Postponing the Evaluation of Attributes with a High Number of Boundary Points

Tapio Elomaa[1] and Juho Rousu[2]

[1] Joint Research Centre, European Commission, elomaa@cs.helsinki.fi
[2] VTT Biotechnology and Food Research, juho.rousu@vtt.fi

Abstract. The efficiency of the otherwise expedient decision tree learning can be impaired in processing data-mining-sized data if superlinear-time processing is required in attribute selection. An example of such a technique is optimal multisplitting of numerical attributes. Its efficiency is hit hard even by a single troublesome attribute in the domain.
Analysis shows that there is a direct connection between the ratio of the numbers of boundary points and training examples and the maximum goodness score of a numerical attribute. Class distribution information from preprocessing can be applied to obtain tighter bounds for an attribute's relevance in class prediction. These analytical bounds, however, are too loose for practical purposes.
We experiment with heuristic methods which postpone the evaluation of attributes that have a high number of boundary points. The results show that substantial time savings can be obtained in the most critical data sets without having to give up on the accuracy of the resulting classifier.

1 Introduction

Identifying and eliminating either irrelevant attributes [4, 13] or untrustworthy training examples [3, 12, 17] prior to classifier construction are techniques used to aid and enhance the induction process (for a comprehensive survey see [1]). Such cleaning methods can be heavier than the actual process of building a classifier. Moreover, irreversible decisions to remove attributes or examples are taken. In this paper we explore efficient ways of enhancing the induction process by overlooking some attributes at some stages, but without losing the possibility to use them later if they turn out to be beneficial then.

Inductive process that is based on univariate partitioning of the given data set—e.g., top-down induction of decision trees—is inherently myopic to interrelations between attributes. Its stronghold is the extreme efficiency on mid-sized data sets. However, when large databases are processed even this advantage may vanish; in particular, if the attribute selection entails processing that requires superlinear time in the number of examples or some other characteristic figure.

Evaluating nominal attributes is efficient. Numerical attribute domains, on the other hand, need to be discretized, which may be time consuming if the domain at hand has a very high number of candidate cut points. Even a linear-time method like *binarization* can require substantial amount of time. This presents

a particular problem for learning algorithms that have to manipulate numerical attributes exhaustively; e.g., optimal [8, 11] or greedy [10] multisplitters in decision tree learning. The inconvenience for all attribute selection strategies alike is that the time consumption of attribute selection is dominated by the attributes that require the heaviest evaluation. Hence, even a single difficult attribute can ruin the efficiency of an otherwise manageable domain.

This paper studies how *boundary points* [9] can be utilized to determine the relevance of an attribute in univariate induction. It is shown that an attribute with many boundary points is not relevant for class prediction. As evaluating such an attribute is also time consuming, postponing its evaluation should turn out beneficial in the resulting classifiers quality and speed of classifier construction. We do not want to trade accuracy for efficiency or simplicity, but strive to maintain the prediction ability of the resulting decision tree while speeding up the classifier construction by simple and efficient dynamic data processing.

During the iterative top-down induction of a decision tree the number of boundary points that have to be taken into account in one dimension decreases, since the recursive partitioning of the data removes possible cut points—and boundary points as well. Also, the number of available training examples decreases during tree construction as the training set gets partitioned into smaller and smaller subsets. Due to this dynamics, we do not definitely disregard an attribute, which at some point has a too high number of boundary points, but keep it for further evaluation in the changed situation.

2 Preliminaries

All numerical dimension of data represented as attribute value assignments share as a common characteristic figure the *number of instances*, n. Another characteristic figure is the *number of different values*, V, for the attribute. Numerical attributes can have a very large, even infinite, domain. As a third figure numerical dimensions have the *number of boundary points*, $B - 1$. Intuitively, boundary points are such values of a numerical value range that partitioning the data with those values as thresholds will not needlessly separate two instances of the same class to different subsets of the partition. Such a partitioning will not obviously harm the prediction of the example class labels.

The basic relationships of these three figures are $B \leq V \leq n$, but it is the common (mis)conception that $B \ll V \ll n$ in real-world data. Recently the relationship of these figures have been studied in detail [8] for a large collection of the most commonly used machine learning data sets from the UCI data repository [16]. It turns out that most typically the number of boundary points in a numerical dimension is at least half of the total number of existing values in the data. The claim $V \ll n$ is better grounded, and $B \ll n$ even more so.

The minimum preprocessing in handling a numerical attribute is to sort the training data by its value. The data cannot be partitioned in this dimension so that two examples with equal values for the underlying attribute would belong to different subsets. Therefore, we can consider a categorized version of the data,

where all examples with an equal value constitute a *bin* of examples. There are as many bins as distinct values for the attribute, V.

Fayyad and Irani's [9] analysis of the binarization technique proved that for the information gain function [18] only *boundary points* need to be considered as potential cut points, because optimal binary splits always fall on them due to the convexity of the function. Codrington and Brodley [5] present further studies of the convexity properties of many common attribute evaluation functions.

Definition 1. Let a sequence S of examples be sorted by the value of a numerical attribute A. The set of *boundary points* is defined as follows: A value $T \in \mathrm{Dom}(A)$ is a boundary point if and only if there exists a pair of examples $s_1, s_2 \in S$, having different classes, such that $\mathrm{val}_A(s_1) = T < \mathrm{val}_A(s_2)$; and there does not exist another example $s \in S$ such that $\mathrm{val}_A(s_1) < \mathrm{val}_A(s) < \mathrm{val}_A(s_2)$.

In the original definition a boundary point was taken to be a value that is strictly in between the values $\mathrm{val}_A(s_1)$ and $\mathrm{val}_A(s_2)$ [9]. The above definition leads to partitions with the same subsets. Let us now define a *block* of examples. It is a concept that facilitates the discovery of all boundary points of a data set.

Definition 2. Let the example set S be ordered by the value of a numerical attribute A. Let C be the class attribute. A *block* of examples is a *maximal-length* sequence of consecutive examples $s_1, \ldots, s_b \subseteq S$ such that

1. $\mathrm{val}_C(s_1) = \ldots = \mathrm{val}_C(s_b)$ and there does not exist an example $s \in S$ such that $\mathrm{val}_A(s_1) \leq \mathrm{val}_A(s) \leq \mathrm{val}_A(s_b)$ and $\mathrm{val}_C(s) \neq \mathrm{val}_C(s_1)$, or
2. $\mathrm{val}_A(s_1) = \ldots = \mathrm{val}_A(s_b)$ and there exists s_i, $i \in \{2, \ldots, b\}$, such that $\mathrm{val}_C(s_i) \neq \mathrm{val}_C(s_1)$.

Blocks of type (1) are *uniform* ones and those of type (2) are *mixed* ones. Boundary points of a set are exactly the borders of blocks, which makes finding them simple. Blocks are obtained from bins by merging only adjacent class uniform bins with the same class label into a block. Mixed bins are never merged into a block with another bin.

In decision tree learning the number of boundary points in a numerical dimension depends on the phase of tree construction: it is the highest at the root level, when the data has not yet been partitioned, reduces through some splits defined by other attributes, until finally, at the level of the last decision nodes, it reaches a linear correlation with the decision tree's accuracy on the training data (if the numerical attribute in question is to be chosen to the tree).

A *well-behaved* function always has an optimal multisplit on boundary points [8]. All the most commonly used attribute evaluation functions are well-behaved. By using a well-behaved function we may concentrate on boundary points independent of whether the partition arity is limited *a priori* or not. If a well-behaved evaluation function also has the so-called *cumulativity* property, the general optimal partitioning algorithm of Fulton *et al.* [11] can be adapted to operate in time that is quadratic in the number of blocks instead of bins.

3 Boundary points as an indication of attribute relevance

Let us study the well-behaved evaluation function *average class entropy*, ACE. For a partition $\biguplus_i S_i$ of the data set S, ACE is defined to be

$$ACE(\biguplus_i S_i) = (1/|S|) \sum_i |S_i| H(S_i) = (1/n) \sum_i |S_i| H(S_i),$$

where H is the entropy function: $H(S) = -\sum_{j=1}^{m} P(C_j, S) \log_2 P(C_j, S)$, in which m denotes the number of classes and $P(C, S)$ stands for the proportion of examples in S that have class C.

Let us bound the minimum value of average class entropy in the following situation. We are partitioning a numerical attribute's value range into ℓ intervals; there are n training examples and the domain in question contains B blocks.

Since ACE is a well-behaved function, its optimal ℓ-partition is defined by $\ell - 1$ boundary points. Hence, there are $B - \ell$ further boundary points within the partition subsets. It pays to maximize the number of examples belonging to partition subsets that have zero entropy, i.e., such examples that belong to class uniform intervals. To that end, intervals into which the unused boundary points fall, have to be as short as possible. That is obtained if each example in such a subset alone constitutes an uniform block, then there is a boundary point in between every pair of consecutive examples. We are approximating the minimum value of ACE, so we can freely assume there to be only two classes.

Let us now settle the question into how many subsets should the extra boundary points be distributed. As the above motivation shows $|S_i| H(S_i)/b_i$ minimizes when $b_i = |S_i| - 1$. It can be easily verified that the function $|S_i| H(S_i)/b_i = H(S_i)|S_i|/(|S_i| - 1)$ decreases monotonically when $|S_i|$ increases and, hence, it holds that $\sum_{i=1}^{\ell} |S_i| H(S_i)/b_i \geq |\bigcup_{i=1}^{\ell} S_i| H(\bigcup_{i=1}^{\ell} S_i)/(\sum_{i=1}^{\ell} b_i)$ for any set of subsets $S_1, \ldots, S_\ell$ which contain $b_i = |S_i| - 1$ boundary points each. Therefore, packing the extra boundaries into a single interval will lead to a smaller ACE value than segregating the boundary points.

The above construction gives the idealized minimum value of ACE: No other partition subset, except the one into which all unused boundary points have been packed, contributes to the impurity of the partition. Hence, the average class entropy of the partition is $ACE(\biguplus_{i=1}^{\ell} S_i) \geq (B - \ell)/n$. In other words, the lowest obtainable average class entropy of a partition depends directly on the ratio B/n.

Due to the heavily idealized assumptions underlying the above calculations, we do not expect this lower bound to be very tight. Nevertheless, it shows that there is a direct correlation between the B/n ratio and an attribute's relevance for class prediction in univariate induction. The way to apply the bound is straightforward: if the ratio $(B - \ell)/n$ shows that by partitioning the data along this dimension cannot lead to a better choice of an attribute than the current best choice, then we can leave this attribute unevaluated (at this point).

The above calculated minimum value for ACE serves as the basis for an upper bound of the highest obtainable value of the *information gain* function [18]. It is defined as $IG(\biguplus_i S_i) = H(S) - ACE(\biguplus_i S_i)$. $H(S)$—the entropy of the data

set S prior to partitioning it—is constant with respect to the dimensions of the data. Therefore, IG's maximum value coincides with ACE's minimum value and its relevance assignment can, by the same rationale, be bound by the ratio B/n.

Many other evaluation functions use IG as their building block, which means that from the above analysis of ACE we can obtain bounds for the values of these functions as well. Such functions include, e.g., *balanced gain* [8,14], *gain ratio* [19,20], and *normalized distance* measure [15]. Also, the *gini index* (of diversity) [2] has a very similar formula as IG, and ought to be easy to analyze. In this paper we, however, only consider balanced gain, $BG_{\log}$, which is defined as $BG_{\log}(\biguplus_{i=1}^{k} S_i) = IG(\biguplus_{i=1}^{k} S_i)/\log_2 k$. It has turned out to be a function with, in most cases, superior performance than information gain and gain ratio functions. In addition, it has other desirable properties [8].

4 Utilizing information from preprocessing

No matter which partitioning strategy is used to handle numerical attributes, preprocessing of the data is required. At least the examples have to be sorted. Identification of candidate cut points requires a scan over the data set. Hence, the direct approximation of attribute relevance on the basis of the number of boundary points presented in the preceding section requires time that has a linear dependency on the number of examples n. However, from the preprocessing stage we can also extract, at the low cost of $O(mB)$, the class distributions of blocks. In practice, this preprocessing time has been observed to be negligible with respect to the time required by actual evaluation of candidate partitions [8]. These distributions give another possibility to bound (sometimes more tightly) the relevance of an attribute on the basis of boundary points.

For the function ACE it is quite easy to show—using basic information theoretical results—that its optimal (least) value is obtained by the partition that is defined by all the boundary points of the data.

Theorem 3 (Log Sum Inequality [6]). *Given non-negative $a_i, b_i, i = 1, \ldots, k$,*

$$\sum_{i=1}^{k} a_i \log(a_i/b_i) \geq \left(\sum_{i=1}^{k} a_i\right) \log\left(\sum_{i=1}^{k} a_i / \sum_{i=1}^{k} b_i\right)$$

with equality iff a_i/b_i is constant, $i = 1, \ldots, k$.

Let us substitute into the Log Sum Inequality the non-negative fractions $a_i = n_{i,j}/n$ and $b_i = n_i/n$, where $0 \leq n_{i,j} \leq n_i \leq n$, $i = 1, \ldots, k$; we get

$$\sum_{i=1}^{k} (n_{i,j}/n) \log(n_{i,j}/n_i) \geq (n_j/n) \log(n_j/n).$$

Negating both sides and summing over $j = 1, \ldots, m$ we get

$$-\sum_{i=1}^{k} (n_i/n) \sum_{j=1}^{m} (n_{i,j}/n_i) \log(n_{i,j}/n_i) \leq -\sum_{j=1}^{m} (n_j/n) \log(n_j/n).$$

Bringing the notation in accord with the earlier one, we have $n = |S|$, $n_i = |S_i|$, $n_j/n = P(C_j, S)$, and $n_{i,j}/n_i = P(C_j, S_i)$, which maintain the non-negativity of a_i and b_i. Taking, furthermore, the logarithms to have base 2, the above inequality can be rewritten as

$(1/|S|)\sum_i |S_i|H(S_i) \leq H(S) \Leftrightarrow ACE(\biguplus_i S_i) \leq ACE(S)$.

In other words, any partition $\biguplus_i S_i$, $i = 2, \ldots, B$, of a data set S will have at most the same average class entropy as the whole data set.

ACE is convex in between any two consecutive boundary points [5, 9] and any further partitioning of the data on a boundary point reduces the average class entropy of the partition. Hence, the minimum ACE value over a data set is always obtained by the B-partition that has as its subsets all the blocks of the data. Let us denote the value of ACE in such a case by $\sigma_B = (1/n)\sum_{i=1}^{B} |S_i|H(S_i)$. The value of this partition serves as an approximation of a numerical attribute's utility in class prediction: $ACE(\biguplus_i S_i) \geq \sigma_B$, for any partition $\biguplus_i S_i$, $i = 2, \ldots, B$, of the data set S. Clearly, this lower bound can be computed in linear time.

The value of $H(S)$ can, of course, be computed at the same single pass through the data and it is constant for all attributes. $H(S) - \sigma_B$ is a lower bound for information gain of any partition of S. Incidentally, this explains why the information gain function is so eager to favor higher arity partitions of numerical attribute domains and nominal attributes with many potential values [19]. Furthermore, we can use this value to obtain an upper bound for the balanced gain. Observe that $BG_{\log}$ does not (necessarily) obtain its maximum value when all blocks of the data constitute a partition subset of their own since the denominator $\log_2 k$ biases against unnecessary splitting.

It is common to set an upper bound k for the arity of the partition. Obviously, the above-derived approximations are not very tight if $k \ll B$. We cannot use partitions of arity k as our approximation, since enumerating them requires $O(B^2)$ time.

5 Empirical evaluation

This section presents the results of comparative experiments in which C4.5 algorithm [20] changed to multisplit numerical attributes optimally using the balanced gain function and equipped with four different postponing strategies:

- **Analytic**. We combine the two analytically derived bounds and compare the best observed $BG_{\log}$ score with the value $(H(S) - \sigma_{\max})/\log_2 k$, where $\sigma_{\max} = \max\{(B - \ell)/n, \sigma_B\}$.
- **Heuristic1**. This heuristic postpones the evaluation of an attribute if $B/n > t$. As threshold t we try values .5, .2, and .1.
- **Heuristic2**. This heuristic orders the numerical attributes by the number of boundary points and postpones the evaluation $(1 - t)100\%$ of them, those that have the highest number of boundary points. We test values $t = \{.9, .7, .5\}$.
- **Heuristic3**. The final heuristic postpones the evaluation of those numerical attributes that have $B_{\min}/B > t$, where $B_{\min}$ is the least boundary point count among the attributes. Threshold values .9, .7, and .5 are attempted.

Into our comparison we have chosen 15 data sets mainly from the UCI repository with such properties that they contain numerical attributes, have attributes

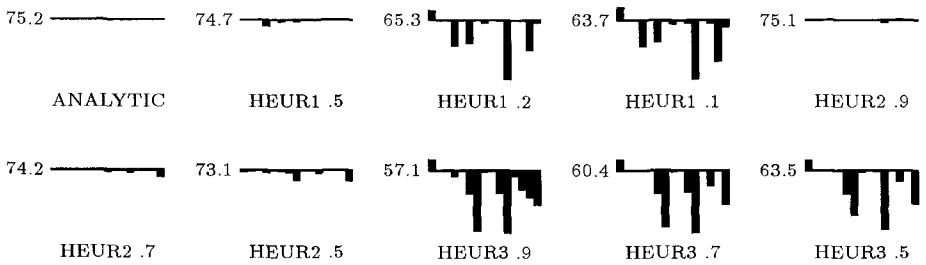

Fig. 1. The average accuracies of the postponing strategies in the 5x2cv test. The average accuracy of not postponing the evaluation of any attributes is 75.2%. The bars represent the relative gain or loss with respect to not postponing attribute evaluation.

with a high number of boundary points, or are large. Most of the domains are well-known; we do not describe them here, for a comprehensive description of their characteristic figures see, e.g., [8]. The domains are Abalone, Adult, Australian, Auto imports, German, Glass, Letter recognition, Mole, Page blocks, Satellite, Segmentation, Shuttle, Vehicle, Waveform, and Yeast.

As the test strategy we use two-fold cross validation testing repeated five times, 5x2cv; it has been observed to be a reliable statistical test in experiments that involve comparison of more than two learning algorithms [7].

The average prediction accuracies obtained using the strategies in the 5x2cv test are depicted in Fig. 1. The most salient observation to be made from these results is that we cannot claim Heuristic3 nor Heuristic1 with thresholds .2 and .1 to maintain the overall level of prediction accuracy that exists when the evaluation of attributes is not delayed. Heuristic2, on the other hand, maintains the overall accuracy even when 50% of attributes are left unevaluated at each attribute selection step. The strategy Analytic does not change the prediction accuracy significantly but, as can be observed from the representation in Fig. 2, that is mainly due to it not postponing the evaluation of attributes near the root level of the tree; only when the number of remaining boundary points approaches that of the partition arity limit, the analytical bounds start to have an effect. The analytically derived bounds are not tight enough to gain speed-up in practice.

The utility of the heuristic methods is ultimately decided on the time saving that is obtained through using them. In particular, on the domains that contain singular malignant attributes that cause the optimal multisplitting algorithm to use excessive amounts of time. The reference time is that of not postponing the evaluation of attributes. The overall performance is summarized by the *geometric mean* of these results.

Fig. 2 shows the average time consumptions of the postponing strategies. We can observe that Heuristic1 with threshold .5—which still maintains the overall prediction accuracy well—cannot bring time savings, except for one domain: Abalone. It is, however, important to notice that for all time critical domains, except Waveform, the tighter thresholds maintain (or even increase) accuracy and bring speed-ups; they are substantial whenever there are individual malig-

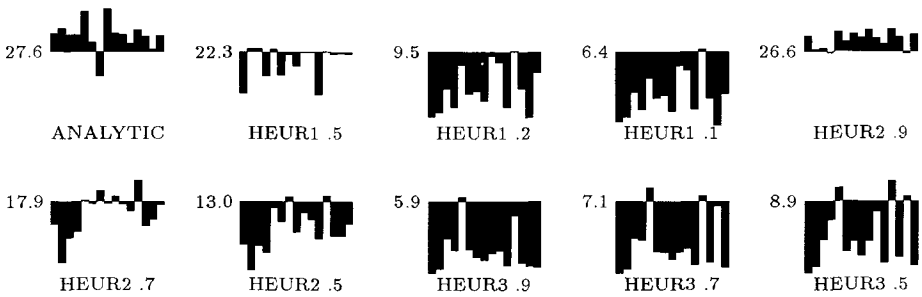

Fig. 2. The geometric mean times of the postponing strategies in the 5x2cv test. The mean time required when not postponing the evaluation of attributes is 23.0 seconds. The bars represent relative speed-up or slow-down on individual domains.

nant attributes in the domain—e.g., Abalone and Adult—but less impressive in other cases—e.g., Page blocks and Shuttle.

From Fig. 2 we can see that the speed-up of Heuristic2 depends on the strictness of the threshold: with parameter value .9 no time savings are obtained, but the lower values bring better results. Again the best results are obtained for the most critical domains. A small accuracy-efficiency tradeoff exists also for this heuristic (cf. Fig. 1). Heuristic3 gains a lot of speed for the decision tree construction, but—with these threshold values—the loss of accuracy is intolerable.

Altogether, all three heuristics do well in getting rid of singular malignant attributes, which are not useful in induction in any case. The achieved speed-up depends on the domain and on the strictness of the threshold. Unfortunately, in other cases the heuristics can work against the accuracy of the result by postponing the evaluation of an important attribute, forcing the learning algorithm to make a less profitable choise. Heuristic 2 appears very safe in this respect.

6 Conclusions and further work

We presented an analysis on the relationship of a numerical attribute's relevance to class prediction and the number of boundary points in the data dimension determined by the attribute. The analytic bounds are not tight enough to screen out malignant attributes, but suggest efficient heuristics that can be used to enhance univariate decision tree induction by postponing the evaluation of attributes that are very likely to have a low relevance and would require substantial amount of time for evaluation. The empirical evaluation confirms the benefits that can be obtained in case of removing malignant attributes, but also show that some heuristics can work against the accuracy of the resulting decision tree.

The most obvious direction for further work is to continue the analysis of the multisplitting of numerical attributes to obtain tighter and more practical bounds for the utility of an attribute in class prediction. In case of the bound that utilizes information from the preprocessing, the most urgent need would be to close the gap between the arity of the lower bound, B, and that of the partition under consideration, ℓ. That gap is the reason for this bound's looseness.

Further heuristics that take the number of boundary points into account are easy to figure out, as well as enhancements to the heuristics studied in this paper. For instance, turning off the postponing in case of small domains or when the tree construction has proceeded to a certain stage would both probably enhance the efficiency of the heuristics.

References

1. Blum, A., Langley, P.: Selection of relevant features and examples in machine learning. Artif. Intell. **97** (1997) 245–271
2. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth (1984)
3. Brodley, C., Friedl, M.: Identifying and eliminating mislabeled training instances. In: Proc. 13th Natl. Conf. on Artificial Intelligence. AAAI Press (1996) 799–805
4. Caruana, R., Freitag, D.: Greedy attribute selection. In: Machine Learning: Proc. 11th Intl. Conf. Morgan Kaufmann (1994) 28–36
5. Codrington, C., Brodley, C.: On the qualitative behavior of impurity-based splitting rules I: The minima-free property. Tech. Rep. 97-5. Purdue Univ., School of Electrical and Computer Engineering, 1997
6. Cover, T., Thomas, J.: Elements of Information Theory. Wiley & Sons (1991)
7. Dietterich, T.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. (to appear)
8. Elomaa, T., Rousu, J.: General and efficient multisplitting of numerical attributes. Mach. Learn. (to appear)
9. Fayyad, U., Irani, K.: On the handling of continuous-valued attributes in decision tree generation. Mach. Learn. **8** (1992) 87–102
10. Fayyad, U., Irani, K.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proc. 13th Intl. Joint Conf. on Artificial Intelligence. Morgan Kaufmann (1993) 1022–1027
11. Fulton, T., Kasif, S., Salzberg, S.: Efficient algorithms for finding multi-way splits for decision trees. In: Machine Learning: Proc. 12th Intl. Conf. Morgan Kaufmann (1995) 244–251
12. John, G.: Robust decision trees: Removing outliers from data. In: Proc. 1st Intl. Conf. on Knowledge Discovery and Data Mining. AAAI Press (1995) 174–179
13. Kohavi, R., John, G.: Wrappers for feature subset selection. Artif. Intell. **97** (1997) 273–324
14. Kononenko, I., Bratko, I., Roškar, E.: Experiments in automatic learning of medical diagnostic rules. Tech. Rep. Josef Stefan Institute (1984)
15. López de Màntaras, R.: A distance-based attribute selection measure for decision tree induction. Mach. Learn. **6** (1991) 81–92
16. Merz, C., Murphy, P.: UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html.
17. Oates, T., Jensen, D.: Large datasets lead to overly complex models: an explanation and a solution. In: Proc. 4th Intl. Conf. on Knowledge Discovery and Data Mining. AAAI Press (to appear)
18. Quinlan, R.: Learning efficient classification procedures and their application to chess end games. In: Michalski, R., Carbonell, J., Mitchell, T. (eds.): Machine Learning: An Artificial Intelligence Approach. Tioga (1983) 391–411
19. Quinlan, R.: Induction of decision trees. Mach. Learn. **1** (1986) 81–106
20. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)