MML Inference of Decision Graphs with Multi-Way Joins and Dynamic Attributes

Peter J. Tan and David L. Dowe

School of Computer Science and Software Engineering, Monash University, Clayton, Vic 3800, Australia
ptan@bruce.csse.monash.edu.au

Abstract. A decision tree is a comprehensible representation that has been widely used in many supervised machine learning domains. But decision trees have two notable problems - those of replication and fragmentation. One way of solving these problems is to introduce the notion of decision graphs - a generalization of the decision tree - which addresses the above problems by allowing for disjunctions, or joins. While various decision graph systems are available, all of these systems impose some forms of restriction on the proposed representations, often leading to either a new redundancy or the original redundancy not being removed. Tan and Dowe (2002) introduced an unrestricted representation called the decision graph with multi-way joins, which has improved representative power and is able to use training data with improved efficiency. In this paper, we resolve the problem of encoding internal repeated structures by introducing dynamic attributes in decision graphs. A refined search heuristic to infer these decision graphs with dynamic attributes using the Minimum Message Length (MML) principle (see Wallace and Boulton (1968), Wallace and Freeman (1987) and Wallace and Dowe (1999)) is also introduced. On both real-world and artificial data, and in terms of both "right"/"wrong" classification accuracy and logarithm of probability "bit-costing" predictive accuracy (for binary and multinomial target attributes), our enhanced multi-way join decision graph program with dynamic attributes improves our Tan and Dowe (2002) multi-way join decision graph program, which in turn significantly outperforms both C4.5 and C5.0. The resultant graphs from the new decision graph scheme are also more concise than both those from C4.5 and from C5.0. We also comment on logarithm of probability as a means of scoring (probabilistic) predictions.

1 Introduction

In spite of the success of decision tree systems in ("right"/"wrong") supervised classification learning, the search for a confirmed improvement of decision trees has remained a continuing topic in the machine learning literature. Two well-known problems from which the decision tree representation suffers have provided incentives for such efforts. The first one is the replication problem, which leads to the duplication of subtrees from disjunctive concepts. The effect of the

replication problem is that many decision tree learning algorithms require an unnecessarily large amount of data to learn disjunctive functions. The second problem is the fragmentation problem, which occurs when the data contains attributes with more than 2 values. Both of the problems increase the size of decision trees and reduce the number of instances in the individual nodes. Several decision graph representations have been introduced to resolve these problems. Decision graphs can be viewed as generalizations of decision trees, and both have decision nodes and leaves. The feature that distinguishes decision graphs from decision trees is that decision graphs may also contain joins (or disjunctions), which are represented by two (or more) nodes having a common child. This representation specifies that two subsets have some common properties, and hence can be considered as one subset. Tan and Dowe recently presented a general decision graph representation called decision graphs with multi-way joins [20], which was more expressive than previous decision graph representations [14, 12, 13, 8, 6, 11, 7]. We also introduced an efficient MML coding scheme for the new decision graph representation. However, the decision graph with multi-way joins [20] is not able to make efficient use of subtrees with internal repeated structures, as has been innovatively done for decision trees in [21]. In this paper, we refine our recent representation [20] by introducing dynamic attributes in decision graphs to solve this problem. We also point out some drawbacks in the search heuristic which led to premature joins in our multi-way join decision graphs [20] and resolve it by proposing a new search heuristic for growing decision graphs. We further advocate (in section 5.1) the merits of logarithm of probability - for binomial [4, 5, 3, 2, 9, 20], multinomial [3, 20] and other [2] distributions - as opposed to other approaches (see e.g., [16, 15]) to scoring probabilistic predictions.

2 Related Works

2.1 Minimum Message Length (MML) and MML Inference

The Minimum Message Length (MML) principle [22, 23, 26, 24] provides a guide for inferring the model of best fit given a set of data. MML inferences involve assigning a code length to each candidate model and searching for the model with minimum two-part message length (code length of the model plus the code length of the data given the model) [23, 26, 24].

MML and the subsequent Minimum Description Length (MDL) principle [19, 8] (see also [24] for a survey) are widely used for model selection in various machine learning problems. In practice, MML and MDL work very well on inference of decision trees. Among efforts that have been put into the development of tree-based classification techniques in recent years, Quinlan and Rivest [18] proposed a method for inferring decision trees using MDL. Wallace and Patrick subsequently [27] presented a refined coding scheme for decision trees using MML in which they identified and corrected some errors in Quinlan and Rivest's derivation of the message length, including pertaining to the issue of probabilistic prediction (cf. section 5.1). Wallace and Patrick also introduced a "Look Ahead" heuristic of arbitrarily many ply for selecting the test attribute

at a node. We re-use the Wallace and Patrick decision tree coding [27] as part of the coding scheme for our new decision graph program. For further details of the implementation, please see [20].

2.2 Decision graphs currently in the literature

As we mentioned in section 1, it is important to resolve the replication and fragmentation problems of decision trees. Many attempts have been made to extend decision trees to decision graphs or graph-like systems. A binary decision graph scheme using MML was introduced by Oliver and Wallace [14, 12, 13]. Other schemes include a generalized decision tree system using MDL [19] proposed by Mehta et al. [8], the HOODG (Hilling-climbing Oblivious read-Once Decision Graphs) system proposed by Kohavi [6], and a decision graph representation called the branch program proposed by Mansour and McAllester [7]. For a more detailed discussion on these systems, see [20, Section 1].

The decision graph system proposed by Tan and Dowe [20] allows multi-way joins. For a similar scheme, see [10, Appendix]. Directed acyclic graphs were used in the Tan and Dowe system [20] as in both the Oliver and Wallace system [14, 12, 13] and the Kohavi system [6]. The main idea behind the coding of the decision graphs with multi-way joins is to decompose a decision graph into a sequence of decision trees and joining patterns [20]. In this way, encoding a decision graph is equivalent to encoding a sequence of decision trees and joining patterns in order. An efficient coding scheme for decision graphs can be achieved by re-using some of the well-proved Wallace-Patrick decision tree coding scheme [27] and devising an efficient coding of the joining patterns [20].

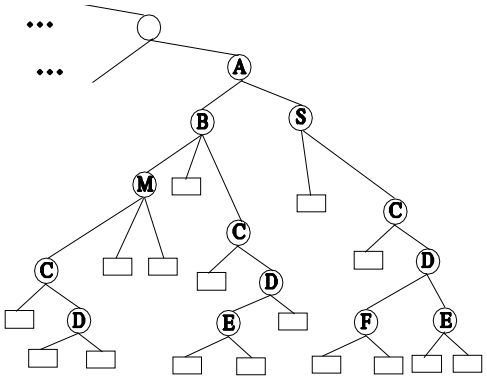

Fig. 1. A decision tree with internal repeated structures (involving C and D)

3 Internal repeated structures and linked decision forests

As discussed in the previous sections, decision graphs are able to represent some duplicated sub-concepts efficiently by uniting these subtrees into one tree. However, in many tree learning problems, these subtrees are not entirely identical

but rather share repeated internal structures. For example, the tree in Figure 1 contains three subtrees with internal repeated structures (involving C and D).

The repeated internal structure problem was first brought forward and studied by Uther and Veloso [21]. Their solution to this problem was to introduce a new representation called the decision linked forest [21], in which the decision tree is turned into a sequence of attribute trees and a root tree. The attribute trees were formed by abstracting the topological structures of the repeated internal structures in the original trees. The attribute trees were then treated as new attributes in the root tree. Their new coding scheme [21] only encodes the repeated sub-concepts once by forming attribute trees as new attributes available to decision trees in the linked decision forest. Their scheme [21] can be explained by Figure 2, which shows how linked decision forests provide a more efficient solution to resolve the internal repeated structure problems in Figure 1.

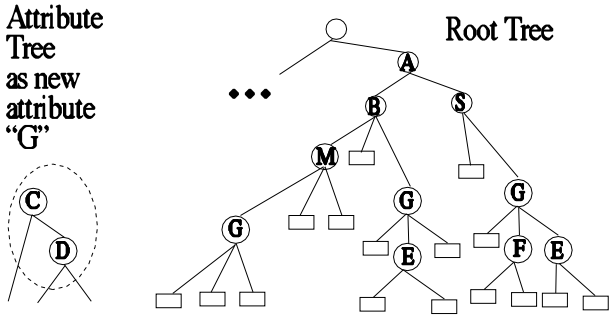

Fig. 2. A linked decision forest [21] with an attribute tree (c.f. Fig. 1)

3.1 Decision graphs with dynamic attributes

Uther and Veloso's novel use of linked decision forests [21] eliminated inefficient coding of the internal repeated structures in a decision tree. However, the root tree of a linked decision forest, where the inference process occurs, is still a decision tree, so the fragmentation problem of decision trees (which is solved by decision graphs) remains unresolved in the linked decision forest. Uther and Veloso [21] claimed that none of the existing decision graph programs was able to resolve the problem of encoding internal repeated structures. We have addressed this issue by introducing dynamic attributes in our decision graph with multi-way joins, which essentially generalises both decision graphs with multi-way joins [20] and Uther-Veloso linked decision forests [21] - as is explained in detail below.

Whenever there is an M-way join operation in a decision graph, a new attribute is created and is made available for every node in the subtrees under the node resulting from the M-way join operation. From the node, back traces are performed along the M joining routes until they reach the root of the decision graph. Then the root and the M routes define a new attribute with arity M. The

purpose of this attribute is to separate the data into M categories corresponding to the way by which the data arrived at the M-way join. When there are several subtrees in a decision graph with internal repeated structures, they are joined to form one subtree consisting of the internal structure. Then, the leaf nodes where there are differences among corresponding leaves in the original subtrees are split on the new attribute. As such, the decision graphs are able to join subtrees with internal repeated structure (immediately before each one would split on this structure) so that the repeated structure is only encoded once. Once created, the new attribute becomes common knowledge to both sender and receiver and thus there is no transmitting cost on its description. (Ideally, the new dynamic attribute is not immediately split on.) We suggest that this scheme provides a more efficient and elegant solution to this problem than linked decision forests and certainly decision graphs described in much of the literature. (We have opted for this scheme rather than for linked decision graph forests.) A solution to resolve the internal repeated structure problems in Figure 1 by our new decision graphs with dynamic attributes is shown in Figure 3.

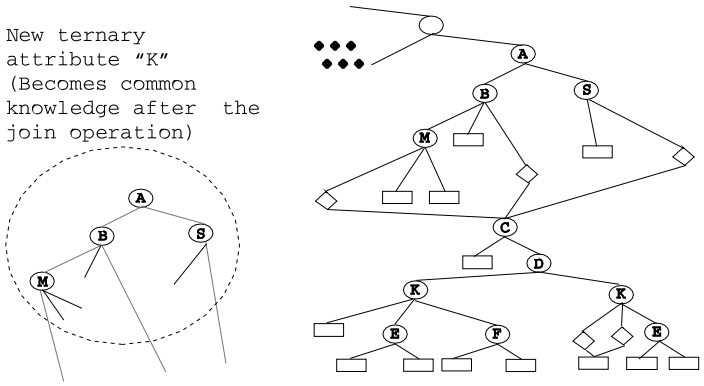

Fig. 3. A decision graph with a dynamic attribute (c.f. Fig. 1 and Fig. 2)

4 Growing a decision graph

While growing a decision tree, the order in which the leaf nodes are expanded is irrelevant to the resultant tree since splitting a leaf node would have no effect on the following actions taken on other leaves. However, the order in which the leaf nodes are expanded or joined is often crucial while growing a decision graph. Such a significant difference between decision tree inference and decision graph inference makes the algorithm used in the former inadequate for the latter. In this section we investigate the MML decision graph growing algorithm implemented for Oliver and Wallace's binary join decision graph program [12–14], and explain a drawback in their search heuristic which makes their program unable to infer

the optimal graph in some circumstances. In section 4.2, we will present our new algorithm for inferring decision graphs with multi-way joins in detail.

4.1 Oliver and Wallace's MML decision graph generation algorithm

Oliver and Wallace's algorithm extends a decision graph by iteratively performing the following procedures until no further improvement can be achieved.

1. For each Leaf, L, determine the attribute A on which it should be split. Record, but do not perform, the alteration (Split L on A) along with its saving in message length.

2. For each pair of leaves, L1 and L2, perform a tentative join. Record, but do not perform, the alteration (Join L1 and L2) along with its saving in message length.

3. Choose the alteration (whether from step 1 - a Split, or from step 2 - a Join) that has the greatest saving. If this alteration creates a saving in message length, then perform that alteration on the graph.

Oliveira et al. [11] reported that this algorithm tended to perform premature joins on complex systems and similar observations were obtained in our tests. The example in Figure 4 shows how and why this could happen.

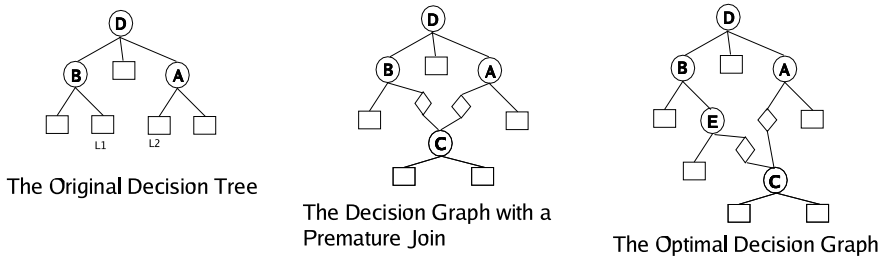

Fig. 4. An example illustrating how the premature joins are generated

Suppose it is decided to grow the decision tree shown on the left of Figure 4. Thus, for leaf L1, splitting on attribute C will save S_1 bits in message length while splitting on attribute E will yield S_1' bits saving in message length. If $S_1' > S_1$, then according to the algorithm above, the alteration that splits L1 on attribute E is recorded. For leaf L2, the same is done for the alteration that splits L2 on attribute C with S_2 bits saving in message length. When performing a tentative join, the same is done again for the alteration that joins L1 and L2 with S_j bits saving in message length. When estimating the saving from a tentative join, a lookahead search whose aim is to look for the subtree with the minimum message length is conducted on the node resulting from the join. Since expanding the node resulting from joining leaf L1 and leaf L2 would be viewed as merging expanded L1 with expanded leaf L2, so the saving is $S_j = S_1 + S_2 - S_g + S_t$, where S_g is the cost in message length to transmit the join, and S_t is the cost to transmit the topological structure of one of the subtrees. In the case when

$S_j > S'_1$, the resultant graph would be the graph shown in the middle of Figure 4 instead of the optimal one shown on the right of Figure 4. The Oliver and Wallace algorithm has a bias toward joins because it compares the sum of the savings from expanding the two leaf nodes with the saving from expanding just one leaf node. This shows why Oliver and Wallace's decision graph growing algorithm produces premature joins in some circumstances.

4.2 The new MML decision graph growing algorithm

If we implement the above algorithm in our decision graph inference scheme, the fact that we allow multi-way joins could only increase such bias. So we propose the following algorithm to eliminate the premature joins. To grow a decision graph, we begin with a graph having one node, with the root being a leaf. We grow the graph by performing the following procedures iteratively until no further improvement can be achieved.

1. For each leaf L, perform tentative splits on each available attribute in the leaf, and determine the attribute A that will lead to the shortest message length when L is split on A. Record, but do not perform, the alteration (Split L on A) along with its rate of communication saving - the communication saving divided by the number of data items in the leaf.

2. For each leaf L, perform tentative joins with other leaves. Record, but do not perform, the alterations (join L_i and L_j; ...; join $L_i, L_j, \ldots, L_k$; etc.) along with its rate of communication savings - the communication saving divided by the number of data items in the join.

3. Sort the alterations from step 1 and step 2 by their communication savings. Choose the alteration (whether from step 1 or from step 2) that has greatest rate of saving.

When splitting on any continuous-valued attributes, we implement a simple single cut-point search algorithm, in which the information gained from the cut is the objective function. Then the cost in message length to state the cut-point is log(the number of values of the attribute in this node - 1). In each iteration, we manipulate the data (i.e., split a leaf or join leaves) so that the greatest rate of saving in message length can be achieved. Thus, it is guaranteed that in the later iterations we will generate a decision graph better or not worse than the possible optimal graph expected in the current iteration. Of course, the algorithm with this search heuristic is only locally optimal.

5 Experiments

One artificially generated data set and eight real-world data sets were used in our tests. The only artificial data set is the XD6 data set [18, 27, 14, 20], which consists of 10 (9 input, 1 output) binary attributes. It was generated according to the boolean function of attributes 1 to 9:

(A1 $\wedge$ A2 $\wedge$ A3) $\vee$ (A4 $\wedge$ A5 $\wedge$ A6) $\vee$ (A7 $\wedge$ A8 $\wedge$ A9)

with 10% noise added to the target attribute. The other eight real-world data sets were downloaded from the UCI machine learning repository [1] and have been

widely tested in other decision tree or decision graph systems [14, 6, 17]. In order to rigorously examine the proposed algorithms, 10 10-fold cross-validations were performed on each of the nine data sets. This amounted to 10x10=100 tests for one single data set. Each pair of training/test data from these tests was fed into four different decision tree and graph algorithms: the well-known decision tree classification programs C4.5, C5 [17], the decision graph with multi-way joins [20] and our new decision graphs with multi-way joins and dynamic attributes.

The experimental results are presented in Tables 1, 2, 3 and 4. In table 1, the run time recorded the execution time of one test by the algorithms on a PIII 1G Linux Redhat7.3 PC. In table 2, "Error Rate" describes the rate of "right"/"wrong" classification errors. In table 3, "pr costing" describes Good's (binomial) probabilistic costing [4, 5], or logarithmic 'bit costing' [2, 3, 20, 5, 4, 9]. In table 4, we compare the size of resultant decision trees and graphs by recording the number of leaf nodes in them. For the data sets on which 10 10-fold cross-validations were performed, the rate of classification errors and probabilistic costings are presented as mean $\pm$ standard deviation, $\mu \pm \sigma$.

Table 1. Summary of Data Sets

Data-set Name	size	Discrete Attributes	Continuous Attributes	Number of Classes	Run Time (C4.5, C5)	Run Time dGraph[20]	Run Time dG (dyn atts)
abalone	4177	1	7	29	1.35s	1062s	1132s
car	1728	6	0	4	0.03s	2.07s	2.42s
cmc	1473	8	2	2	0.06s	11.0s	13.7s
credit	690	9	6	2	0.04s	29.9s	38.9s
led	500	7	0	10	0.01s	4.50s	5.90s
scale	625	4	0	3	0.01s	1.10s	1.70s
tic-tac-toe	958	9	0	3	0.01s	3152s	4031s
vote	435	16	0	2	0.00s	0.01s	0.01s
XD6	500	9	0	2	0.01s	2.55s	3.22s

5.1 Comparing and scoring probabilistic predictions

Decision trees and graphs are often used as classifiers in many machine learning problems. In the case in which the target attribute is multinomial, each leaf node in a tree or graph is given a class label corresponding to the class with the highest inferred probability for this node. However, the multinomial distribution in each leaf node can also be interpreted as a probabilistic prediction model. In this way, the decision trees and decision graphs are not only classifiers, but they can also provide a probabilistic prediction model. Provost and Domingos [16] showed that with some modifications, tree inductions programs can produce very high quality probability estimation trees (PETs). Perlich, Provost and Simonoff [15] also observed that for large data sets, tree induction often produces probability-based rankings that are superior to those generated by logistic regression. Thus, in

addition to the conventional classification accuracy, a metric called probabilistic costing [5, 4, 2, 3, 20, 9] was implemented in our tests for comparisons of probabilistic predictions with C4.5 and C5. It is defined as $-\sum_{i=1}^{n} \log(p_i)$, where n is the total number of test data and p_i is the predicted probability of the true class associated with the corresponding data item [5, 4, 2, 3]. The reader can interpret the metric as the optimal coding length for the test data given the resultant tree and graph models. This metric can be used to approximate (within a constant) the Kullback-Leibler distance between the true (test) model and the inferred model. Its relation to log-likelihood via $-\log(\prod_{i=1}^{n} p_i) = -\sum_{i=1}^{n} \log(p_i)$, its relation to Kullback-Leibler distance and its corresponding general applicability to a wide range of probability distributions (recall section 1) [5, 4, 2, 3, 20, 9] strongly recommend this log(prob) bit costing as a statistically-based general alternative to metrics such as ROC and AUC (Area Under Curve) [16, 15].

Table 2. Test Results ('right'/'wrong' Error rates) %

Data-set Name	C4.5	C5	dGraph with M-way joins [20]	dGraph with dynamic atts
abalone	78.9 ± 1.9	79.0 ±1.8	74.3 ±2.1	74.3 ± 2.1
car	7.8 ± 2.2	7.8±2.1	8.5 ±2.8	6.7 ± 2.8
cmc	48.2 ±3.6	48.5 ± 3.6	48.4 ±3.6	48.2 ±3.6
credit	14.4 ±3.6	14.5 ± 3.6	14.2±4.3	14.2 ±4.3
led	30.0 ±5.2	30.0 ± 5.2	30.0±5.8	30.0 ±5.8
scale	35.6 ±4.9	35.4 ± 3.3	22.0±5.3	22.0 ±5.3
tic-tac-toe	14.4 ±3.4	14.0 ± 3.5	11.9±4.8	10.7 ±4.9
vote	5.0 ± 3.1	5.0 ± 3.1	4.4±3.3	4.4 ± 3.3
XD6	14.1 ±4.9	14.2 ± 5.0	9.2±4.0	9.2 ±4.0

Logarithm of probability (bit) scoring, Pr_cost, enables us to compare probabilistic prediction accuracy of inferences from an identical training data set by various decision tree and graph algorithms. The lower the value of the Pr_cost, the more consistent the predicted probabilistic model is with the true model.

Given an array of occurrences of events of an m-state multinomial distribution $(c_1, c_2, \ldots, c_m)$, the probability of a certain event j can be estimated by (either)

$\hat{p_j} = \frac{c_j + 0.5}{(\sum c_i) + m/2}$ [22, p187 (4), p194 (28), p186 (2)][26][25, p75][20] or

$\hat{p_j} = \frac{c_j + 1}{(\sum c_i) + m}$ [22, p187 (3), p189 (30)][20], the latter being known as the Laplace estimate and also corresponding (with uniform prior) to both the posterior mean and the minimum expected Kullback-Leibler distance estimator [24]. In our experiments, the first (+0.5) was used in MML multinomial message length calculations, $\hat{p_j}$ estimations and calculations of the log(prob) bit costing.

Table 3. Test Results ($-\log(Prob)$ Costing) bits

Data-set Name	C4.5 (+0.5)	C5 (+0.5)	dGraph with M-way joins [20] (+0.5)	dGraph with dyn atts (+0.5)
abalone	1810.3 ± 26.3	1814.5±25.9	1269.6±32.0	1269.8±33.4
car	60.0 ± 9.9	61.2±9.8	49.8 ±10.5	40.7 ±12.0
cmc	221.4±13.3	222.1 ±13.4	202.4±9.9	202.2 ±9.9
credit	38.3 ±8.1	38.4 ± 8.0	35.2±7.5	35.2 ±7.5
led	79.3 ±9.8	79.3±9.7	76.2±10.0	76.2 ±10.0
scale	82.9 ±6.8	80.1±6.6	55.0±10.0	55.0 ±10.0
tic-tac-toe	46.4 ±8.4	45.4 ±7.1	44.7±13.5	41.1 ±14.7
vote	9.8 ± 6.2	9.8 ±6.1	8.6±5.5	8.6 ± 5.5
XD6	28.5 ±7.8	28.4 ±7.1	22.4±6.7	22.4 ±6.7

5.2 Discussions of above test results

Tables 2 and 3 clearly show the decision graph with dynamic attributes to always be either outright first or (sometimes) equal first. When testing on the data sets with disjunctions (like abalone, scale, tic-tac-toe and XD6), decision graph with dynamic attributes has a much lower error rate. On other data sets, it returns results not worse than those from C4.5 and C5. Results from the decision graph with dynamic attributes are either identical or marginally better than those from the decision graphs with multi-way joins [20]. In the cases that the decision graph with dynamic attributes performs better, generated dynamic attributes are found in the resultant graphs. This proves the importance of the dynamic attributes and also shows that decision graphs with dynamic attributes are a superset of decision graphs with multi-way joins [20]. In table 3, the Pr_cost from both kinds of decision graph are clearly lower than those from both C4.5 and C5. This suggests that the decision graphs inferred by MML are resistant to over-fitting. As such, the decision graphs not only produce excellent "right"/"wrong" predictions, but also provide inferred probabilistic models that are clearly more consistent with the test data (cf. also [20, Tables 1 to 3]).

From table 4, we find that the resultant multi-way join decision graphs [20] are similar in size to the decision graphs with dynamic attributes. The sizes of the resultant graphs tend to be substantially smaller than the sizes of both the C4.5 and C5 trees. From table 1, both kinds of MML decision graph take longer to infer than C4.5 and C5 trees. Tables 3, 2, and 4 suggest it is well worth the wait. Nonetheless, we do intend to trim the decision graph searches.

6 Conclusion and discussion

In this paper, we have refined the decision graph with multi-way joins [20] representation by introducing dynamic attributes in the decision graphs. Using the Minimum Message Length principle, an improved coding scheme for inferring the new decision graphs has been devised to address some of the inefficiencies in previous decision tree and decision graph coding schemes. Our experimental

Table 4. Size of Resultant Tree or Graph (Number of leaf nodes)

Data-set Name	C4.5	C5	dGraph with M-Way joins [20]	dGraph with dynamic atts
abalone	2103	1062	315	313
car	170	122	36	38
cmc	230	132	10	10
credit	34	15	7	7
led	42	22	22	22
scale	38	34	21	21
tic-tac-toe	132	88	37	35
vote	16	8	5	5
XD6	52	25	5	5

results demonstrated that our refined coding scheme compares favourably with other decision tree inference schemes, namely both C4.5 and C5. This favourable comparison holds true both for 'right'/'wrong' prediction accuracy and especially for I.J. Good's logarithm of probability bit costing (recall section 5.1 and table 3), as well as for both artificially generated and real-world data.

In future work, we hope to both speed up the decision graph searches and compare with more programs, such as, e.g., Yin and Han's CPAR [28].

References

1. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html.
2. D.L. Dowe, G.E. Farr, A.J. Hurst, and K.L. Lentin. Information-theoretic football tipping. In N. de Mestre, editor, *Third Australian Conference on Mathematics and Computers in Sport*, pages 233–241. Bond University, Qld, Australia, 1996. http://www.csse.monash.edu.au/~footy.
3. D.L. Dowe and N. Krusel. A decision tree model of bushfire activity. In *(Technical report 93/190) Dept Computer Science, Monash University, Clayton, Vic. 3800, Australia*, 1993.
4. I.J. Good. Rational Decisions. *Journal of the Royal Statistical Society. Series B*, 14:107–114, 1952.
5. I.J. Good. Corroboration, Explanation, Evolving Probability, Simplicity, and a Sharpened Razor. *British Journal of Philosophy of Science*, 19:123–143, 1968.
6. Ron Kohavi. Bottom-up induction of oblivious read-once decision graphs: Strengths and limitations. In *National Conference on Artificial Intelligence*, pages 613–618, 1994.
7. Yishay Mansour and David McAllester. Boosting using branching programs. In *Proc. 13th Annual Conference on Comput. Learning Theory (CoLT)*, pages 220–224. Morgan Kaufmann, San Francisco, 2000.
8. Manish Mehta, Jorma Rissanen, and Rakesh Agrawal. MDL-based Decision Tree Pruning. In *The First International Conference on Knowledge Discovery & Data Mining*, pages 216–221. AAAI Press, 1995.

9. S.L. Needham and D.L. Dowe. Message length as an effective Ockham's razor in decision tree induction. In *Proc. 8th International Workshop on Artificial Intelligence and Statistics (AI+STATS 2001)*, pages 253–260, Key West, Florida, U.S.A., Jan. 2001.
10. Julian R. Neil. *MML discovery of Causal Models.* PhD thesis, Monash University, Clayton 3800, Australia, Computer Science and Software Engineering, 2001.
11. Arlindo L. Oliveira and Alberto L. Sangiovanni-Vincentelli. Using the minimum description length principle to infer reduced ordered decision graphs. *Machine Learning*, 25(1):23–50, 1996.
12. J.J. Oliver. Decision Graphs - An Extension of Decision Trees. In *Proc. 4th International Workshop on Artif. Intelligence and Statistics*, pages 343–350, 1993.
13. J.J. Oliver, D.L. Dowe, and C.S. Wallace. Inferring Decision Graphs Using the Minimum Message Length Principle. In *Proceedings of the 5th Joint Conference on Artificial Intelligence*, pages 361–367. World Scientific, Singapore, 1992.
14. J.J. Oliver and C.S. Wallace. Inferring Decision Graphs. In *Workshop 8 International Joint Conference on AI (IJCAI)*, Sydney, Australia, August 1991.
15. C. Perlich, F. Provost, and J.S. Simonoff. Tree induction versus logistic regression: a learning-curve analysis. *Journal of Machine Learning Research*, 4:211–255, 2003.
16. Foster Provost and Pedro Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52:199–215, Sept. 2003.
17. J.R. Quinlan. *C4.5 : Programs for Machine Learning.* Morgan Kaufmann,San Mateo,CA, 1992. The latest version of C5 is available from http://www.rulequest.com.
18. J.R. Quinlan and R. Rivest. Inferring Decision Trees Using the Minimum Description Length Principle. *Information and Computation*, 80:227–248, 1989.
19. J.J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.
20. P.J. Tan and D.L. Dowe. MML inference of decision graphs with multi-way joins. In *Proc. 15th Australian Joint Conf. on AI, LNAI 2557 (Springer)*, pages 131–142, Canberra, Australia, 2-6 Dec. 2002.
21. W.T.B. Uther and M.M. Veloso. The Lumberjack Algorithm for Learning Linked Decision Tree. In *Proc. 6th Pacific Rim International Conf. on Artificial Intelligence (PRICAI'2000), LNAI 1886 (Springer)*, pages 156–166, 2000.
22. C.S. Wallace and D.M. Boulton. An Information Measure for Classification. *Computer Journal*, 11:185–194, 1968.
23. C.S. Wallace and D.M. Boulton. An Invariant Bayes Method for Point Estimation. *Classification Society Bull.*, 3:11–34, 1975.
24. C.S. Wallace and D.L. Dowe. Minimum Message Length and Kolmogorov Complexity. *Computer Journal, Special Issue - Kolmogorov Complexity*, 42(4):270–283, 1999.
25. C.S. Wallace and D.L. Dowe. MML Clustering of multi-state, Poisson, von Mises circular and Gaussian distributions. *Statistics and Computing*, 10(1):73–83, Jan 2000.
26. C.S. Wallace and P.R. Freeman. Estimation and Inference by Compact Coding. *Journal of the Royal Statistical Society. Series B*, 49(3):240–265, 1987.
27. C.S Wallace and J.D. Patrick. Coding Decision Trees. *Machine Learning*, 11:7–22, 1993.
28. Xiaoxin Yin and Jiawei Han. CPAR: Classification based on predictive association rules. In *SIAM International Conference on Data Mining.* San Francisco, CA, USA, May 2003.