T3: A Classification Algorithm for Data Mining

Christos Tjortjis and John Keane

Department of Computation, UMIST, P.O. Box 88, Manchester, M60 1QD, UK
{christos, jak}@co.umist.ac.uk

Abstract. This paper describes and evaluates T3, an algorithm that builds trees of depth at most three, and results in high accuracy whilst keeping the size of the tree reasonably small. T3 is an improvement over T2 in that it builds larger trees and adopts a less greedy approach. T3 gave better results than both T2 and C4.5 when run against publicly available data sets: T3 decreased classification error on average by 47% and generalisation error by 29%, compared to T2; and T3 resulted in 46% smaller trees and 32% less classification error compared to C4.5. Due to its way of handling unknown values, T3 outperforms C4.5 in generalisation by 99% to 66%, on a specific medical dataset.

1 Introduction

Classification produces a function that maps a data item into one of several predefined classes, by inputting a training data set and building a model of the class attribute based on the rest of the attributes. Decision tree classification has an intuitive nature that matches the user's conceptual model without loss of accuracy [4]. However no clear winner exists [5] amongst decision tree classifiers when taking into account *tree size*, *classification* and *generalisation accuracy*[1].

This paper describes and evaluates T3, an algorithm that builds trees of depth at most three, and results in high accuracy whilst keeping the size of the tree reasonably small. T3 outperforms C4.5 and T2 on average. The key concepts where T3 differs from T2 are the maximum *depth* of the tree permitted to be built and the *Maximum Acceptable Error* (MAE) allowed at any node as a tree building stop criterion.

The paper is structured as follows: C4.5 and T2 are briefly described and compared in sections 2 and 3; T3 is presented in section 4; experimental results are given in section 5 and evaluated in section 6; conclusions and future work are presented in section 7.

2 Description of C4.5 and T2

C4.5 is a well-known classification algorithm that constructs decision trees of arbitrary depth in a top-down recursive divide-and-conquer strategy with splits

[1] *Tree size* is measured by counting the number of its nodes, *classification accuracy* is the proportion of records in the training set that are correctly classified and *generalisation accuracy* is the proportion of records in the test set that are correctly classified.

H. Yin et al. (Eds.): IDEAL 2002, LNCS 2412, pp. 50-55, 2002.

maximising the *Gain Ratio* [7]. It is biased, however, in favour of continuous attributes, a weakness partly addressed by later improvements [8]. C4.5 employs a pruning technique that replaces subtrees with leaves, thus reducing overfitting. In a number of datasets the accuracy achieved by C4.5 was comparatively high [7, 8].

T2 calculates optimal decision trees up to depth 2 using two kinds of decision nodes: (1) discrete splits on a discrete attribute, where the node has as many branches as there are possible attribute values, and (2) interval splits on continuous attributes where the node has as many branches as there are intervals and the number of intervals is restricted to be either at most as many as the user specifies if all the branches of the decision node lead to leaves, or otherwise to be at most 2 [2]. The attribute value ‘‘unknown” is treated as a special attribute value and each node has an additional branch, that takes care of unknown attribute values.

3 Comparing T2 with C4.5

T2 was reported to perform better than C4.5 in terms of accuracy in 5 out of 15 data sets, of size up to 3196 records and number of attributes varying between 4 and 60 [2]. C4.5 resulted in a higher accuracy of 4% on average.

These experiments have been verified using 8 of the publicly available datasets from the UCI repository [9] used in [2]. The pruned version of C4.5 trees is used to compare generalisation accuracy, as C4.5 unpruned trees have lower performance. Table 1 illustrates the results in terms of generalisation accuracy. The last column is the quotient of T2's accuracy over that of C4.5. T2 performed better in only 1 out of 8 datasets having on average a 6.3% worse generalisation accuracy than C4.5. T2 performed 2.7% on average worse than C4.5 in terms of classification accuracy.

Table 1: Comparing the generalisation accuracy of T2 and C4.5 pruned trees

Data sets	Generalisation Accuracy (%)		T2 over C4.5 pruned
	T2	C4.5 pruned	
Iris	94.0	92.0	1.02
Hepatitis	67.3	80.8	0.83
Breast-cancer	70.5	74.7	0.94
Cleve	70.3	77.2	0.91
Crx	75.5	83.0	0.91
Pima	76.6	76.6	1.00
Hypotheroid	99.1	99.2	1.00
Chess	86.6	99.5	0.87

4 T3: An Enhancement of T2

Despite its simplicity and its ability to produce reasonably accurate results, T2 has deficiencies such as decreased efficiency when dealing with data sets containing many categorical attributes [1, 3] caused by the greedy approach used for discrete splits; problems when the classification task involves more than four classes [3];

inability to cope with very large data sets [3, 8]; the lack of useful information derived when data sets present complex interrelations, that cannot be fully described by a two-level decision tree; and possible overfitting of the training set [1].

T2's behaviour for various data set sizes has been studied and the maximum depth of 2 restricts its efficiency when dealing with large sets. Hence, the approach here is to enhance T2 with the ability to build trees of depth up to 3. This enhancement is termed *T3* and uses the same building tree approach. The cost of allowing T3's trees to grow bigger, needs to be balanced by limiting the tree size to only that necessary. The approach taken introduces a new parameter called *Maximum Acceptable Error* (MAE). MAE is a positive real number less than 1, used as a stopping criterion during tree building. The idea is based on the observation that T2 uses a greedy tree building approach meaning that further splitting at a node would stop only if the records already classified in this node, belonged to a single class.

However, this greedy approach is not optimal, as minimising the error in the leaf nodes does not necessarily result in minimising the overall error in the whole tree. In fact, it has been proved that a strategy choosing locally optimal splits necessarily produces sub-optimal trees [6]. Furthermore, even minimising classification error does not always cause minimisation of the generalisation error, due to overfitting.

By introducing MAE, the user can specify the level of "purity" in the leaves and stop further building of the tree, concerning a potential node split. MAE has been set to have 4 distinct values, namely 0.0, 0.1, 0.2 and 0.3, meaning that splitting at a node stops even if the error in that node is equal to or below a threshold of 0, 10, 20 or 30% respectively[2].

More precisely, building the tree would stop at a node in two cases: (1) when the maximum depth is reached; (2) at that node when all the records remaining there to be classified belong to the same class in a minimum proportion of 70,80, 90 or 100%.

5 Experimental Results

Several experiments have been done using 22 data sets from the UCI repository that were converted to MLC++ format [5] and one real stroke register data set[3]. The selection included the data sets used in section 3 plus other sets with different number of records, attributes, classes, missing values and different proportions of continuous and discrete attributes. Table 2 displays the selected data sets together with the number of records, attributes, continuous attributes and classes.

The following naming convention is used: T3.0, T3.1, T3.2 and T3.3 are the versions of T3 with depth 3 and MAE set to 0.0, 0.1, 0.2 and 0.3 respectively, while T2.0, T2.1, T2.2 and T2.3 are the versions of T3 with depth 2 and MAE set to 0.0, 0.1, 0.2 and 0.3 respectively. Hence, T2.0 is actually the original T2.

The 8 different versions of T3 were run against all 23 data sets. In 7 of them results were identical for all versions of T3. Those were namely: Breast, Diabetes, Heart,

[2] Higher values of MAE were also used but resulted in lower accuracy and/or trivial trees built.

[3] *Med_123* was provided by Dr Theodoulidis & Dr Saraee, Department of Computation, UMIST. Their contribution to the evaluation of results is also acknowledged.

Pima, Iris, Waveform-21 and Waveform-40. The following discussion concerns the rest of the data sets.

In the remaining 16 data sets T2.3 was the best version in terms of generalisation accuracy, resulting in higher performance on 7 sets. T2.2 was second best achieving highest performance 6 times out of 16. In terms of classification accuracy, T3.0 was a clear winner in all of the 16 sets. The second best version was T3.1 achieving equal to T3.0 performance in 11 cases. Finally, in terms of tree size, T2.3 resulted in smaller trees in 16 out of 16 sets, leaving T2.2 in the second place, as it achieved equally small trees in 8 cases.

Table 2: The data sets used for experimenting with T3

Data sets	Rec.	Att.	Cont	Cl.
Lenses	24	4	-	3
Lymphography	148	18	3	4
Iris	150	4	4	3
Hepatitis	155	19	6	2
Heart	270	13	13	2
Breast-cancer	286	9	-	2
Cleve	303	13	6	2
Monk1	556	6	-	2
Monk2	601	6	-	2
Monk3	554	6	-	2
Vote	435	16	-	2
Soybean	683	35	-	19
Australian	690	14	6	2
Crx	690	15	6	2
Breast	699	10	10	2
Diabetes	768	8	8	2
Pima	768	8	8	2
Med_123	795	37	11	2
Hypotheroid	3163	25	7	2
Chess	3196	36	-	2
Waveform-40	5000	40	40	3
Waveform-21	5000	21	21	3
Mushroom	8124	22	-	2

6 Performance Evaluation

T2.3 had the best performance in all 16 cases in terms of size and in 8 out of 16 cases in terms of generalisation accuracy. That means that T2.3 is by far the best version of T3. However, as expected, T3.0 is better for classification accuracy in all 16 cases. Furthermore in no case did T3.0 result in minimal trees. T3.1, T3.2 and T3.3 resulted in minimal trees in 1, 1 and 3 cases respectively out of 16. A conclusion to be drawn from this is that the less greedy is the approach the smaller is the tree. An explanation is that less greedy approaches, i.e. higher values for maximum acceptable error, cause a “premature” stop to the tree building phase. This argument is also justified by the fact that T2.0, T2.1 and T2.2 resulted in minimal trees in 5, 7 and 8 cases respectively out of 16. A general conclusion is that increasing the size of a tree, increases the classification accuracy, but results in decrease of generalisation accuracy.

Table 3 displays the best version of T3 for each of the 16 data sets, with the relevant tree size, classification and generalisation error of them, in comparison to T2. The table illustrates that classification error is on average decreased by 47% and the generalisation error, decreased on average by 29%, while the relevant trees are on average double the size of the ones built by T2.

Comparing the best overall versions of T3 and C4.5 for classification accuracy, namely T3.0 and C4.5 unpruned, shows that in 9 out of 16 cases T3.0 performed better than C4.5 unpruned, and they were equal twice. On average T3.0 resulted in

32% less classification error than C4.5 unpruned. Similarly the ‘best’ versions can be compared for tree size and generalisation accuracy, that is T2.3 and C4.5 pruned. T2.3 resulted in smaller trees 12 out of 16 times. It also resulted in higher generalisation accuracy in 4 out of 16 cases and was equal 3 times. On average T2.3 resulted in 46% smaller trees but 15% more generalisation error than C4.5 pruned. Results are presented in Table 4.

Table 3: *An evaluation of how much* T3 *improves T2’s performance*

Data sets	T3				T2			T3 over T2		
	best	size	class	gen	size	class	gen	size	class	gen
Lenses	3.0	20	0.0	62.5	10	12.5	62.5	2	0	1
Lymphography	2.0	28	15.3	22.0	28	15.3	22	1	1	1
Hepatitis	3.0	33	1.0	26.9	12	6.8	32.7	2.8	0.15	0.82
Breast-cancer	2.3	46	22	25.3	99	19.9	29.5	0.5	1.11	0.86
Cleve	3.1	49	5.9	22.8	20	15.8	29.7	2.5	0.37	0.77
Monk1	3.0	47	0	0	17	16.9	16.7	2.8	0	0
Monk2	3.0	48	20.0	38.9	14	33.7	39.4	3.4	0.59	0.99
Monk3	2.0	15	6.6	2.8	15	6.6	2.8	1	1	1
Vote	3.1	45	2.7	3.7	17	4	3.7	2.6	0.68	1
Soybean	3.3	72	11.9	12.7	28	28.6	33.8	2.6	0.42	0.38
Australian	3.3	49	12.0	14.3	60	12.6	19.1	0.8	0.95	0.75
Crx	2.2	27	11.8	17.5	65	11.4	24.5	0.4	1.04	0.71
Med_123	3.2	54	0	0.4	18	0.2	0.8	3	0	0.5
Hypotheroid	3.0	24	0.5	0.9	11	0.7	0.9	2.2	0.71	1
Chess	3.0	19	5.9	6.8	10	12.9	13.4	1.9	0.46	0.51
Mushroom	3.0	124	0	0	75	0.5	0.7	1.7	0	0

Of particular interest is the performance of T3 when used on real stroke register data compared to C4.5. More specifically, T3.0 resulted in 0% and 0.8% classification and generalisation error as compared to the respective 20.9% and 33.6% achieved by C4.5. This indicates that T3 may have much potential when used on “real” data that have not been extensively pre-processed like the sets found in [9].

7 Conclusions and Future Work

Experimental results have shown that T3 produces relatively small sized and comprehensible trees with high accuracy in generalisation and classification. It improves the performance of T2, in terms of both generalisation accuracy and particularly classification accuracy. T3 also outperforms C4.5 in terms of tree size and classification accuracy. However, T3’s generalisation accuracy remains lower than that of C4.5. It should be noted that T3 performed exceptionally well on “real” data. T3 addresses T2’s deficiency when dealing with data sets containing many categorical attributes by using a less greedy approach for discrete splits. This is demonstrated by the results for the *Mushroom* and *Breast-cancer* data sets containing many multi-valued discrete attributes. Another reported weakness of T2, that of dealing with data sets that have more than four classes, is addressed by building larger trees as indicated by results for *Soybean-Large*. T3 also partly tackles the potential problem of T2 in capturing less useful information when used on data sets presenting complex interrelations that cannot be fully described by a two-level decision tree. Further work will address the way continuous attributes are treated, as the current algorithm does not improve on T2 in this respect. Scalability is another known weakness of T2 that has not been addressed yet by T3. T3, as T2, seems to achieve better performance for small or medium size data sets.

Table 4: A comparison between T3 and C4.5

Data set	T2.3		C4.5 pr.		T3.0		C4.5 unpr.	
	size	gen	size	gen	size	class	size	class
Lenses	1	62.5	7	37.5	20	0	7	6.2
Lymphography	19	22.0	21	24.0	74	2.0	25	6.1
Hepatitis	1	13.5	11	19.2	33	1.0	17	4.9
Breast-cancer	42	25.3	41	25.3	257	7.9	120	12.6
Cleve	12	26.7	27	22.8	53	5.4	55	5.0
Monk1	17	16.7	18	24.3	47	0	43	9.7
Monk2	14	39.4	31	35.0	48	20.0	73	14.2
Monk3	15	2.8	12	2.8	55	4.1	25	3.3
Vote	9	3.0	7	3.0	45	2.0	25	2.7
Soybean-large	28	33.8	68	10.5	82	10.5	150	3.5
Australian	11	13.9	58	13.0	140	4.6	124	5.0
Crx	33	19.0	58	17.0	171	3.7	90	3.9
Med_123	18	0.8	1	33.6	39	0	186	20.9
Hypotheroid	1	5.2	7	0.8	24	0.5	17	0.5
Chess	7	22.1	53	0.5	19	5.9	63	0.3
Mushroom	16	1.8	30	0	124	0	30	0

References

1. Aha, D.W., Breslow, L.A: Comparing Simplification Procedures for Decision Trees on an Economics Classification, NRL/FR/5510 98-9881, (Technical Report AIC-98-009), May 11, 1998.
2. Auer, P. Holte, R.C., Maass, W.: Theory and Applications of Agnostic PAC-Learning with Small Decision Trees, Proc. 12th Int'l Machine Learning Conf. San Francisco, Morgan Kaufmann 1995, pp. 21-29.
3. Breslow, L., Aha, D.W.: Comparing Tree-Simplification Procedures, Proc. 6th Int'l Workshop Artificial Intelligence and Statistics, Ft. Lauderdale, 1997, pp. 67-74.
4. Ganti, V., Gehrke, J., Ramakrishnan, R.: Mining Very Large Databases, IEEE Computer, Special issue on Data Mining, August 1999.
5. Kohavi, R., Sommerfield, D., Dougherty, J.: Data Mining using MLC++: A Machine Learning Library in C++, Tools with AI, 1996.
6. Murthy, S., Saltzberg, S.: Decision Tree Induction: How effective is the Greedy Heuristic?, Proc. 1st Int'l Conf. on KDD and DM, 1995, pp. 156-161.
7. Quinlan, J.R.: C4.5: Programs for Machine Learning, San Mateo, Morgan Kaufmann, 1993.
8. Quinlan, J.R.: Improved Use of Continuous Attributes in C4.5, Journal of AI Research 4, Morgan Kaufmann 1996, pp. 77-90.
9. http://www.ics.uci.edu/~mlearn/MLRepository.html UCI Machine Learning Repository data sets converted to MLC++ format, http://www.sgi.com/tech/mlc/db/ (last accessed 5/02).