation for this choice is twofold. First, among the three types of attribute selection strategies, the filter strategy is the easiest to analyze and evaluate in an isolated manner, since it is independent from the classification algorithm. Secondly, it is much faster than the wrapper approach.

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 detailed in the next section, the lazy attribute selection strategy that we propose in this paper is based on individual evaluation of attributes, using entropy (Quinlan, 1986) to measure the relevance of each attribute.

3 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 data set and no longer participate in the classification process.

Here, we propose 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 1, 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 select attribute X in detriment of Y, regardless of the instances that are submitted for classification. According to (Liu and Motoda, 2008b), a key advantage of lazy approaches in general is that they can respond to unexpected queries in ways not available to eager learners, since they do not lose crucial information that can be used for generating accurate predictions.

Hence, the main motivation behind the proposed lazy attribute selection

Table 1: Data Set Example

Data Set Sorted by X			Data Set 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

is the ability to assess the attribute values of the instance to be classified, and use this information to select attributes that discriminate the classes well for those particular values. As a result, for each instance we can select attributes that are useful for classifying that particular instance.

In this paper, we chose to use the entropy concept (Yang and Pedersen, 1997) to evaluate the quality of each attribute value for the classification of an instance. Specifically, entropy will be used to measure how well the values of the attributes of an instance determine its class. 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.

Let $D(A_1, A_2, ..., A_n, C)$, $n \geq 1$, be a data set with $n+1$ 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 Formula 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. Formula 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 data set $D(A_1, A_2, ..., A_n, C)$, an instance $I[v_1, v_2, ..., v_n]$ to be classified with its attribute values; 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 propose to evaluate the n attributes based on a lazy measure ($LazyEnt$), defined in Formula 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 is 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 will select the r attributes which present the r lowest $LazyEnt$ values.

4 Experimental Results

We implemented the lazy strategy described above within the Weka tool (Witten and Frank, 2005) – version 3.4.11 – and tested it in combination with different classifiers. The lazy selection occurs when the classifier receives a new instance to be classified. Since for each new instance a distinct subset of attributes must be considered by the classifier, the attributes not selected by the lazy strategy for a given instance are not removed from the data set, but only disregarded by the classification procedure.

As a baseline for comparison we used the eager attribute selection strategy most similar to our lazy strategy, which is the Information Gain Attribute Ranking technique (Yang and Pedersen, 1997). This technique is available within the Weka tool with the name "InfoGainAttributeEval". Both of them use the entropy concept for ranking attributes, are supervised strategies and need to know in advance the number of attributes that should be selected.

The strategies were initially tested on a large number of data sets from the UCI Machine Learning Repository (Asuncion and Newman, 2007). A total of 40 data sets which have a wide variation in size, complexity and application area were chosen. Table 2 presents some information about these data sets: name, number of attributes, number of classes and number of instances.

The entropy measure used to evaluate the quality of each attribute in our proposed technique requires discrete attribute values. Therefore we adopted the recursive entropy minimization heuristic proposed in (Fayyad and Irani, 1993) to discretize continuous attributes and coupled this with a minimum description length criterion (Rissanen, 1986) to control the number of intervals produced over the continuous space.

We have incorporated our lazy attribute selection strategy into two distinct classification techniques. In Subsection 4.1, we present the results obtained with the lazy k-Nearest Neighbor classifier and, in Subsection 4.2, we present the results obtained with the Naive Bayes. In Subsection 4.3, we explored a method for estimating if the lazy attribute selection strategy is most likely to have a better performance than the eager approach. And in

Table 2: Data sets from the UCI Repository

Data set	Attributes	Classes	Instances
anneal	38	5	898
audiology	69	24	226
autos	25	6	205
breast-cancer	9	2	286
breast-w	9	2	699
chess-Kr-vs-Kp	36	2	3196
credit-a	15	2	690
diabetes	8	2	768
flags	29	8	194
glass	9	6	214
heart-cleveland	13	2	303
heart-hungarian	13	2	294
hepatitis	19	2	155
horse-colic	27	2	368
hypothyroid	29	4	3772
ionosphere	34	2	351
labor	16	2	57
letter-recognition	16	26	20000
lymph	18	4	148
mol-bio-promoters	57	2	106
mol-bio-splice	60	3	3190
mushroom	22	2	8124
optdigits	64	10	5620
pendigits	16	10	10992
postoperative	8	3	90
primary-tumor	17	21	339
solar-flare1	12	6	323
solar-flare2	12	6	1066
sonar	60	2	208
soybean-large	35	19	683
spambase	57	2	4601
statlog-heart	13	2	270
statlog-segment	19	7	2310
statlog-vehicle	18	4	846
thyroid-sick	29	2	3772
vote	16	2	435
vowel	13	11	990
waveform-5000	40	3	5000
wine	13	3	178
zoo	17	7	101

Subsection 4.4, we evaluated the lazy strategy larger data sets from the NIPS 2003 challenge of feature selection (Guyon et al, 2004).

4.1 Experimental Results with k-NN

The k-NN algorithm assigns to a new instance to be classified the majority class among its k closest instances, from a given training data set (Cover and Hart, 1967; Dasarathy, 1991). The distance between each training instance and the new instance is calculated by a function defined on the values of their attributes. Hence, the use of lazy attribute selection implies that the calculation of the distances will be done using different subsets of attributes for different instances to be classified.

We compared the results of the original k-NN implementation available within the Weka tool, called IbK, with that obtained with our adapted version of this implementation which executes the lazy attribute selection before classifying each test instance. Both of them were executed with different values of the parameter k: 1, 3 and 5. In all experiments reported in this work, parameters not mentioned are set to their default values in the Weka tool.

Initially, we report the results obtained by both eager and lazy selection strategies when executed with three specific data sets: Wine, Heart-Hungarian and Vowel. The main goal of this first analysis is to show some results in detail so that our experimental methodology can be better understood. Also, it gives some evidence that the lazy strategy can indeed outperform the eager strategy for some data sets.

Tables 3, 4 and 5 show the predictive accuracies obtained by the k-NN algorithm, using both eager and lazy selection, with parameter k equal to 1 (columns 2 and 3), k equal to 3 (columns 4 and 5), and k equal to 5 (columns 6 and 7), for the Wine, Heart-Hungarian and Vowel data sets. Each row of these tables represents the execution of the selection strategies with a fixed number of attributes to be selected. The first column indicates the number of attributes to be selected as a percentage of the total number of attributes in the data set – the absolute number appears in parentheses. The last line (100%) represents the execution using the whole attribute set, that is, when no attribute selection is performed. Each value of predictive accuracy is obtained by a 10-fold cross-validation procedure (Han and Kamber, 2006). When comparing the two attribute selection strategies, bold-faced values indicate the best behavior.

Table 3 shows the results for the Wine data set. We can observe that the lazy strategy presented much better results than the eager strategy, regardless of the number of attributes we choose to select. The results with the lazy strategy are also better than the ones obtained without attribute selection. The lazy strategy was able to reach 100% of accuracy when just 3 out of 13 attributes were selected, when k was equal to 3 and 5. These first results are evidence that the lazy strategy can be better than the eager approach and better than the execution of the classifier without attribute selection.

Table 3: Accuracies for the Wine data set

Attributes	1-NN		3-NN		5-NN	
Selected	Eager	Lazy	Eager	Lazy	Eager	Lazy
10% (1)	82.6	**95.5**	82.6	**95.5**	82.6	**95.5**
20% (3)	95.5	**97.8**	94.4	**100.0**	94.4	**100.0**
30% (4)	95.5	**98.3**	93.3	**98.9**	93.3	**98.9**
40% (5)	97.2	**98.9**	93.3	**98.9**	93.3	**98.9**
50% (7)	97.8	**98.3**	96.6	**98.3**	96.6	**98.3**
60% (8)	98.9	98.9	96.6	**98.3**	96.6	**98.3**
70% (9)	97.2	**97.8**	94.9	**97.8**	94.9	**97.8**
80% (10)	97.2	**98.3**	94.4	**97.2**	94.4	**97.2**
90% (12)	96.6	**97.8**	96.1	**96.6**	96.1	**96.6**
100% (13)	98.3		96.1		96.1	

Table 4 presents the results obtained with the data set Heart-Hungarian. In this experiment, the eager strategy, mainly with number of attributes limited to 50%, had a better behavior than the lazy strategy. Both strategies presented the same accuracies when the number of selected attributes is greater than 60%. Also, with both strategies we are able to obtain better results than with no attribute selection.

Table 4: Accuracies for the Heart-Hungarian data set

Attributes	1-NN		3-NN		5-NN	
Selected	Eager	Lazy	Eager	Lazy	Eager	Lazy
10% (1)	**80.3**	78.6	**80.3**	79.3	**80.3**	79.3
20% (3)	**82.0**	81.3	**84.0**	83.0	**84.0**	83.0
30% (4)	**82.7**	81.3	**83.0**	82.7	**83.0**	82.7
40% (5)	**80.6**	79.6	**81.6**	81.0	**81.6**	81.0
50% (7)	80.6	80.6	**83.0**	82.7	**83.0**	82.7
60% (8)	81.0	**81.6**	82.7	**83.3**	82.7	**83.3**
70% (9)	80.3	80.3	83.0	83.0	83.0	83.0
80% (10)	80.3	80.3	82.3	82.3	82.3	82.3
90% (12)	80.3	80.3	82.3	82.3	82.3	82.3
100% (13)	80.3		82.3		82.3	

A third kind of result, shown in Table 5 for the Vowel data set, represents the cases where the selection of attributes does not contribute to the classification process. The results for this data set show that neither the lazy nor the eager strategies were able to outperform the results obtained without any attribute selection. This indicates that all attributes in this data set contribute to the overall classifier performance. It is worth observing that, in this experiment, the lower the number of selected attributes the worse is the accuracy, for both strategies.

Table 5: Accuracies for the Vowel data set

Attributes Selected	**1-NN Eager**	**1-NN Lazy**	**3-NN Eager**	**3-NN Lazy**	**5-NN Eager**	**5-NN Lazy**
10% (1)	26.5	**30.5**	26.5	**30.5**	26.5	**30.5**
20% (3)	51.6	**53.1**	50.1	**50.9**	50.1	**50.9**
30% (4)	**59.7**	59.3	52.2	**52.9**	52.2	**52.9**
40% (5)	**69.9**	68.9	**59.3**	58.0	**59.3**	58.0
50% (7)	76.5	**77.6**	64.2	**64.6**	64.2	**64.6**
60% (8)	80.1	**80.7**	**69.6**	68.1	**69.6**	68.1
70% (9)	82.7	82.7	71.2	71.2	71.2	71.2
80% (10)	86.3	86.3	74.0	74.0	74.0	74.0
90% (12)	89.8	89.8	**78.3**	78.1	**78.3**	78.1
100% (13)	89.8		78.6		78.6	

In Table 6, we summarize the results obtained by both eager and lazy strategies when executed for the 40 UCI data sets, using 1-NN, 3-NN and 5-NN classifiers. For each data set and classifier, we compare the accuracy of the strategies when we vary the percentage of attributes selected from 10% to 90% with a regular increment of 10%. The “Lazy” and “Eager” columns indicate the number of times each strategy obtained the higher predictive accuracy considering these nine different executions. Superior behavior is reported by bold-faced values. The “Ties” column represents the number of ties, that is, when both strategies obtained exactly the same accuracy. The last row reveals the total sum of best results obtained with each strategy.

As can be observed in the last row of Table 6, regardless of the k value, for most of the data sets the lazy strategy achieved a greater number of better results than the eager strategy. When k is equal to 1, the lazy strategy obtained 220 times the best results against 87 times for the eager strategy, with 53 ties. For k equal to 3 and 5, the behavior is very similar.

For the data sets Letter, Mol-Bio-Splice, Pendigits and Wine, the lazy strategy obtained the best results in all nine tests, regardless of the k parameter. For the data sets Chess-kr-vs-kp, Ionosphere, Labor, Primary-Tumor, Spambase, for at least one value of k, the lazy strategy achieved the best result in all nine tests. For no combination of data set and parameter k, the eager strategy was able to beat the lazy strategy in all nine tests.

For k equal to 1, in 28 out of the 40 data sets, the lazy strategy obtained a number of best results greater than the number of eager best results and greater than the number of ties. The eager strategy proceeded this way only 7 times. For k equal to 3 and 5, theses numbers of times were again 28 for the lazy and 7 for the eager strategy, which indicates a regularity in the results and that the good behavior of the lazy strategy does not greatly depend on the value of parameter k.

Up to this point, we have compared the strategies considering that they would select the same number of attributes. In the next analysis, we evaluate the results in a hypothetical scenario where we would know the best number of attributes to be selected for each strategy. In Table 7, for each data set, we report the best accuracies obtained with each attribute selection strategy considering the nine different selection percentage values. For 1-NN, 3-NN and 5-NN, we present, in the "Eager" and "Lazy" columns, the best accuracy obtained by each strategy. The number in parentheses represents the percentage of attributes selected with which this accuracy was obtained. The "No Sel" columns represent the accuracy obtained when no attribute selection was executed. For each data set and each k-NN group, the bold-faced values indicate the best result obtained. In the bottom row of this table, we indicate the number of times each selection strategy achieved the best overall result, out of the 40 UCI data sets.

Table 6: Number of executions where each strategy achieved the best result using the k-NN classifier

Data set	**1-NN**			**3-NN**			**5-NN**		
	Eager	**Lazy**	**Ties**	**Eager**	**Lazy**	**Ties**	**Eager**	**Lazy**	**Ties**
anneal	1	**5**	3	1	**5**	3	1	**5**	3
audiology	0	**5**	4	4	4	1	4	4	1
autos	**6**	1	2	**6**	2	1	**6**	2	1
breast-cancer	4	**5**	0	3	**6**	0	3	**6**	0
breast-w	1	**7**	1	4	4	1	4	4	1
chess-kr-vs-kp	0	**9**	0	2	**7**	0	2	**7**	0
credit-a	4	**5**	0	3	**6**	0	3	**6**	0
diabetes	3	3	3	**4**	2	3	**4**	2	3
flags	**7**	2	0	**6**	3	0	**6**	2	1
glass	0	**7**	2	0	**7**	2	0	**7**	2
heartcleveland	2	**4**	3	3	3	3	3	3	3
hearthungarian	**4**	2	3	**5**	2	2	**5**	1	3
hepatitis	1	**8**	0	2	**7**	0	2	**7**	0
horse-colic	4	**5**	0	4	**5**	0	4	**5**	0
hypo-thyroid	1	**8**	0	1	**8**	0	1	**8**	0
ionosphere	0	**9**	0	4	**5**	0	4	**5**	0
labor	0	**9**	0	1	**7**	1	2	**6**	1
letter	0	**9**	0	0	**9**	0	0	**9**	0
lymph	3	**6**	0	2	**7**	0	2	**7**	0
mol-bio-promot	4	**5**	0	4	**5**	0	4	**5**	0
mol-bio-splice	0	**9**	0	0	**9**	0	0	**9**	0
mushroom	0	3	6	0	3	6	0	3	6
optdigits	1	**6**	2	3	**5**	1	3	**5**	1
pendigits	0	**9**	0	0	**9**	0	0	**9**	0
postoperative	2	**4**	3	0	**6**	3	0	**5**	4
primary-tumor	3	**6**	0	0	**9**	0	0	**9**	0
solar-flare1	**7**	2	0	3	**6**	0	3	**5**	1
solar-flare2	**4**	3	2	1	**7**	1	1	**6**	2
sonar	**7**	2	0	2	**7**	0	2	**7**	0
soybean-large	1	**7**	1	0	**8**	1	0	**8**	1
spambase	0	**9**	0	3	**6**	0	3	**6**	0
statlog-heart	3	3	3	**4**	2	3	**4**	2	3
statlog-segment	1	**8**	0	1	**8**	0	1	**8**	0
statlog-vehicle	2	**7**	0	2	**7**	0	2	**7**	0
thyroid-sick	**5**	2	2	**5**	3	1	**5**	3	1
vote	2	**6**	1	1	**7**	1	1	**7**	1
vowel	2	**4**	3	3	**4**	2	3	**4**	2
waveform-5000	1	3	5	1	3	5	1	3	5
wine	0	**9**	0	0	**9**	0	0	**9**	0
zoo	1	4	4	**5**	3	1	**5**	3	1
Totals	87	**220**	53	93	**225**	42	94	**219**	47

Table 7: Best predictive accuracies achieved by each strategy using the k-NN classifier

Data set	1-NN			3-NN			5-NN		
	Eager	**Lazy**	**No Sel**	**Eager**	**Lazy**	**No Sel**	**Eager**	**Lazy**	**No Sel**
anneal	99.3 (60)	**99.6** (30)	99.2	98.0 (70)	**98.4** (20)	98.0	97.2 (80)	**97.8** (20)	97.2
audiology	75.7 (30)	**77.0** (20)	76.1	68.6 (20)	**72.1** (20)	65.9	69.5 (30)	**73.0** (20)	60.6
autos	**88.3** (50)	87.3 (50)	85.9	**81.5** (90)	**81.5** (90)	**81.5**	**77.1** (90)	**77.1** (90)	**77.1**
breast-cancer	72.7 (70)	**74.1** (30)	69.9	72.7 (70)	**73.4** (30)	70.3	74.1 (90)	**75.2** (70)	74.1
breast-w	**97.3** (70)	97.1 (40)	97.1	**97.0** (80)	**97.0** (90)	96.9	**97.1** (80)	**97.1** (90)	97.0
chess-kr-kp	96.4 (40)	**96.8** (40)	96.6	96.1 (40)	96.2 (40)	**96.6**	95.7 (40)	95.6 (40)	**96.1**
credit-a	**85.5** (10)	**85.5** (30)	82.3	85.5 (10)	**85.9** (30)	84.2	**85.9** (80)	85.8 (80)	84.6
diabetes	77.3 (60)	**78.0** (50)	76.4	78.0 (60)	**78.3** (50)	76.8	**78.1** (60)	78.0 (60)	77.2
flags	**63.4** (10)	60.8 (30)	59.8	**63.4** (40)	61.3 (30)	55.2	62.4 (30)	**63.4** (30)	57.7
glass	**77.1** (80)	**77.1** (80)	**77.1**	**75.7** (80)	**75.7** (80)	**75.7**	**73.8** (80)	**73.8** (80)	**73.8**
heart-cleveland	**84.5** (20)	82.8 (20)	80.5	**84.2** (20)	82.8 (20)	82.5	**83.5** (20)	82.8 (90)	82.8
heart-hungarian	**82.7** (30)	81.6 (60)	80.3	83.0 (60)	**83.7** (60)	83.0	**84.0** (20)	83.3 (60)	82.3
hepatitis	83.9 (40)	**86.5** (70)	83.9	**86.5** (50)	**86.5** (60)	83.9	84.5 (50)	**85.2** (50)	84.5
horse-colic	83.4 (20)	**83.7** (30)	78.5	**83.2** (20)	82.9 (30)	77.2	**83.2** (20)	82.1 (20)	77.4
hypo-thyroid	94.6 (20)	**97.0** (10)	91.5	94.8 (20)	**97.5** (10)	93.2	94.6 (20)	**97.3** (10)	93.3
ionosphere	93.4 (60)	**94.6** (40)	92.6	91.5 (50)	**92.3** (20)	90.6	90.9 (50)	**92.3** (10)	89.7
labor	96.5 (60)	**100** (60)	96.5	96.5 (90)	**98.2** (90)	96.5	**96.5** (80)	**96.5** (60)	91.2
letter	91.9 (70)	**91.9** (70)	91.9	90.4 (90)	**90.6** (90)	90.6	89.7 (90)	89.8 (90)	**89.8**
lymph	**85.1** (70)	**85.1** (90)	82.4	83.1 (80)	**83.8** (90)	**83.8**	84.5 (90)	**85.1** (90)	83.8
mol-bio-promot	**90.6** (10)	89.6 (10)	80.2	**91.5** (10)	88.7 (10)	80.2	87.7 (10)	**89.6** (30)	79.2
mol-bio-splice	89.7 (10)	**90.7** (10)	73.3	88.5 (10)	**91.2** (10)	77.4	88.2 (10)	**91.2** (10)	79.4
mushroom	**100** (40)	**100** (30)	**100**	**100** (40)	**100** (30)	**100**	**100** (40)	**100** (40)	**100**
optdigits	94.8 (70)	**94.8** (70)	94.3	95.4 (60)	**95.5** (60)	95.3	**95.6** (80)	95.5 (80)	95.5
pendigits	96.3 (90)	96.8 (90)	**97.1**	96.0 (90)	96.5 (90)	**96.7**	95.5 (90)	96.1 (90)	**96.5**
postoperative	**71.1** (20)	**71.1** (20)	63.3	**71.1** (10)	**71.1** (10)	68.9	**71.1** (10)	**71.1** (10)	**71.1**
primary-tumor	**42.8** (40)	42.5 (20)	38.3	44.2 (70)	**44.5** (30)	43.7	45.7 (90)	**46.6** (80)	46.3
solar-flare1	**72.8** (20)	71.5 (20)	65.9	**73.1** (20)	72.1 (20)	65.3	**71.2** (20)	70.9 (50)	66.6
solar-flare2	75.9 (20)	**76.3** (20)	73.5	75.7 (20)	**76.0** (20)	74.0	75.7 (20)	**76.2** (20)	73.8
sonar	**88.5** (60)	86.5 (60)	86.5	88.0 (90)	**88.5** (40)	86.1	83.7 (30)	**85.6** (80)	84.6
soybean-large	**93.4** (80)	**93.4** (80)	92.2	**92.4** (90)	**92.4** (90)	91.5	**91.9** (90)	**91.9** (90)	90.8
spambase	93.1 (90)	**93.7** (30)	93.0	93.3 (90)	**93.6** (90)	93.3	93.2 (90)	**93.3** (90)	93.2
statlog-heart	84.8 (30)	**85.2** (50)	84.1	**84.8** (30)	84.4 (50)	80.7	85.6 (50)	**85.9** (40)	82.2
statlog-segment	**94.8** (90)	94.7 (90)	94.7	93.9 (90)	93.9 (90)	**94.0**	**93.1** (90)	92.9 (90)	**93.1**
statlog-vehicle	**71.4** (50)	**71.4** (90)	70.9	71.4 (90)	**71.5** (80)	71.3	70.6 (90)	**71.0** (70)	70.7
thyroid-sick	**97.7** (40)	97.5 (60)	97.5	**97.5** (10)	97.3 (20)	97.1	**97.6** (40)	97.5 (20)	96.6
vote	95.2 (50)	**96.1** (10)	92.2	95.2 (50)	**95.6** (20)	92.4	95.2 (10)	**95.4** (10)	92.9
vowel	**89.8** (90)	**89.8** (90)	**89.8**	84.1 (90)	84.1 (90)	**84.6**	78.3 (90)	78.1 (90)	**78.6**
waveform-5000	74.6 (40)	**75.5** (10)	73.8	79.0 (40)	**79.1** (40)	78.6	80.8 (40)	**80.8** (40)	79.7
wine	**98.9** (60)	**98.9** (40)	98.3	97.2 (50)	**98.9** (20)	96.6	96.6 (50)	**100** (20)	96.1
zoo	**97.0** (80)	**97.0** (80)	96.0	**97.0** (80)	**97.0** (80)	93.1	**93.1** (90)	92.1 (80)	**93.1**
Total No. of Wins	21	28	4	15	29	8	17	26	10

We observe that, in this scenario, regardless of the parameter k, the lazy strategy tends to achieve greater accuracies than the eager strategy. For k equal to 1, the lazy strategy achieved the best accuracy 28 times whereas the eager strategy achieved the best accuracy 21 times. For k equal to 3, the results were 29 for the lazy strategy and 15 for the eager strategy. And for k equal to 5, similar results were obtained: 26 for the lazy strategy and 17 for the eager strategy. We note that for just a few of the data sets, the attribute selection is not useful, that is, for about 10 data sets the executions without attribute selection – reported in the "No Sel" column – obtained the best result. Another remarkable result is that for some data sets, like Wine, Audiology, Hypo-Thyroid and Ionosphere, the maximum accuracy was obtained with a relatively small number of selected attributes.

In the results presented so far, we have compared accuracies without taking into account statistical significance. Next, we employ the paired two-tailed Student's t-test technique with the goal of identifying which compared predictive accuracies are actually significantly different.

In the comparisons we conducted before, each predictive accuracy value was calculated as the average of a 10-fold cross-validation procedure. Now, we look at the 10 individual results from each fold to apply the paired t-test analysis. Table 8 presents the result of this statistical analysis. Each row represents the results obtained by a different application of k-NN, with k equal to 1, 3 and 5. The second and third columns represent the numbers reported in the last row of Table 6, which is the number of times each strategy (eager and lazy) obtained the better accuracy value. In parentheses, we find the number of times each strategy obtained the better accuracy, considering a statistical significance with a p-value less than 0.05, which means that the probability of the difference of performance being due to random chance alone is less than 0.05. The last column shows the number of cases in which the two accuracy values were equal. We can observe that the results with the lazy strategy were again superior to the ones with the eager strategy. For the 1-NN executions, the lazy strategy obtained the best results with statistical significance 87 times against just 7 times for the eager strategy. For 3-NN and 5-NN the results are similar and show a better performance of the lazy strategy.

We also note that after taking into account the statistical significance test, the lazy strategy achieved an accuracy better or equal to the eager strategy in 82.5% of the data sets, regardless of the number of attributes selected. This indicates that selecting the attributes in a lazy way is generally a better

Table 8: Number of wins of each attribute selection method and number of statistically significant wins according to a t-test with significance level of 0.05

Classifier	**Eager**	**Lazy**	**Tie**
1-NN	87 (7)	220 (87)	53
3-NN	93 (10)	225 (84)	42
5-NN	94 (8)	219 (78)	47

choice than performing the eager selection in a preprocessing phase.

4.2 Experimental Results with Naive Bayes

Similarly to the experiments with k-NN, the lazy selection strategy was assessed in combination with the Naive Bayes classifier, which is a classification technique based on the Bayes theorem. 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 reasonable results in practice.

We have incorporated the lazy attribute selection into the eager implementation of the Naive Bayes classifier available within the Weka tool, called NaiveBayes. The lazy attribute selection was executed within the algorithm just before the actual classification takes place. The Naive Bayes predictor considered only the r attributes selected in a lazy manner to compute its probabilities, for each test instance. In the same manner as in the k-NN experiments, this implies that for different instances distinct subsets of attributes were used. The experiments were executed with the default parameter settings in the Weka tool, using the original Naive Bayes classifier. No other relevant features were altered to implement the lazy selection.

Table 9 shows the number of times each strategy, either lazy or eager, achieved a higher accuracy than the other, for the nine executions in which we varied the percentage of attributes between 10% and 90%, in increments of 10%. As seen in the last row labeled "Totals", the results show a prevalence of the lazy strategy over the eager strategy, even though this prevalence is less pronounced that the one we observed with k-NN.

Table 9: Experiments with the Naive Bayes classifier

Data set	**Eager**	**Lazy**	**Ties**
anneal	3	**4**	2
audiology	0	**8**	1
autos	**7**	2	0
breast-cancer	**7**	2	0
breast-w	2	**6**	1
chess-kr-vs-kp	**5**	4	0
credit-a	4	**5**	0
diabetes	**4**	2	3
flags	**6**	1	2
glass	2	**5**	2
heart-cleveland	3	**4**	2
heart-hungarian	3	1	**5**
hepatitis	1	**7**	1
horse-colic	**5**	4	0
hypo-thyroid	**8**	1	0
ionosphere	2	**7**	0
labor	2	**4**	3
letter	0	**9**	0
lymph	2	**6**	1
mol-bio-promoters	**6**	3	0
mol-bio-splice	**9**	0	0
mushroom	3	**5**	1
optdigits	0	**8**	1
pendigits	0	**9**	0
postoperative	1	**5**	3
primary-tumor	1	**8**	0
solar-flare1	**5**	4	0
solar-flare2	**5**	3	1
sonar	2	**7**	0
soybean-large	1	**7**	1
spambase	0	**9**	0
statlog-heart	3	3	3
statlog-segment	1	**7**	1
statlog-vehicle	1	**8**	0
thyroid-sick	1	**5**	3
vote	**5**	3	1
vowel	**4**	3	2
waveform-5000	1	3	**5**
wine	0	**8**	1
zoo	3	**5**	1
Totals	118	195	47

In Table 10 the best accuracies for each selection strategy are presented for the Naive Bayes classifier, similarly to what was shown in Table 7 for the k-NN classifier. The lazy strategy achieved the best accuracy in 27 cases and the eager strategy in 24 cases. These results show that a considerable number of data sets can benefit from the lazy selection.

Finally, a Student's t-test was performed for the Naive Bayes results, with the same parameters used before for the k-NN results, i.e., $p = 0.05$ with a paired two-tailed setup. The results are the following: the eager strategy prevailed 118 times over the lazy strategy, but only 25 times with statistical significance. The lazy strategy outperformed the eager strategy 195 times, and from these 80 times with statistical significance. In 47 tests a definitive tie occurred.

It is worth reporting that the computational cost introduced by the lazy strategy into the k-NN and Naive Bayes classification procedure is not considerable. The average execution time of the lazy selection procedure for each instance varied from 0.08 milliseconds (for the Autos data set) to 0.31 milliseconds (for the Lymph data set). This is negligible considering that for these two data sets the average execution time of the 3-NN classification of an instance was 1.7 and 10.4 milliseconds. These experiments were performed on a 2.0 GHz Intel Core 2 Duo CPU 4400 with 2 Gbytes of RAM.

4.3 Predicting the lazy strategy effectiveness

In spite of the promising results showed before, we can see that it is not always the case that selecting attributes in a lazy fashion is preferable to executing an eager selection. Therefore, it could be interesting to have a method for estimating when the lazy strategy is most likely to be useful.

Intuitively, the lazy strategy is more valuable when there is a great variation in the entropy derived from each different value of an attribute. When this is the case, some attributes are likely to be important for some of the instances but not for the others. Therefore, we need to evaluate for each attribute A_j, $1 \leq j \leq n$, of a data set $D(A_1, A_2, ..., A_n, C)$, the average of the modulus (absolute value) of the differences between the value $Ent(D, A_j)$ and each of its values $Ent(D, A_j, a_{ji})$, $1 \leq i \leq k_j$, for each value a_{ji} of the attribute A_j. This average value is then defined by:

$$V(D, A_j) = \frac{1}{k_j} * \sum_{i=1}^{k_j} [|Ent(D, A_j, a_{ji}) - Ent(D, A_j)|]. \tag{5}$$

Table 10: Accuracies by the Naive Bayes classifier

Data set	**Eager**		**Lazy**		**No Sel**
anneal	94.9	(80)	**96.3**	(10)	94.9
audiology	74.3	(40)	**76.6**	(60)	73.0
autos	**81.0**	(20)	77.6	(20)	72.7
breast-cancer	**73.1**	(40)	72.4	(30)	71.7
breast-w	**97.6**	(90)	**97.6**	(90)	97.0
chess-kr-vs-kp	89.6	(20)	**91.2**	(20)	87.7
credit-a	**86.7**	(70)	86.5	(70)	86.4
diabetes	78.8	(60)	**79.0**	(60)	78.1
flags	**62.9**	(60)	61.9	(90)	61.3
glass	**74.3**	(80)	**74.3**	(80)	**74.3**
heart-cleveland	**84.5**	(20)	84.2	(50)	82.8
heart-hungarian	**84.0**	(20)	**84.0**	(50)	**84.0**
hepatitis	85.8	(50)	**86.5**	(40)	83.9
horse-colic	**85.6**	(20)	84.5	(20)	82.1
hypo-thyroid	**95.3**	(60)	94.8	(80)	**95.3**
ionosphere	92.0	(50)	**92.6**	(10)	90.9
labor	**98.3**	(60)	**98.3**	(70)	**98.3**
letter	74.3	(70)	**74.7**	(70)	74.0
lymph	**86.5**	(70)	**86.5**	(80)	**86.5**
mol-bio-promot	**94.3**	(10)	93.4	(80)	93.4
mol-bio-splice	**96.1**	(60)	95.3	(90)	95.5
mushroom	98.8	(10)	**99.4**	(10)	95.5
optdigits	**92.5**	(90)	**92.5**	(70)	**92.5**
pendigits	87.2	(90)	87.5	(90)	**87.8**
postoperative	**71.1**	(20)	**71.1**	(20)	70.0
primary-tumor	49.6	(80)	**50.7**	(80)	49.3
solar-flare1	**72.8**	(30)	72.1	(30)	65.0
solar-flare2	75.2	(30)	**75.7**	(10)	74.8
sonar	68.3	(10)	**71.6**	(10)	67.8
soybean-large	89.9	(80)	**90.0**	(30)	89.9
spambase	90.5	(20)	**91.6**	(10)	90.2
statlog-heart	**84.4**	(50)	**84.4**	(40)	83.0
statlog-segment	**91.6**	(90)	**91.6**	(90)	**91.6**
statlog-vehicle	62.9	(60)	**63.0**	(70)	62.2
thyroid-sick	**97.2**	(10)	**97.2**	(30)	97.0
vote	**95.2**	(10)	94.3	(10)	90.1
vowel	**57.7**	(70)	**57.7**	(70)	52.7
waveform-5000	**81.1**	(40)	81.1	(40)	80.7
wine	98.9	(90)	**100.0**	(30)	98.9
zoo	**96.0**	(80)	**96.0**	(80)	93.1
Total No. of Wins	24		27		8

For each data set, we can compute the metric $V(D)$ by taking the average of the values $V(D, A_j)$ for all attributes A_j. The higher this value, the most likely it is that the data set D would benefit from the lazy strategy. However, to define a more specific and appropriate metric, we need to take into account the number of attributes to be selected. Therefore, for a data set D and a percentage x of attributes to be selected, we calculate the metric $V(D, x)$ by the average of the $V(D, A_j)$ values for the set of attributes with the highest $x\%$ values of $V(D, A_j)$. Analogously, the higher the $V(D, x)$ value, the most likely it is that the data set D would benefit from the lazy strategy when selecting $x\%$ of the attributes.

Figure 1 shows that a high value for this metric does indeed imply a better confidence in the superiority of the lazy attribute selection. This analysis is based on all executions of the lazy and eager strategies with the 1-NN classifier, taking as input all the 40 UCI data sets (D) and the variation of the percentage (x) of selected attributes from 10% to 90%, in increments of 10%, which represents a total of 360 executions per strategy.

For each threshold value in the horizontal axis the correspondent percentage value in the vertical axis, in Curve 1, represents the percentage of the cases where the lazy strategy achieved higher predictive accuracy than the eager strategy out of all the 360 cases where the respective value of $V(D, x)$ is greater than or equal to the corresponding threshold value in the horizontal axis. In Curve 2, only the comparisons with statistical significance, out of the 360 cases, are considered in the percentage calculation.

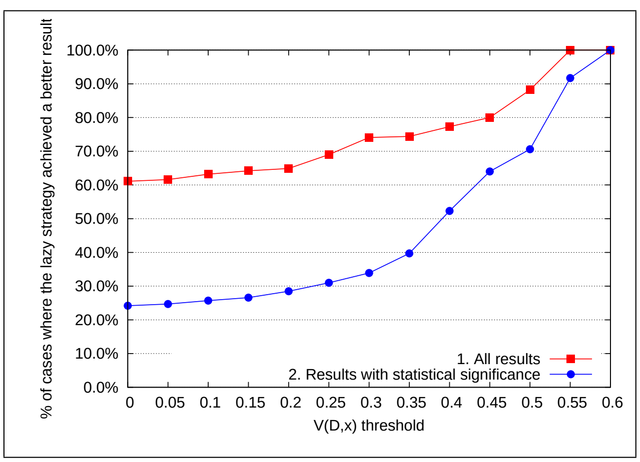

Figure 1: Evaluation of the $V(D, x)$ metric.

The leftmost point of the Curve 1 indicates that, for all combinations of D and x in which $V(D, x) \geq 0$ (in all 360 cases) the lazy strategy in about 61.1% of the time (or 220 out of 360 executions) achieved better results than the eager strategy. When we restrict the experiments to just the data sets D and value x in which $V(D, x) \geq 0.2$ the lazy strategy prevailed 64.9% of the time (148 out of 228 executions) and for $V(D, x) \geq 0.4$, the results indicate 77.3% of prevailing lazy executions (or 34 out of 44 total executions).

The same occurs in the experiments conducted with a statistical significance analysis, represented by Curve 2: the higher the value of $V(D, x)$, the most probable it is that the lazy selection strategy achieves a better result than the eager strategy.

Similar results were obtained with the 3-NN, 5-NN and Naive Bayes classifiers, showing that indeed this metric can be used to estimate with more conviction whether the lazy strategy is able to yield a superior result for a specific data set, given the percentage of attributes to be selected.

4.4 Experiments with large data sets

The experiments with the UCI data sets revealed that the k-NN and Naive Bayes classifiers benefit from the lazy attribute selection in most cases.

For these data sets the number of features varies, between 8 and 69, so we cannot infer from these experiments the behavior of the lazy attribute selection for data sets with a much larger number of attributes.

In order to evaluate if the lazy selection is scalable and effective on larger data sets, additional experiments were performed with data sets from the NIPS 2003 challenge on feature selection (Guyon et al, 2004). This competition took place in the NIPS 2003 conference, and made available five data sets to be used as benchmarks for attribute selection methods.

The experiments with these large data sets were performed as follows. There were three sets of instances per data set – training, validation and test – and the class label of each instance was provided only for the training and validation data. These two collections were merged in one data set, and the cross-validation procedure adopted in the earlier experiments was also employed for them. Table 11 summarizes the characteristics of each NIPS data set: name, number of attributes, number of classes and their distribution, and the total number of instances.

The same procedure to discretize continuous attributes was adopted for these data sets, except for the Dorothea, which has only binary attributes (0/1).

Table 11: Data sets from NIPS 2003 Feature Selection Challenge

Data set	**Attributes**	**Classes**	**Instances**	**Class A**	**Class B**
Arcene	10000	2	200	56%	44%
Madelon	500	2	2600	50%	50%
Gisette	5000	2	7000	50%	50%
Dexter	20000	2	600	50%	50%
Dorothea	100000	2	1150	90%	10%

Some irrelevant and random attributes, referred to as "probes" in (Guyon et al, 2004), are present in these data sets. In many cases, these attributes were discretized into a single bin by the discretization procedure. When this happened, we removed the attribute from the data set.

Table 12 shows the number of times each strategy achieved a higher accuracy on these data sets, using the same procedure from the earlier experiments: nine executions with the percentage of attributes selected varying from 10% to 90%. Again, a major predominance of best results for the lazy strategy was achieved. Also, the total number of victories taking into account statistical significance confirmed a higher number of successes with the lazy selection.

Table 12: Number of executions where each strategy achieved the best result using the NIPS 2003 large data sets

Data set	**1-NN**		**3-NN**		**5-NN**		**NB**	
	Eager	**Lazy**	**Eager**	**Lazy**	**Eager**	**Lazy**	**Eager**	**Lazy**
Arcene	4	**5**	3	**6**	**6**	3	0	**9**
Madelon	3	**6**	3	**6**	3	**6**	3	**6**
Gisette	0	**9**	1	**8**	0	**9**	0	**9**
Dexter	0	**9**	1	**8**	0	**9**	0	**9**
Dorothea	**7**	2	**8**	1	**9**	0	2	**7**
Totals	14	**31**	16	**29**	18	**27**	5	**40**
Totals with statistical significance (p=0.05)	6	**10**	6	**8**	5	**13**	0	**22**

The best accuracies achieved by each strategy are shown in Table 13. Each result is summarized for the five data sets, using both the k-NN and

Naive Bayes classifiers. For each data set and classifier, we compare the accuracy of the strategies when we vary the percentage of attributes selected from 10% to 90% with a regular increment of 10%. The "Lazy" and "Eager" rows present the best accuracy obtained by each strategy. The number in parentheses represents the percentage of attributes selected with which this accuracy was obtained. The "No Sel" rows represent the accuracy obtained when no attribute selection was executed. For each data set and each classifier, the bold-faced values indicate the best result obtained.

Similarly to the experiments with the UCI repository, the results with larger data sets from the NIPS 2003 feature selection challenge have also presented favorable results for the lazy strategy. For three data sets – Arcene, Gisette and Dexter – the best overall accuracies were reached by the lazy strategy. The data set Madelon seems not to benefit from any of the attribute selection strategies, and the Dorothea data set was the only one for which the eager approach achieved better accuracy values. These results indicate that the lazy strategy can lead to a better behavior also for larger data sets.

Table 13: Best predictive accuracies achieved by each strategy using the NIPS 2003 large data sets

		Arcene		Madelon		Gisette		Dexter		Dorothea	
1-NN	Eager	90.0	(80%)	71.9	(90%)	96.7	(80%)	90.2	(50%)	**91.5**	(10%)
	Lazy	**92.5**	(80%)	72.7	(90%)	**97.1**	(60%)	**93.7**	(20%)	90.3	(10%)
	No Sel	91.0		**73.0**		96.8		88.3		90.6	
3-NN	Eager	87.0	(10%)	72.0	(90%)	96.8	(90%)	91.0	(90%)	**91.6**	(40%)
	Lazy	87.5	(70%)	72.6	(90%)	**97.4**	(20%)	**93.5**	(20%)	90.3	(10%)
	No Sel	**88.0**		**73.0**		96.8		90.2		90.3	
5-NN	Eager	**88.0**	(60%)	71.5	(90%)	96.6	(90%)	92.2	(90%)	**92.6**	(50%)
	Lazy	87.0	(60%)	**72.4**	(90%)	**97.2**	(10%)	**94.0**	(20%)	90.3	(10%)
	No Sel	87.5		72.3		96.7		91.8		90.3	
NB	Eager	70.0	(10%)	**65.1**	(40%)	90.0	(90%)	94.5	(90%)	**91.8**	(10%)
	Lazy	**80.5**	(20%)	64.2	(20%)	**90.3**	(30%)	**95.2**	(20%)	90.6	(20%)
	No Sel	67.0		62.8		90.2		94.5		90.0	

5 Conclusions

In this paper, we have proposed using a lazy strategy to perform attribute selection for the classification problem. Although our strategy is general, here we concentrated on a specific version based on entropy ranking and compare it with the analogous eager strategy. Our experimental results show that by postponing the choice of attributes to the moment when a new instance is ready to be classified, we can in most cases improve the accuracy of classification, when compared with the attribute selection performed eagerly as a data preprocessing phase. We have also proposed a metric that can be used to predict if a specific data set might take advantage of the lazy attribute selection approach.

The proposed lazy selection strategy is naturally able to work embedded in lazy classifiers. Thus, it is particularly appropriate for traditional lazy classification techniques, such as k-NN, for eager techniques that can be easily implemented in a lazy scenario, such as Naive Bayes, and for lazy versions of other eager techniques, such as lazy decision trees, lazy associative classification and lazy rule induction. In an extreme case, one could also consider using the lazy selection strategy in combination with a traditional eager technique. When an instance is presented for classification the most appropriate attributes would be selected in a lazy fashion and then used to construct a model for that instance using the eager technique. This would be computationally expensive because a complete model would be constructed for each instance, but could still be profitable given the potential gain in predictive accuracy associated with lazy selection as shown in this paper (and recalling that predictive accuracy is normally considered significantly more important than computational time in classification).

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 model to filter strategies that evaluate subsets of attributes instead of weighting them individually, like Correlation-based Feature Selection (Hall, 2000) and Consistency-based Feature Selection (Liu and Setiono, 1996). Furthermore, we expect to be able to apply the lazy idea to a wrapper attribute selection technique, and evaluate its results and performance.

References

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

Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2:121–168

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

Dasarathy BV (1991) Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press

Duda RO, Hart PE, Stork DG (2001) Pattern Classification, 2nd edn. John Wiley & Sons

Fayyad UM, Irani KB (1993) Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI'93), pp 1022–1029

Friedman JH, Kohavi R, Yun Y (1996) Lazy decision trees. In: Procs. of the 13th National Conference on Artificial Intelligence (AAAI'96), pp 717–724

Gora G, Wojna A (2002) Local attribute value grouping for lazy rule induction. In: Proceedings of the 3rd International Conference on Rough Sets and Current Trends in Computing (RSCTC'02), LNAI 2475, pp 405–412

Guyon I, Elisseeff A (2006) An introduction to feature extraction. In: Guyon I, Gunn S, Nikravesh M, Zadeh L (eds) Feature Extraction, Foundations and Applications, Springer, pp 1–24

Guyon I, Gunn S, Ben-Hur A, Dror G (2004) Result analysis of the nips 2003 feature selection challenge. NIPS, MIT Press 17:545–552

Guyon I, Gunn S, Nikravesh M, Zadeh L (eds) (2006) Feature Extraction, Foundations and Applications. Springer

Hall MA (2000) A correlation-based feature selection for discrete and numeric class machine learning. In: Proceedings of the 17th International Conference on Machine Learning (ICML'00)

Han J, Kamber M (2006) Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann

Kira K, Rendell L (1992) A practical approach to feature selection. In: Proceedings of the 9th International Conference on Machine Learning (ICML'92), pp 249–256

Kononenko I (1994) Estimating attributes: Analysis and extensions of relief. In: Proceedings of the 7th European Conference on Machine Learning (ECML'94), pp 171–182

Liu B, Hsu W, Ma Y (1998) Integrating classification and association rule mining. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD'98), pp 80–86

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

Liu H, Motoda H (2008b) Less is more. In: Liu H, Motoda H (eds) Computational Methods of Feature Selection, Chapman & Hall/CRC, pp 3–17

Liu H, Setiono R (1996) A probabilistic approach to feature selection: A filter solution. In: Proceedings of the 13th International Conference on Machine Learning (ICML'96), pp 319–327

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

Quinlan JR (1986) Induction of decision trees. Machine Learning 1:81–106

Quinlan JR (1993) C4.5: Programs for Machine Learning. Morgan Kaufmann

Ripley BD (1996) Pattern Recognition and Neural Networks. Cambridge University Press

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

Veloso A, Meira W, Zaki MJ (2006) Lazy associative classification. In: Proceedings of the 6th International Conference on Data Mining (ICDM'06), pp 645–654

Witten IH, Frank E (2005) Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann

Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. In: Proceedings of the 14th International Conference on Machine Learning (ICML'97), pp 412–420