EWSL-91 (Lecture Notes in AI)

ON ESTIMATING PROBABILITIES IN TREE PRUNING

Bojan Cestnik[1] Ivan Bratko[1,2]

[1] Jožef Stefan Institute, Jamova 39, 61000 Ljubljana, Yugoslavia
E-mail: cestnik@ijs.ac.mail.yu

[2] Faculty of Electrical Eng. and Computer Science
Tržaška 25, 61000 Ljubljana, Yugoslavia

Abstract

In this paper we introduce a new method for decision tree pruning, based on the minimisation of the expected classification error method by Niblett and Bratko. The original Niblett-Bratko pruning algorithm uses Laplace probability estimates. Here we introduce a new, more general Bayesian approach to estimating probabilities which we call *m-probability-estimation*. By varying a parameter m in this method, tree pruning can be adjusted to particular properties of the learning domain, such as level of noise. The resulting pruning method improves on the original Niblett-Bratko pruning in the following respects: apriori probabilities can be incorporated into error estimation, several trees pruned to various degrees can be generated, and the degree of pruning is not affected by the number of classes. These improvements are supported by experimental findings. m-probability-estimation also enables the combination of learning data obtained from various sources.

1 Introduction

The importance of tree pruning and rule simplification in machine learning (ML) from incomplete and unreliable data has been widely acknowledged within the ML community. Several approaches to tree pruning have been developed in the last few years including (Breiman *et al*, 1984) , (Quinlan, 1986, 1987), (Niblett & Bratko, 1986), (Cestnik *et al*, 1987), (Clark & Niblett, 1987), (Smyth *et al*, 1990). The approaches differ in their use of various criteria for deciding whether to prune at a certain stage or not. That is the reason why they are difficult to compare. An attempt to summarise and empirically compare five tree pruning methods was made by Mingers (1989).

In this article we present a new method for tree pruning. It is based on the minimisation of the total expected classification error method by Niblett and Bratko (1986) and improved with a new Bayesian approach for estimating probabilities (Cestnik, 1990). Niblett and Bratko used Laplace's law of succession to estimate the required

probabilities. For a k-class problem, this law states that if in the sample of N trials there were n outcomes of a class c, the probability of c in the next trial is $(n+1)/(N+k)$. This assumes that the apriori probability distribution is uniform and equal for all classes. Laplace's law of succession is only a special case of a more general Bayesian method for estimating probabilities that assumes the beta prior probability distribution (Good, 1965; Berger, 1985). For the k-class problem, after n successes in N trials, the probability p of c in the next trial is, according to this Bayesian method, the following:

$$p = \frac{n + p_a m}{N + m} \tag{1}$$

where p_a is apriori probability of c and m is a parameter of the estimation method.

This formula, developed in (Cestnik, 1990), will be called *m-probability-estimation*, and the resulting probability estimates *m-estimates*. The new method for tree pruning, using m-estimates, inherits the theoretical clarity from the original Niblett-Bratko method. By introducing one additional degree of freedom (parameter m) it alleviates some problems with the Niblett-Bratko method found by Cestnik *et al* (1987) and Mingers (1989). First, instead of assuming a uniform initial distribution of classes, which is seldom true in practice, prior probabilities can be incorporated in the estimation. Second, by varying the parameter m, several trees, pruned to different degrees, can be obtained. And third, the number of classes no longer affects the degree of pruning; the degree of pruning is determined by the choice of m. m can be adjusted to some essential properties of the learning domain, such as the level of noise in the learning data.

In section 2 the approach to generating and pruning trees with m-probability-estimation is presented. Section 3 describes the experiments with the new method. In section 4, the obtained results are discussed and interpreted. The m-probability-estimation method also enables a new method for combining evidence from various sources. This is described in section 5. Finally, in section 6, the conclusions are drawn and some directions for further research are presented.

2 Generating and pruning with m-probability-estimates

The Bayesian approach to estimating a probability p can be summarised in the following way (Good, 1965): first, one has to assume an initial (apriori) probability distribution for p. Then, the evidence E converts this distribution into a final (aposteriori) distribution, from which the expectation and the variance of p can be taken. Usually, a class of initial density functions is given by the beta form, proportional to $p^a(1-p)^{m-a}$, where $a > -1$ and $m-a > -1$. The initial distribution depends on the initial expectation and variance (Good, 1965), and can be practically determined with parameters a and m. In (Cestnik, 1990) it is shown that a and m should satisfy the constraint $a = p_a m$, where p_a is the initial expectation of p. The parameter m is related to the initial variance. In fact,

m determines the impact of the apriori probability p_a to the estimation of p. Further interpretation of parameter m is given at the end of this section.

In general, the need for manual estimation is considered as a basic limitation of the Bayesian approach. In the case when multiple attributes interact, a large number of parameters must be estimated in a subjective manner (Pearl, 1988). On the other hand, in the context of learning rules from data, a modified Bayesian approach can be used. Namely, the probability estimate of a given class in the whole learning set can be used as apriori probability in the estimation of conditional probabilities (given some attributes) of the class. This approach turned out to work very well in the framework of the "naive" Bayes formula (Cestnik, 1990). However, the problem of determining the value of m still remains. For the sake of simplicity we will assume that m is equal for all attributes and their combinations in a given domain. Yet, in practice, this might not be the case; m might depend on a particular attribute or even a particular value of an attribute.

When building classifiers in the form of a decision tree from incomplete and unreliable data (class probability trees) there are three stages in which probability estimates are needed: first, in the tree construction, when selecting an attribute for the root of a subtree according to some criterion (eg. information gain); second, in the tree pruning phase, when the branches with little statistical validity have to be identified and removed; and third, in the classification phase, when, based on the examples in a given leaf, probabilities have to be assigned to the classes. Once we have decided upon the method for estimating apriori probabilities, we have to select a suitable m for each of the three phases.

As a general scheme, we propose $m = 0$ in the tree construction phase, because a constructed full-sized tree is only a new form of representing the learning data set. Therefore, there is no need to incorporate "apriori information" since it may unnecessarily complicate the tree. On the other hand, in the tree pruning phase we want to adjust the constructed tree to the whole domain of learning (i.e. also to previously unseen objects – independent data set). At this stage, we have to consider apriori probabilities; consequently, m has to be greater than 0. The actual value of m depends on the level of noise in a given domain (larger m for more noise). So, m can be proposed by the domain expert. On the other hand, one can use heuristics to set m; for example, m can be set so as to maximise the classification accuracy on an independent data set. A similar approach to determining the parameter α for pruning in *CART* is described in (Breiman *et al*, 1985).

With m-probability-estimates, the "static error" E_s in a given node is given by the following formula:

$$E_s = 1 - \frac{n_c + p_{ac}m}{N + m} = \frac{N - n_c + (1 - p_{ac})m}{N + m} \tag{2}$$

where N is the total number of examples in the node, n_c is the number of examples in class c that minimises E_s for the given m, p_{ac} is the apriori probability of class c, and m is the parameter of the estimation method.

Note that if m is equal to the number of classes and $p_{ac} = 1/m$ then the formula (2) is equivalent to Laplace's law of succession used by Niblett and Bratko.

The backed-up error E_b of a given node is computed considering the error estimates for the subtrees. In the case of a binary tree E_b is computed as follows:

$$E_b = p_1 E_1 + p_2 E_2 \tag{3}$$

where p_i is the probability that an object will fall in the i-th subtree, and E_i is the error estimate for the i-th subtree.

Again, (3) is the combination formula from the Niblett-Bratko approach. However, the difference is in the estimation of p_i. Instead of previously used relative frequency estimate we propose the m-estimates with $m = 2$. It should be admitted that here m is chosen arbitrarily. However, we found out that the results do not depend much on the choice of m in this formula.

The pruning algorithm is presented in (Niblett & Bratko, 1986). It basically states that if, in a given node, the backed-up error E_b is greater than the static error E_s, the tree is pruned at this node.

When adjusting a constructed full-sized tree to the whole domain, the tree is pruned so as to minimise the expected classification error given a certain value of m. The same value of m must then be used also in the classification of a new object with the pruned tree. It might happen that, with changing the value of m, the class that minimises the estimated error for some m is different from the "minimal error" class for another m. A situation like this is presented in Figure 1. There are two classes, '+' and '−'. Their apriori probabilities are: $p_a('+') = 0.4$ and $p_a('-') = 0.6$. Now, suppose that there is a leaf with 10 examples, 7 '+' and 3 '−'. Figure 1 shows that for $m < 20$ the m-estimate probability of '+' is higher than 0.5, when $m = 20$ both estimates are equal, and for $m > 20$ the probability estimate of '−' takes over the lead. As a result, an object, fallen in this leaf, would be classified as '+' if $m < 20$, undecided if $m = 20$, and as '−' if $m > 20$.

To obtain a clearer understanding of the meaning of m, we will consider how m affects the combination of evidence from data and apriori probabilities. The formula (1) can be rewritten in the following way:

$$p = \frac{N}{N + m}\hat{p} + \frac{m}{N + m}p_a \tag{4}$$

where N is the number of examples, $\hat{p}$ is the relative frequency estimation n/N, p_a is apriori probability and m is the parameter of the estimation method.

From (4) it can be seen that if $m = 0$ then p is equal to relative frequency estimate $\hat{p} = n/N$. When $m = N$ (m is equal to the number of examples) p is equal to the average of $\hat{p}$ and apriori probability p_a. In the limit, when m is infinite, p is equal to p_a. So, by varying the parameter m one actually determines the effect of the

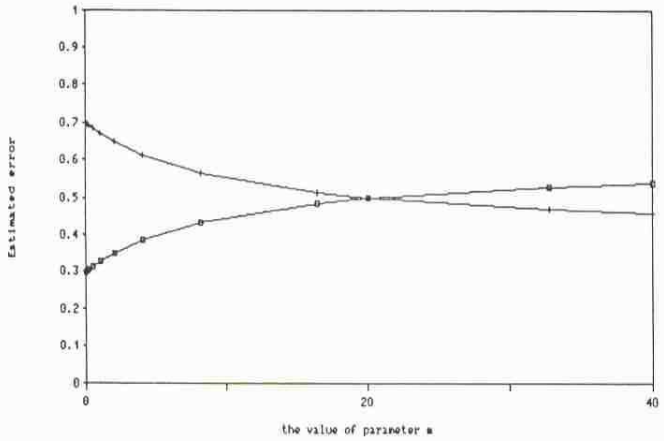

Figure 1: Estimated static errors for the classes '+' and '−' in a leaf with 10 examples (7 '+' and 3 '−'). Apriori probabilities for '+' and '−' are 0.4 and 0.6 respectively. At $m = 20$ the estimated errors for the two classes are equal.

apriori probability and the relative frequency on the final probability estimate. A further interpretation of formula (4) is possible: the formula combines the apriori probability and the new evidence obtained from N examples. The combination is such as if the apriori probability was obtained from m prior examples and estimated with relative frequency.

3 Experiments and empirical findings

We did two sets of experiments: one with artificial, synthetic trees, and one with trees generated from learning data from eight natural domains. We will first describe the experiments with synthetic trees.

To measure the impact of the number of classes, the class distribution and the value of m on the degree of pruning with the new method the following experiment was performed. First, complete binary tree of depth 10 (1024 leaves) was constructed. Second, each leaf was randomly assigned one example from a given class distribution. And third, the tree was pruned according to the value of m. In each experiment the size of the final tree and its accuracy on the "learning" examples (from the original tree) were measured. In experiments we varied the number of classes, the class distribution and parameter m. For each combination of these three variables, 10 experiments were performed to obtain the average values for the above-mentioned measured parameters. We repeated this experiment with complete binary trees of other depths between 8 and 12, and found that the relative degree of pruning (in percents) does not depend on the depth of the tree.

In the experiment we selected the following values for the parameters: The values for the number of classes were 2, 4, 8, 16 and 32. There were three different class distributions: uniform (0), where $p_i = c_0$ for all i; linear (1), where $p_i = p_{i-1} + c_1$ for $i > 1$ and $p_1 = c_1$; and geometric (2), where $p_i = p_{i-1}c_2$ for $i > 2$ and $p_1 = c_2$. The constants c_0, c_1 and c_2 are adjusted so that the sum of p_i over all classes equals to 1 for each distribution. For the value of m values 0, 0.01, 0.5, 1, 2, 3, 4, 8, 12, 16, 32, 64, 128 and 999 were selected.

The results obtained for uniform, linear and geometric class distribution are shown in Figures 2, 3 and 4, respectively. In each figure the points (labeled x) for the same class and different m, are connected with lines. The circle ○ represents the result of pruning with original Niblett-Bratko method. These experimental results will be interpreted in section 4.

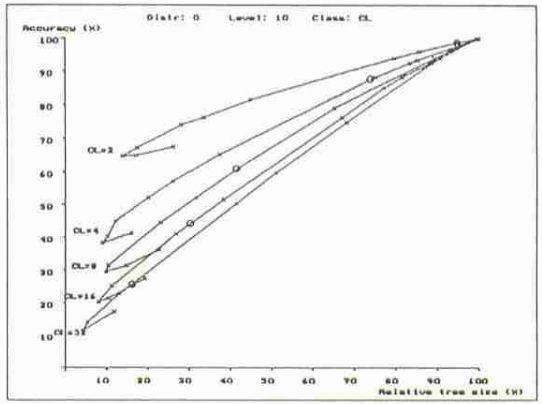

Figure 2: Accuracy vs. size of the tree for uniform distribution of classes

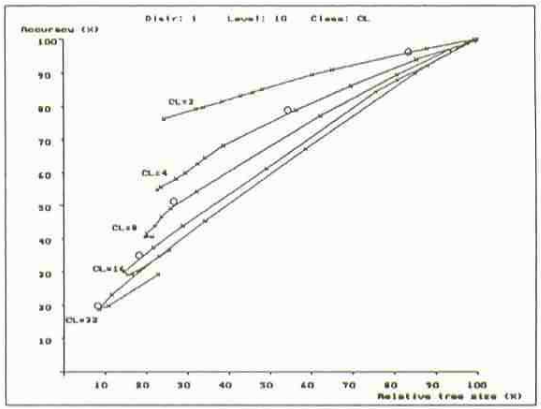

Figure 3: Accuracy vs. size of the tree for linear distribution of classes

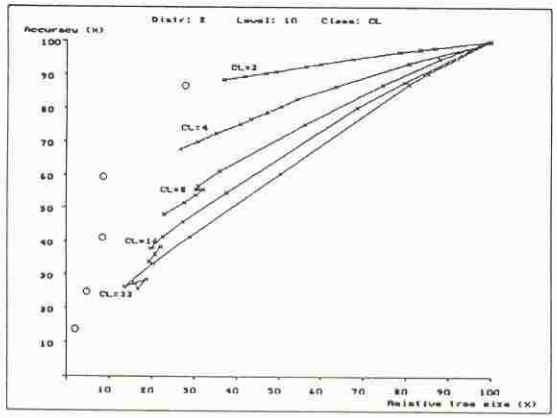

Figure 4: Accuracy vs. size of the tree for geometric distribution of classes

The new method was also tested on eight real-world domains. The domains are described in Table 1. The four rheumatology domains are essentially equal except for the number of classes that are hierarchically decomposed from three to six, eight and finally twelve classes. The first four domains are described in more detail in (Bratko & Kononenko, 1986).

Domain		#clas.	#attr.	#exam.	%maj.class
Hepatitis	HEPA	2	19	155	79.4
Lymphography	LYMP	4	18	148	54.7
Breast cancer	BREA	2	10	288	79.9
Primary tumor	PRIM	22	17	339	24.8
Rheumatology 3	RE03	3	58	462	66.5
Rheumatology 6	RE06	6	58	462	61.9
Rheumatology 8	RE08	8	58	462	34.2
Rheumatology 12	RE12	12	58	462	34.2

Table 1: Description of eight medical domains

The results, presented in Table 2, are averaged over 10 experiments. Each time 70% of examples were randomly taken for learning and the rest 30% for testing. All the methods were tested on the same learning-testing partition.

In Table 2 there are three numbers for each error estimate (m) in each domain. The first number stands for the number of leaves in the tree. The second one shows classification accuracy on the learning data set and the third one classification accuracy on the testing data set. The results are presented graphically in Figures 5 and 6.

	HEPA			LYMP			BREA			PRIM		
Laplace	12.1	89.0	82.1	16.9	98.2	77.1	51.6	88.7	77.8	33.5	31.9	30.6
0.0	13.2	91.3	79.8	17.8	100.0	77.5	63.8	94.0	73.6	90.5	70.5	40.3
0.01	13.1	91.3	79.8	17.8	100.0	77.3	64.3	94.0	74.1	90.3	70.5	40.4
0.5	13.0	90.9	81.5	17.8	100.0	77.3	61.3	94.0	74.3	82.6	69.6	40.1
1	12.7	90.4	81.1	17.8	100.0	77.3	55.5	93.9	74.3	74.3	67.6	40.7
2	12.1	89.0	82.1	17.4	99.0	77.3	51.6	88.7	77.8	61.4	63.8	40.6
3	12.5	88.5	83.6	17.0	98.8	77.3	47.9	88.5	77.9	54.9	61.3	41.4
4	10.8	88.3	84.5	16.9	98.2	77.1	47.4	84.3	78.8	50.2	58.4	40.3
8	10.4	86.4	84.0	15.7	94.6	76.8	41.3	81.7	79.3	39.9	47.0	36.8
12	9.4	85.5	85.5	13.5	90.8	75.9	40.0	80.2	80.0	39.6	41.3	35.1
16	9.4	85.0	85.5	11.8	88.3	75.0	37.7	79.8	80.0	35.6	35.0	31.4
32	9.7	77.4	83.8	7.6	82.6	75.9	29.9	79.8	80.0	37.9	30.5	29.0
64	9.7	77.4	83.8	7.4	81.1	74.8	26.3	79.8	80.0	44.4	26.4	25.0
128	9.4	77.4	83.8	7.8	74.8	69.1	22.2	79.8	80.0	47.5	24.8	24.6
999	9.4	77.4	83.8	9.1	54.1	56.1	21.6	79.8	80.0	46.4	24.8	24.6
m	RE03			RE06			RE08			RE12		
Laplace	40.4	79.5	70.0	46.3	66.8	64.5	49.1	64.4	47.7	44.2	58.0	47.6
0.0	60.4	86.2	67.6	70.2	80.3	56.8	87.0	81.7	47.0	86.3	78.0	45.6
0.01	59.8	86.1	67.8	70.2	80.3	56.8	87.0	81.7	46.6	86.3	78.0	45.6
0.5	52.5	84.5	68.1	66.0	79.6	57.2	85.2	81.3	46.3	85.4	77.2	46.0
1	49.4	83.5	69.2	63.2	78.6	58.5	83.0	79.9	46.2	81.5	76.4	45.6
2	44.1	82.5	69.7	56.5	76.5	60.4	74.9	77.2	46.2	76.7	74.5	46.0
3	40.4	79.5	70.0	50.4	72.9	62.0	67.2	75.0	46.6	71.8	73.0	46.5
4	36.7	77.4	70.3	44.9	70.2	63.0	61.8	72.2	47.6	64.4	70.0	47.7
8	32.5	76.4	71.4	46.1	65.8	64.0	49.1	64.4	47.7	48.3	60.7	47.9
12	31.7	73.8	70.8	43.6	64.3	63.0	42.4	60.0	48.4	44.2	58.0	47.6
16	32.3	71.0	69.8	43.8	62.6	63.5	41.1	58.1	48.9	41.8	56.0	47.8
32	33.0	69.0	68.9	40.7	61.2	63.5	44.1	52.0	46.3	40.4	50.3	45.1
64	32.5	66.0	67.5	34.8	61.2	63.5	40.3	46.6	43.5	42.2	46.1	43.6
128	30.2	66.0	67.5	35.1	61.2	63.5	39.2	44.3	42.9	40.8	43.9	43.5
999	28.3	66.0	67.5	35.1	61.2	63.5	36.0	33.5	35.9	35.8	33.5	35.9

Table 2: Results of pruning with the Laplace's estimates and with m-estimates for different values of m. The three numbers are the number of leaves in the tree, classification accuracy on the training set and classification accuracy on the independent testing set, respectively.

4 Phenomena and interpretation

The original Niblett-Bratko pruning method with Laplace's error estimate was empirically observed to occasionally over-prune (Cestnik *et al*, 1987) or under-prune (Mingers, 1989). Results from the previous section (see Figures 2, 3 and 4) explain the effect of over-pruning when apriori probabilities of classes are unequal (linear or geometric) in combination with high number of classes. In those cases Laplace's estimates prune too much, much more than the m-error-estimates with properly selected m. Similarly, the effect of under-pruning can be observed in the cases with uniform class distribution and small number of classes. In general, we view m as a domain parameter, and believe that it should be set so as to correspond to the amount of noise in the domain.

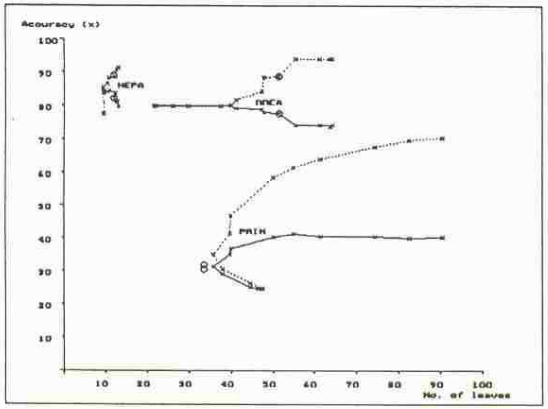

Figure 5: Results from Table 2 for domains HEPA, BREA and PRIM, presented in a graphical form

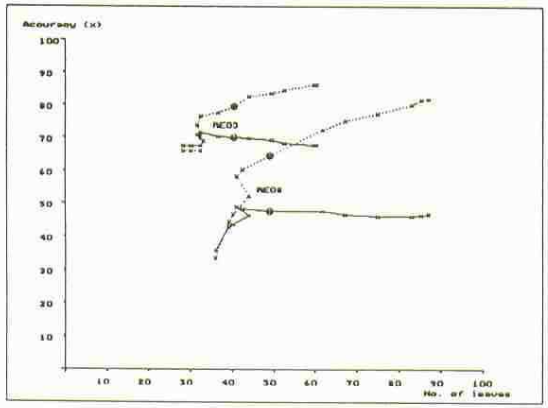

Figure 6: Results from Table 2 for domains RE03 and RE08, presented in a graphical form

The ideal pruning procedure should, when m approaches infinity, prune maximally with only the root left (because all estimated probabilities would then be equal to the prior probabilities). However, the computed error estimates are only almost equal to the apriori probabilities and the pruning criterion $E_s \leq E_b$ is satisfied in about 50% of cases. Propagating this criterion upwards leaves about half of the tree unpruned. So, with very large m, only about half of a binary tree is pruned due to the random statistical behavior of the criterion $E_s \leq E_b$. This explains the irregularities at the left ends of some of the curves in Figures 2–4. It also suggests an obvious improvement that would prevent this anomalous behaviour: the pruning criterion should be "stabilised" by changing it to $E_s \leq E_b + \varepsilon$ where ε is some small positive number.

Mingers in his study (1989) argues that various trees, pruned to different degrees, would be desirable in practice in knowledge acquisition for expert systems. This was considered as a drawback of the Niblett-Bratko method producing one tree only. As an improvement in this respect, the sequence of trees of different sizes can now be generated by varying the value of parameter m.

Mingers also found that in Laplace's error estimation a change in the number of classes drastically affects the degree of pruning. This paper quantitatively supports this conclusion. However, with properly setting the parameter m the degree of pruning becomes stable. Our results also show the effect of prior class probabilities to the degree of pruning. Regarding the observed correlation between the degree of pruning in the Niblett-Bratko method and the number of classes, we believe the following. One should expect more pruning when there are more classes. This can be simply explained by the fact that with more classes and equally sized learning set the majority class estimates become less reliable and therefore more pruning is warranted. However, the degree of pruning with Laplace's estimate seems to exaggerate in this respect, and it also wildly varies with different apriori probability distributions for classes, which is not appropriate.

In some machine learning applications, some learning examples are designed by the expert simply to introduce known facts about the domain. This is typical of some engineering applications. How can such "absolute examples" be handled in this framework? In such situations we want to make sure that one example of class c is enough to estimate $p_c = 1$ in the corresponding leaf. This can be handled by setting m to 0. In a way, this setting ($m = 0$) corresponds to perfect data (no noise, no uncertainty, and complete information, i.e. sufficient set of attributes and examples to determine the class completely). On the other hand, large m would correspond to very uncertain, noisy data.

The results in eight real-world domains show that the reality itself is very complicated. Usually, the distribution of classes is neither uniform, nor any other simple function. In addition, it seems that the optimal value for m critically depends on noise (incomplete and unreliable data, missing data, etc.). The number of classes that is implicitly taken for m in Laplace's estimate can only be viewed as an unsuccessful guess. For example, the best m for Hepatitis (2 classes) is 12, the best m for Lymphography (4 classes) is 0, and the best m for Primary tumor (22 classes) is 3 (see Table 2).

5 Combination of various sources of evidence

The formula (4), that was used to interpret the parameter m, represents a way of combining the evidence from data (relative frequency estimation) and prior belief (apriori probability). Now, the question arises whether data from various sources can be combined using the same principle. For example, in a medical domain we may have some examples from patient histories (unreliable, incomplete, noisy) and some special cases constructed by an expert and supplied as absolute truth. Such situations can be handled according to the following generalisation of formula (4):

$$p = \sum_i \frac{w_i N_i}{\sum_j w_j N_j} p_i \qquad (5)$$

where p_i is the probability from i-th source, N_i is the number of examples from i-th source, and w_i is the weight of i-th source.

In the above mentioned medical case we might have three sources of information: apriori probability, patient histories and expert supplied examples. Suppose that we have 100 patient histories and 10 expert supplied cases. From the patient histories we estimate relative frequency probability p_1 and from expert supplied cases probability p_2. Suppose that the weight of expert cases is 10 and the weight of patient histories is 1. For the apriori probability p_a we set m $(= N_3)$ to 5. In such a case we can compute the combined probability estimate in the following way:

$$p = \frac{1 \times 100}{205} p_1 + \frac{10 \times 10}{205} p_2 + \frac{5}{205} p_a$$

Note that the total weight of expert supplied probability is equal to the weight of the probability based on patient histories, in spite of the different number of examples in both cases.

6 Conclusion

In this paper we presented a new method for minimal-error pruning of decision trees that is based on the Niblett and Bratko (1986) method and modified with the m-probability-estimation. Our main aim was to upgrade the above-mentioned approach so as to preserve the desirable theoretical properties and to alleviate some shortcomings observed by Cestnik *et al* (1987) and Mingers (1989).

The main improvements with respect to the original Niblett-Bratko method are the incorporation of apriori probabilities in the error estimation and a new degree of freedom represented by a choice of parameter m. The advantages, resulting from these two improvements, are the following: first, instead of assuming the uniform initial distribution of classes, which is seldom true in practice, prior probabilities can be incorporated in the error estimation. Second, by varying the parameter m, several trees, pruned to different degrees, can be obtained, which is useful in mechanised knowledge acquisition. And third, the number of classes no longer affects the degree of pruning; the degree of pruning is determined by the choice of m. Experimental results in synthetic and natural domains, presented in section 3, confirm the contributions of the new approach.

The proposed method for estimating probabilities (Cestnik, 1990) can be used at various stages: tree construction, tree pruning and classification. We propose a general scheme regarding the choice of m: generate a tree with $m = 0$, prune with $m' > 0$ and classify with $m' > 0$. The scheme is explained and justified in section 2. The fine adjustment of m for pruning depends on the properties of the domain and degree of noise, and can be left to the domain expert.

The m-probability-estimation method also enables a new method for combining evidence from various sources which was presented in section 5.

For further work, we plan to design a method to automatically select the "best" m. There are various possibilities how it can be done. First, the selection can be based on an independent data set (to measure classification accuracy). The shortcoming of this is that it requires a separate data set. Next, the selection can be based on heuristics taking into account properties of the learning set (observing the slope of classification accuracy with respect to the size of a tree). Last but not least, domain expert can supply important information regarding the choice of m, such as: how much noise there is in the data, how truthworthy particular sources of data are, etc.

Acknowledgments

The data for lymphographic investigation, primary tumor, breast cancer recurrence and rheumatology were obtained from the University Medical Center in Ljubljana. We would like to thank M.Zwitter, M.Soklic and V.Pirnat for providing the data. We thank G.Gong from Carnegie-Mellon University for providing the data for hepatitis problem. Long-term efforts of I.Kononenko and his colleagues were significant in turning these data into easy-to-use experimental material. This work was financially supported by the Research Council of Slovenia and carried out as part of European project ESPRIT II Basic Research Action No. 3059, Project ECOLES.

References

Berger, J.O. (1985), *Statistical Decision Theory and Bayesian Analysis*, Springer-Verlag, New York.

Bratko, I., Kononenko, I. (1986), Learning diagnostic rules from incomplete and noisy data, AI Methods in Statistics, UNICOM Seminar, London, December 1986. Also in *Interactions in AI and Statistics* (ed. B.Phelps) London; Gower Technical Press, 1987.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J. (1984), *Classification and Regression Trees*, Belmont, California: Wadsworth Int. Group.

Cestnik, B., Kononenko, I., Bratko, I. (1987), ASSISTANT 86: A Knowledge-Elicitation Tool for Sophisticated Users, *Progress in Machine Learning*, Eds. I.Bratko & N.Lavrac, Sigma Press, Wilmslow.

Cestnik, B. (1990), Estimating Probabilities: A Crucial Task in Machine Learning. In *Proceedings of ECAI 90*, Stockholm, August 1990.

Clark, P., Niblett, T. (1987), Induction in Noisy Domains, *Progress in Machine Learning*, Eds. I.Bratko & N.Lavrac, Sigma Press, Wilmslow.

Good, I.J. (1965), *The Estimation of Probabilities*, M.I.T. Press, Cambridge, Massachusetts.

Mingers, J. (1989), An Empirical Comparison of Pruning Methods for Decision Tree Induction, *Machine Learning* vol. 4, no. 2, Kluwer Academic Publishers.

Niblett, T., Bratko, I. (1986), Learning decision rules in noisy domains, Expert Systems 86, Cambridge University Press (*Proceddings of Expert Systems 86 Conf.*, Brighton 1986).

Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann: San Mateo, CA.

Quinlan, J.R. (1986), Learning from noisy data, *Machine Learning* vol. 2, Eds. R.Michalski, J.Carbonell and T.Mitchel, Palo Alto, CA: Tioga.

Quinlan, J.R. (1987), Simplifying decision trees, *International Journal of Man-Machine Studies*, 27, pp. 221–234.

Smyth, P., Goodman, R.M., Higgins, C. (1990), A Hybrid Rule-based/Bayesian Classifier, In *Proceedings of ECAI 90*, Stockholm, August 1990.

Rule Induction with CN2: Some Recent Improvements

Peter Clark Robin Boswell
The Turing Institute, 36 N.Hanover St., Glasgow
email: {pete,robin}@turing.ac.uk

Abstract

The CN2 algorithm induces an ordered list of classification rules from examples using entropy as its search heuristic. In this short paper, we describe two improvements to this algorithm. Firstly, we present the use of the Laplacian error estimate as an alternative evaluation function and secondly, we show how unordered as well as ordered rules can be generated. We experimentally demonstrate significantly improved performances resulting from these changes, thus enhancing the usefulness of CN2 as an inductive tool. Comparisons with Quinlan's C4.5 are also made.

Keywords: learning, rule induction, CN2, Laplace, noise

1 Introduction

Rule induction from examples has established itself as a basic component of many machine learning systems, and has been the first ML technology to deliver commercially successful applications (eg. the systems GASOIL [Slocombe et al., 1986], BMT [Hayes-Michie, 1990], and in process control [Leech, 1986]). The continuing development of inductive techniques is thus valuable to pursue.

CN2 is an algorithm designed to induce 'if...then...' rules in domains where there might be noise. The algorithm is described in [Clark and Niblett, 1989] and [Clark and Niblett, 1987], and is summarised in this paper. The original algorithm used entropy as its search heuristic, and was only able to generate an ordered list of rules. In this paper, we demonstrate how using the Laplacian error estimate as a heuristic significantly improves the algorithm's performance, and describe how the algorithm can also be used to generate unordered rules. These improvements are important as they enhance the accuracy and scope of applicability of the algorithm.