End User Friendly Data Mining with Decision Trees – a Reality or a Wish?

P. POVALEJ, P. KOKOL
Laboratory for system design
Faculty for electrical engineering and computer science
University of Maribor
Smetanova ulica 17, 2000 Maribor
SLOVENIA

Abstract: - The main focus of data mining is to present hidden knowledge located in large amount of data in human understandable form. Therefore the knowledge representation has to be simple and easy to interpret, possibly without the computer. Decision trees are one of the most transparent methods often used in data mining, but can we make them user friendly? In the process of decision tree induction a lot of input parameters have to be fine-tuned in order to obtain good results. To brain the right combination of input parameters for a specific problem is a hard task usually performed by data mining expert. So, to make decision tree based data mining end user friendly we explored various alternatives of decision tree induction, concentrating on purity measures. We introduced new hybrid purity measures and tested their adequacy on real world databases. Additionally we constructed a meta decision tree to determine the best combination of input parameters.

Key-Words: - knowledge discovery, decision trees, purity measures, hybrid approach

1 Introduction

Focus on end user is the most important new trend in data mining. Web analytics, customer behavior analysis, customer relationship management, decision making support in health care all reflect a new trend – solutions to problems increasingly embed data mining technology. Hence, data mining applications are increasingly developed and designed specifically for end users and not any more for data mining experts. Thereafter transparency and user friendliness are two of the main challenges for the wider use of data mining and growth of the market and technology development.

Decision trees are one of the most transparent methods in data mining – but can we make them user friendly? The answer seems obvious at the first glance – they are very simple and easy to interpret, but the real obstacle for user friendliness lies in the fact that they are not easy to induce. Not because the induction algorithms are complex, but because so many input parameters have to be fine-tuned to obtain good results. These parameters include for example type of purity measure, level of tolerance, method of discretization, pruning methods not to mention more exotic improvements like bagging or boosting. To obtain the right combination of input parameters settings for a specific problem is a hard task even for data mining expert but almost impossible for the end user. So to make decision tree based data mining end user friendly we should to invent a parameter free decision tree induction process or at least find some rules in parameter settings.

In this paper we explored various alternatives to make this possible, concentrating first on purity measures and boosting algorithm. We tried to find if there is some metric or boosting algorithm, which works best in the majority of problems, then we tried to combine various approaches and finally we built a meta decision tree to automatically find the best combination of input parameters.

Some studies concerning impact of impurity measures for attribute selection and other parameters to decision tree induction have already been made. In [1] the authors concentrate on two well known measures gini index and information gain ratio on a single synthetic dataset. The most recent study of impact of different impurity measures is presented [2] and is mainly focused on the improvements of look-Ahead criteria. The paper is organized as follows. In section 2 we describe a greedy decision tree induction algorithm and the most frequently used classic purity measures. Since it is hard to evaluate which purity measure selects the most appropriate attributes in general, we compared the effectiveness of different purity measures on 56 UCI (University of California Irvine) databases [3] from MLC++.

The next section (3) briefly introduces boosting – a method for generating an ensemble of classifiers by successive reweightings of the training cases. AdaBoost algorithm introduced by Freund and Schapire [4] is used for boosting decision trees induced on the basis of different purity measures.
New hybrid purity measures are introduced in section 4 and tested on all databases used in previous experiments. In the final section experimental results and observations are discussed and the results of our attempt in building a meta decision tree are presented. Paper concludes with some general comments and outlines directions for further research.

2 Greedy decision tree induction method and purity measures

A decision tree is constructed from a training set which consists of training objects (cases). Every object is described by a set of attributes and a class label. The values of the attributes can be nominal or discrete, but all nominal attributes have to be mapped into discrete space.
A decision tree contains zero or more inner nodes and one or more leaf nodes. Every inner node represents a test of a value of a specific attribute and therefore splits the dataset into different subsets. All inner nodes have two or more child nodes. Edges from inner node to child nodes are labeled with different outcomes of the test at inner node. Each leaf node has a class label associated with it. Greedy top-down decision tree induction is a commonly used method for tree growing. Starting with an empty tree and the entire training set the following algorithm is applied until no more splits are possible:

A greedy decision tree induction algorithm:

1. If all training examples at the current node t belong to the same class c, create a leaf node with a class c.
2. Otherwise, for each attribute compute its purity with a respect to the class attribute using a goodness measure.
3. Select the attribute (say A_i) with the highest purity gain with a respect to the discretization as the test at the current node.
4. Divide the training samples into separate sets, so that within a set all objects have the same value of A_i using selected discretization. Create as many child nodes as there are distinct values of A_i.
5. Label edges between the parent and the child nodes with outcomes of A_i and partition the training samples into the child nodes.
6. A child node is said to be "pure" if all the training samples at the node belong to the same class.
7. Repeat the previous steps on all impure child nodes.

A new case is thus dropped down the decision tree from the root of the tree to a single leaf node. As a consequence, the instance space is partitioned into mutually exclusive and exhaustive regions, one for each leaf. The number of leafs is therefore a good measure for complexity of the decision tree. A quality of induced decision tree is then tested on unseen testing cases and described with total and class accuracy.

2.1 Purity measures

As described before the problem in decision tree induction process is to select the attribute that is the most successful in discriminating the input data with the respect to a class attribute. All purity measures are defined in a way that the best attribute maximizes the measure. In this section four most popular purity measures used in our experiments are described. Before we proceed to description of purity measures let us define the terminology.
Let S be the whole training set described with A attributes and C classes. Let V be the number of values of a given attribute respectively. Let $n_{..}$ denote the number of training instances, $n_{i.}$ the number of training instances from class C_i, $n_{.j}$ the number of instances with j-th value of a given attribute and n_{ij} the number of instances from class C_i with a j-th value of a given attribute. Let further $p_{ij} = n_{ij}/n_{..}$, $p_{i.} = n_{i.}/n_{..}$, $p_{.j} = n_{.j}/n_{..}$, and $p_{i|j} = n_{ij}/n_{.j}$ denote the probabilities from the training set.

2.1.1 Information-gain, information-gain ratio

One of the oldest and most commonly used purity measures, which was used already in Quinlan's ID3 algorithm [5], is the information gain. It is based on Shannon's entropy from information theory [6], which has its origins in thermodynamics and statistical physics. In the latter entropy represents the degree of disorder in a substance or system. Similarly, entropy in information theory measures the uncertainty of a message as an information source. The more information contains the message, the smaller the value of the entropy.
Let E_C, E_A, E_{CA} denote the entropy of class distribution, the entropy of the values of a given attribute and the entropy of the joint distribution class - attribute value:

$$E_C = -\sum_i p_{i.} \log_2 p_{i.} \qquad E_A = -\sum_j p_{.j} \log_2 p_{.j}$$

$$E_{CA} = -\sum_i \sum_j p_{ij} \log_2 p_{ij}$$

The expected entropy of the class distribution with regard to attribute A is defined as $E_{C|A} = E_{CA} - E_A$.
When compared to the entropy E_C of the class distribution, the $E_{C|A}$ gives the reduction of the entropy (the gain of information) to be expected

when the attribute A is selected for the split. Hence the information gain I_{gain} is defined as $I_{gain}(A) = E_C - E_{C|A}$.

Therefore, in every inner node attribute that yields the highest value of I_{gain} is selected for the split.

As information gain shows a strong bias towards the multi-valued attributes, Quinlan [7] introduced the information gain ratio in C4.5, which is defined as $I_{gainratio}(A) = \frac{I_{gain}(A)}{E_A}$.

Dividing the information gain by the entropy of the attribute value distribution strongly reduces the bias towards the multi-valued attributes [8, 9].

2.1.1 Gini index

Another well-known purity measure is Gini index that has been also used for tree induction in statistics by Breiman et al. [10] (i.e. CART). It is defined as: $Gini(A) = -\sum_j p_{.j} \sum_i p_{i|j}^2 - \sum_i p_{i.}^2$

The attribute yielding the highest value of Gini index is selected for the split. It emphasizes equal sized offspring and purity of both children. Breiman et al. also pointed out that Gini index has difficulties when the class attribute has relatively large number of classes.

2.1.2 Chi-square goodness-of-fit

Different types of chi-square tests [11] are frequently used for significance testing in statistics.

The chi-square goodness-of-fit test is used to test if an observed distribution conforms to any other distribution, such as one based on theory (exp. normal distribution) or one based on some known distribution.

It compares the expected frequency e_{ij} with the observed frequency n_{ij} of instances from class C_i with a j-th value of a given attribute [12]. More specifically,

$$\chi^2(A) = \sum_i \sum_j \frac{(e_{ij} - n_{ij})^2}{e_{ij}} \quad e_{ij} = \frac{n_{.j} n_{i.}}{n_{..}} \quad .$$

Clearly a larger value of χ^2 indicates that the split is more pure. Just as for information gain and gini index the attribute with the highest value of χ^2 is selected for the split.

2.1.3 J-measure

J-measure was introduced by Smyth and Goodman [13] as an informatics theoretic means of quantifying the information content of the rule.

The J_j-measure or cross-entropy is appropriate for selecting a single attribute value of a give attribute A for rule generation and it is defined by the equation $J_j(A) = p_{.j} \sum_i p_{i|j} \log \frac{p_{i|j}}{p_{i.}}$.

Generalization upon all values of the attribute gives the attribute purity measure $J(A) = \sum_j J_j(A)$.

J-measure was also used as a basis for reducing overfitting by pre-pruning branches during decision tree induction [15].

2.2 Experiments and results

For comparing the quality of different purity measures, 56 UCI (University of California Irvine) databases were used. A training set was used for greedy decision tree induction based on a specific purity measure. A quality of induced decision trees was evaluated on testing set.

In the process of decision tree induction all attributes are tested in each node using available training instances. The winning combination of attribute and discretization method is selected using chosen purity measure and applied as a test in the decision tree node.

Hence, four different decision trees were induced for every database, each based on different purity measure. In order to make a fair comparison, the decision trees were not pruned at all. That most certainly leads to overfitting and consequently worse performance on the testing set. The quality of induced decision trees was described with total accuracies on testing set, presented in Table 1.

When comparing the accuracies of decision trees induced on the basis of different purity measures it can be seen that the decision trees based on chi-square and J-measure did not reach 100% accuracy on learning sets for the most of the databases. The results on the testing sets show that none of the used purity measures outperforms others in general.

Therefore our next attempt was to investigate the influence of boosting algorithm on induced decision trees with a respect to different purity measures.

3 Boosting

Boosting is a general method for improving the accuracy of any given learning algorithm. It works by running the learning algorithm on the training set multiple times, each time focusing the learner's attention on the difficult cases. At every iteration a classifier is built from the weighted training cases and each case is then reweighted according to the accuracy of classification by present classifier.

Table 1: Average accuracy on testing set

Data	ID3	Chi-square	Gini	J-measure
Anneal-U	99.00	98.33	99.33	94.00
Anneal	99.00	98.33	99.33	94.00
australian	77.39	79.57	75.22	80.44
Auto	68.12	53.62	68.12	59.42
balance-scale	79.90	82.78	79.43	78.95
breast-cancer	63.16	63.16	68.42	63.16
Breast	93.99	93.56	92.28	93.99
breastLoss	100.00	99.79	100.00	98.07
Cars	96.95	96.95	96.95	93.89
Cleve	72.28	73.27	67.33	75.25
Crx	77.50	77.00	75.50	79.00
Diabetes	67.19	66.41	64.84	70.70
german-org	66.77	63.77	67.07	64.07
German	67.37	65.57	65.57	64.37
Glass	59.72	66.67	63.89	65.28
glass2	76.36	76.36	72.73	80.00
Golf	100.00	100.00	100.00	78.57
Heart	68.89	75.56	68.89	80.00
Hepatitis	78.85	80.77	76.92	76.92
horse-colic	76.47	72.06	73.53	67.64
ionosphere	92.31	89.74	91.45	85.47
Iris	96.00	94.00	96.00	82.00
labor-neg	82.35	88.24	82.35	64.71
led7	66.23	66.60	67.57	67.40
Lenses	62.50	62.50	62.50	50.00
lymphography	70.00	80.00	70.00	76.00
monk1-bin	83.80	90.74	84.26	68.98
monk1corrupt	65.28	61.81	63.19	48.61
monk1-cross	100.00	97.22	100.00	79.63
monk1-full	80.56	97.69	78.94	72.69
monk1-local	90.05	86.11	96.30	72.22
monk1-org	80.56	97.69	78.94	72.69
monk1	80.56	97.69	78.94	72.69
monk2-bin	70.37	66.44	70.37	65.51
monk2-local	87.04	73.84	81.48	70.14
monk2	71.76	71.07	71.76	71.53
monk3-full	95.37	92.59	95.37	95.83
monk3-local	90.05	93.98	93.29	93.98
monk3-org	95.37	92.59	95.37	95.83
monk3	95.37	92.59	95.37	95.83
mux6	100.00	93.75	100.00	81.25
parity5+5	50.98	50.78	50.98	50.00
Pima	71.48	66.41	70.31	75.39
sameLabel	100.00	100.00	100.00	100.00
Solar	70.37	68.52	70.37	61.11
soybean-large	91.67	89.91	89.04	68.42
soybean-small	100.00	100.00	100.00	75.00
tic-tac-toe	81.25	85.94	86.56	80.63
unknown	50.00	50.00	50.00	50.00
Vehicle	65.25	66.31	62.77	68.79
vote-irvine	95.17	94.48	95.17	94.48
Vote	94.07	97.78	94.07	94.82
waveform-21	64.92	67.00	68.72	68.96
waveform-40	67.00	66.94	66.38	68.15
Wine	90.00	90.00	80.00	86.67
Zoo	85.29	73.53	85.29	61.77

After the boosting process is finished, the composite classifier is obtained by voting each of the component classifiers. Therefore a new case will have a class with the greatest total vote assigned. Boosting has so far proved to be highly accurate on the training set and usually that also stands for the testing set.

In this section we will describe the popular AdaBoost algorithm derived by Freund and Schapire. We used the AdaBoost algorithm to boost the decision trees induced on the basis of different purity measures and compare the results.

3.1 AdaBoost algorithm

As mentioned above, AdaBoost algorithm, introduced by Freund and Schapire, is a method for generating an ensemble of classifiers by successive reweightings of the training cases [4].

The final composite classifier generally performs well on the training cases even when its constituent classifiers are weak. Although boosting in general increases the accuracy, it sometimes leads to deterioration. That can be put down to overfitting or very skewed class distribution across the weight vectors w_t.

AdaBoost algorithm

Given: a set of training cases $i = 1,2,...,N$

Trials: $t = 1,2,...,T$

Initialize: for every case i initial weight $w_1[i] = 1/N$ ($w_t[i]$…weight for case i in the trial t)

For trial $t = 1,2,...,T$:

- Train classifier C_t from the training cases using the weights w_t.
- Calculate the error rate ε_t of the classifier C_t on the training data as the sum of the weights $w_t[i]$ for each misclassified case i.
- If $\varepsilon_t = 0$ or $\varepsilon_t \geq ½$, terminate the process
  - otherwise update the weights $w_{t+1}[i]$ as follows:

$$w_{t+1}[i] = \begin{cases} \frac{w_t[i]}{2\varepsilon_t}; & \text{…if } C_t \text{ misclassifies case } i \\ \frac{w_t[i]}{2(1-\varepsilon_t)}; & \text{…otherwise.} \end{cases}$$

- To classify a case x:
  - Choose class k to maximize the sum

$$\sum \log \frac{1-\varepsilon_t}{\varepsilon_t}$$ for every classifier C_t that predicts class k.

3.2 Experiments and results

Boosting was applied to the decision trees induced on the basis of a specific purity measure. When comparing the successfulness of boosting on the testing sets an observation was made that boosting has made significant improvement on some

databases. We also noticed a slight increase of average accuracy over all databases compared to classic methods.

4 Hybrid purity measures

Each purity measure uses different approach for assessing the information value of attributes with a respect to the class attribute. Therefore it is hard to predict which measure constructs the best hypothesis for a specific problem. For that reason we have introduced new hybrid purity measures, which combine previously described classic purity measures.

First we constructed a new hybrid purity measure based on a sum of pairs of different classic purity measures : $H^{+}(M_1, M_2, A) = M_1(A) + M_2(A)$,

where M_i represents a purity measure.

More specifically, in each node both purity measures were separately calculated for each attribute split and then summed together (i.e. $H^{+}(I_{gainratio}, Gini, A)$ represents a sum of Information gain ratio and Gini). An attribute with the highest summed value was chosen for the test. Thus, the whole decision tree was induced on the basis of a hybrid purity measure H^{+}.

In similar way a hybrid purity measure based on a product of pairs of different purity measures was defined (i.e. Information gain ratio * Chi square):

$H^{\bullet}(M_1, M_2, A) = M_1(A) \cdot M_2(A)$.

Motivated by the basic idea of hybrid purity measures described above we have tried to find a generalized way to combine all classic purity measures. Observing the idea of base purity measures the next logical step seemed to be a construction of general linear combination of those:

$$H^{L}(\mathbf{M}, \mathbf{w}, A) = \mathbf{w}^{T} \cdot \mathbf{M}(A) = \sum_{i} w_i M_i(A)$$

where w_i are randomly generated coefficients.

Each new hybrid purity measure H^L is defined with a vector $\boldsymbol{w}$.

4.1 Experiments and results

The quality of hybrid purity measures described above was tested on all databases from the previous experiments. Since boosting proved to produce better results compared to classic methods, a further experiment was carried using hybrid purity measures out in order to establish the influence of boosting on the greedy decision tree induction method. The results were compared to the results gained with classic and classic boosted methods. In the Table 2 the efficiency of each purity measure is presented as the number of databases where the decision tree induced on a specific purity measure gave best accuracy on a testing set. A hybrid purity measure based on linear combination of classic purity measures proved most successful on 20 databases. However that is still only 11.2% of all databases, which implies that hybrid purity measures cannot be used as a default parameter.

Table 2: Efficiency of purity measures on 56 databases

Purity measure	No. of databases
Greedy Inf. Gain ratio	11
AdaBoost Inf. gain ratio	9
Greedy Chi square	7
AdaBoost Greedy Chi square	7
Greedy Gini	11
AdaBoost Greedy Gini	9
Greedy J measure	5
AdaBoost Greedy J measure	5
Greedy Chi square + Inf. gain ratio	10
AdaBoost Greedy Chi square + Inf. gain ratio	5
Greedy Chi square * Inf. gain ratio	8
AdaBoost Greedy Chi square * Inf. Gain ratio	6
Greedy Gini + Inf. gain ratio	11
AdaBoost Greedy Gini + Inf. gain ratio	11
Greedy Gini * Inf. gain ratio	11
AdaBoost Greedy Gini * Inf. gain ratio	11
Greedy Gini + Chi square	9
AdaBoost Greedy Gini + Chi square	7
Greedy Gini * Chi square	8
AdaBoost Greedy Gini * Chi square	7
Greedy J measure + Inf. gain ratio	12
AdaBoost Greedy J measure + Inf. gain ratio	7
Greedy J measure * Inf. gain ratio	12
AdaBoost Greedy J measure * Inf. gain ratio	5
Greedy J measure + Chi square	7
AdaBoost Greedy J measure + Chi square	8
Greedy J measure * Chi square	8
AdaBoost Greedy J measure * Chi square	6
Greedy J measure + Gini	9
AdaBoost Greedy J measure + Gini	6
Greedy J measure * Gini	12
AdaBoost Greedy J measure * Gini	5
Greedy Boost linear	14
Greedy linear	20

5 Conclusion

In this research we attempted to find general method for fine-tuning the input parameters used in decision tree induction method in order to make decision tree

based data mining user friendly. With that obstacle in mind we tested different purity measures, which are frequently used in the process of greedy decision tree induction. We also introduced new hybrid purity measures and compared their efficiency on 56 UCI databases. In order to improve the induced decision trees we also used the AdaBoost algorithm. Boosting proved to be successful on most databases.

Using hybrid purity measures in greedy algorithm for decision tree induction can be more time-consuming when there is a large amount of data described with many different attributes included. Boosting combined with complex algorithm can be even more demanding. Naturally, one needs to consider whether the improvement in error is worth the additional computational time.

From the results presented in the paper we can summarize that among all purity measures used in this research there was no general one that would behave best on the most of the databases. For that reason we also observed the possible influence of the database domain on the effectiveness of used parameters. We built a meta decision tree on the basis of elementary data about the databases, such as a number of instances, a number of attributes, a percent of missing data, the database domain field, etc. The resulted decision tree showed that there was no significant correlation among used parameters and database domain.

To conclude, presented results show that finding the most suitable parameters for decision tree induction is a very demanding and time-consuming process. The best method that we can recommend from our experiences is a greedy search over all possible parameters.

A final note: Some of the databases were also used by Quinlan in [16, 17], where boosting was compared to a single tree and bagging using C4.5. The accuracy on the testing sets is in most cases higher than the accuracy presented in this paper mostly because we did not use pruning at all. Therefore all decision trees were overfitted but for that very reason the comparison of different purity measures was fair. Due to general overfitting it also seems to be pointless to observe the complexity of the decision trees. But eventually it will be also interesting to observe the influence of using hybrid purity measures on the complexity of the induced decision tree.

References:

[1] Karalič, A. The estimation of probabilities in attribute selection measures for decision tree induction. *Proceedings of ITI-91*, 1991

[2] Kothari, R., Dong, M. Decision Trees for Classification: A review and some new results, *Lecture Notes in Pattern Recognition,*Singapore, 2001

[3] Blake, C.L., Merz, C.J., UCI Repository of machine learning databases, Irvine, *University of California, Department of Information and Computer Science*, 1998

[4] Freund, Y.; Schapire, R. E., Experiments with a new boosting algorithm, *Proceedings Thirteenth International Conference on Machine Learning*, *Morgan Kaufman,* San Francisco, 1996, 148-156

[5] Quinlan, J.R., Discovering Rules by Induction from Large Collections of Examples, *Expert Systems in the Microelectronic Age*, *Ed. D. Michie, Edinburgh University Press,* 1979

[6] Shannon, Weaver, The mathematical theory of communications, *Urbana: The University of Illinois Press*, 1949

[7] Quinlan, J. R., Induction of decision trees, *Machine Learning vol. 1, Kluwer Academic Publishers*,1986

[8] Konenko, I., On Biases in Estimating Multi-Valued Attributes, *Proc. 1st Int. Conf. On Knowledge Discovery and Data Mining*, 1034-1040, Montreal, 1995

[9] Quinlan J.R., C4.5: Programs for Machine Learning, *Morgan Kaufman*, 1993

[10] Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J., Classification and Regression Trees, *Wadsworth International Group*, Belmond, CA, 1984

[11] Snedecor, G. W., Cochran, W. G., Statistical Methods, Eighth Edition, *Iowa State University Press*, 1989

[12] White, A. P., Liu W. Z., Bias in information-based measures in decision tree induction, *Machine Learning*, 1994, Vol.15, pp. 321-329

[13] Smyth, P., Goodman, R. M., Rule induction using information theory, *Piatetsky-Schapiro, G.; Frawley, W. J. (Eds.), Knowledge Discovery in Databases, AAAI Press,* 1991, pp. 159-176

[14] Nazar K., Bramer M.A., Estimating Concept Difficulty with Cross-Entropy, *Knowledge Discovery and Data Mining (ed. A.Bramer), IEEE*, 1999

[15] Bramer M.A., Using J-pruning to reduce overfitting in classification, *Knowledge Based Systems*, 2002, Vol.15, No.5-6, pp. 301-308

[16] Quinlan, J. R., Bagging, boosting and C4.5, *Proceedings Thirteenth National Conference on Artificial Intelligence, AAAI Press,* 1996, pp. 725-730

[17] Quinlan, J. R., MiniBoosting Decision Trees, *Proceedings of Fifteenth National Conference on Artificial Intelligence, AAAI Press*, 1998