Decision Trees for Knowledge Representation*

Mohammad Azad[1], Igor Chikalov[2], and Mikhail Moshkov[2]

[1] Jouf University
College of Computer and Information Sciences
Sakaka 72388, Saudi Arabia
mmazad@ju.edu.sa

[2] King Abdullah University of Science and Technology (KAUST)
Computer, Electrical and Mathematical Sciences & Engineering Division
Thuwal 23955-6900, Saudi Arabia
igor.chikalov@gmail.com, mikhail.moshkov@kaust.edu.sa

Abstract. In this paper, we consider decision trees as a means of knowledge representation. To this end, we design three algorithms for decision tree construction that are based on extensions of dynamic programming. We study three parameters of the decision trees constructed by these algorithms: number of nodes, global misclassification rate, and local misclassification rate.

Keywords: knowledge representation, decision trees, extensions of dynamic programming

1 Introduction

Decision trees are widely used as classifiers, as a means of knowledge representation, and as algorithms [3, 5, 8]. In this paper, we consider decision trees as a means of knowledge representation.

Let T be a decision table and Γ be a decision tree for the table T [1]. We study three parameters of the tree Γ:

- $N(\Gamma)$ – the *number of nodes* in Γ.
- $G(T, \Gamma)$ – the *global misclassification rate* which is equal to the number of misclassifications of Γ on T divided by the number of rows in T.
- $L(T, \Gamma)$ – the *local misclassification rate*. For each terminal node v of Γ, we find the number of misclassifications of Γ on rows of T for which the computation finishes in v divided by the number of rows of T for which the computation finishes in v, and consider maximum of this parameter among all terminal nodes of Γ. It is easy to show that $G(T, \Gamma) \leq L(T, \Gamma)$.

To be understandable, the decision tree Γ should have a reasonable number of nodes. To represent properly knowledge from the decision table T, the decision tree Γ should have a reasonable accuracy. The consideration of only the global misclassification rate may be insufficient: the misclassifications may be unevenly distributed and, for some terminal nodes, the fraction of misclassifications can be high. To deal with this situation, we should consider also the local misclassification rate.

In this paper, we design three algorithms for decision tree construction which are applicable to medium-sized decision tables with only categorical attributes. These algorithms are based on extensions of dynamic programming – bi-criteria optimization of decision trees relative to the parameters N and G, and relative to the parameters N and L [1]. One of the algorithms (GL-algorithm) is completely new. We apply the considered algorithms to 14 decision tables from the UCI Machine Learning Repository [4], and study three parameters N, G, and L of the constructed trees.

The obtained results show that at least one of the considered algorithms (GL-algorithm) can be useful for the extraction of knowledge from medium-sized decision tables and for its representation by decision trees. This algorithm can be used in different areas of data analysis including rough set theory [6, 7].

The rest of the paper is organized as follows. In Sect. 2, we describe three algorithms for decision tree construction. In Sect. 3, we discuss results of experiments with decision tables from the UCI ML Repository [4]. Section 4 contains short conclusions.

2 Three Algorithms for Decision Tree Construction

In the book [1], we described an algorithm $\mathcal{A}_7$ which, for a given decision table, constructs the set of Pareto optimal points (POPs) for the problem of bi-criteria optimization of decision trees relative to the parameters N and G (see, for example, Fig. 1 (a), (c), (e)). The same algorithm $\mathcal{A}_7$ can also construct the set of POPs for the problem of bi-criteria optimization of decision trees relative to the parameters N and L (see, for example, Fig. 1 (b), (d), (f)). For each POP, we can derive a decision tree with values of the considered parameters equal to the coordinates of this point.

We now describe three algorithms for decision tree construction based on the use of the algorithm $\mathcal{A}_7$.

2.1 *G*-Algorithm

For a given decision table T, we construct using the algorithm $\mathcal{A}_7$ the set of POPs for the parameters N and G. We normalize coordinates of POPs: for each POP, divide each coordinate by the maximum value of this coordinate among all POPs. After that, we choose a normalized POP with the minimum Euclidean distance from the origin. We restore coordinates of this point and derive a decision tree Γ, for which the values of the parameters N and G are equal to the restored coordinates. The tree Γ is the output of G-algorithm.

2.2 *L*-Algorithm

L-algorithm works in the same way as G-algorithm but instead of the parameters N and G it uses the parameters N and L.

2.3 *GL*-Algorithm

We apply G-algorithm to a given decision table T and construct a decision tree Γ_1. After that, using the algorithm $\mathcal{A}_7$ we construct the set of POPs for the parameters N and L, and choose a POP for which the value of the coordinate N is closest to $N(\Gamma_1)$. At the end, we derive a decision tree Γ_2 for which the values of the parameters N and L are equal to the coordinates of the chosen POP. The tree Γ_2 is the output of GL-algorithm.

3 Results of Experiments

We made experiments with 14 decision tables from the UCI ML Repository [4] described in Table 1.

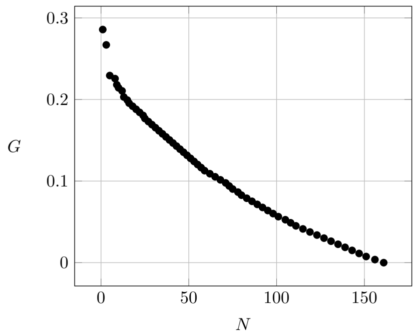
Table 1. Decision tables used in experiments

Decision table	Number of rows	Number of attributes
BALANCE-SCALE	625	5
BREAST-CANCER	266	10
CARS	1728	7
HAYES-ROTH-DATA	69	5
HOUSE-VOTES-84	279	17
IRIS	150	5
LENSES	10	5
LYMPHOGRAPHY	148	19
NURSERY	12960	9
SHUTTLE-LANDING	15	7
SOYBEAN-SMALL	47	36
SPECT-TEST	169	23
TIC-TAC-TOE	958	10
ZOO-DATA	59	17

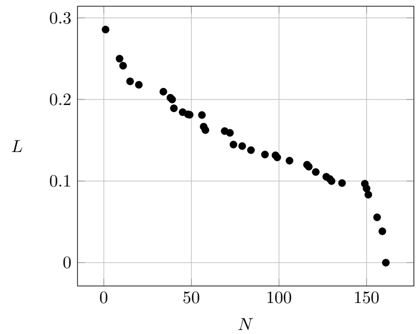
We applied G-algorithm, L-algorithm, and GL-algorithm to each of these tables and found values of the parameters N, G, and L for the constructed decision trees. Results of experiments can be found in Table 2.

Decision trees constructed by G-algorithm have overall reasonable values of the parameters N and G but often have high values of the parameter L.

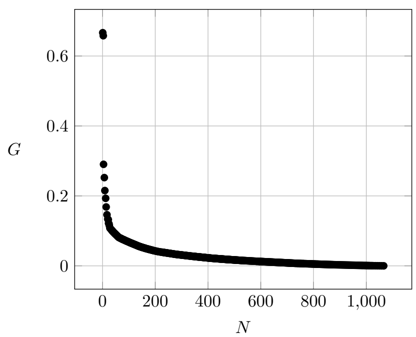
Decision trees constructed by L-algorithm have overall reasonable values of the parameters G and L but sometimes have high values of the parameter N.



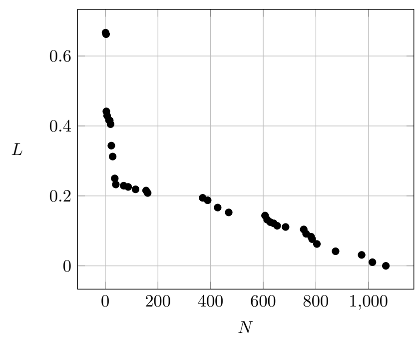
(a) BREAST-CANCER, N and G



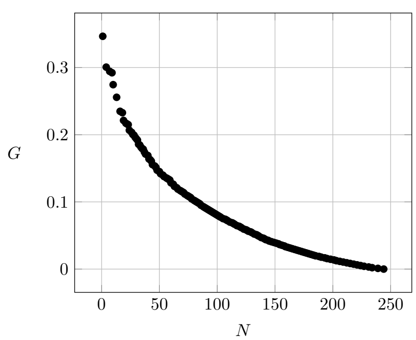
(b) BREAST-CANCER, N and L



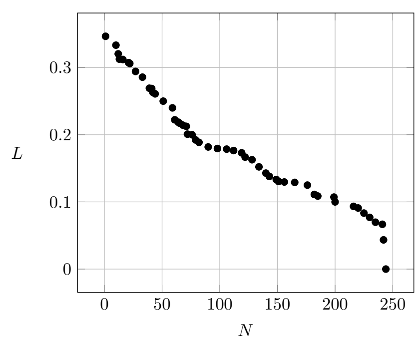
(c) NURSERY, N and G



(d) NURSERY, N and L



(e) TIC-TAC-TOE, N and G



(f) TIC-TAC-TOE, N and L

Fig. 1. Sets of Pareto optimal points for decision tables BREAST-CANCER, NURSERY, and TIC-TAC-TOE for pairs of parameters N, G and N, L

Decision trees constructed by GL-algorithm have overall reasonable values of the parameters N, G, and L. We can use GL-algorithm to construct enough understandable and accurate decision trees.

Table 2. Results of experiments

Decision table	G-algorithm			L-algorithm			GL-algorithm		
	N	G	L	N	G	L	N	G	L
BALANCE-SCALE	106	0.18	0.40	186	0.14	0.24	91	0.22	0.32
BREAST-CANCER	59	0.11	0.33	58	0.13	0.16	58	0.13	0.16
CARS	98	0.08	0.50	338	0.01	0.08	135	0.14	0.29
HAYES-ROTH-DATA	23	0.16	0.50	26	0.13	0.33	26	0.13	0.33
HOUSE-VOTES-84	3	0.06	0.12	11	0.03	0.06	3	0.06	0.13
IRIS	5	0.04	0.09	5	0.04	0.09	5	0.04	0.09
LENSES	6	0.10	0.50	8	0.00	0.00	8	0.00	0.00
LYMPHOGRAPHY	13	0.13	0.33	11	0.16	0.20	11	0.16	0.20
NURSERY	74	0.08	0.34	115	0.09	0.22	70	0.10	0.23
SHUTTLE-LANDING	7	0.20	0.33	5	0.27	0.31	5	0.27	0.31
SOYBEAN-SMALL	3	0.42	0.55	3	0.43	0.50	3	0.43	0.50
SPECT-TEST	17	0.02	0.10	19	0.02	0.02	17	0.02	0.03
TIC-TAC-TOE	72	0.11	0.46	82	0.12	0.19	72	0.15	0.20
ZOO-DATA	8	0.19	0.43	9	0.20	0.28	9	0.20	0.28
Average	35.29	0.13	0.36	62.57	0.13	0.19	36.64	0.15	0.22

4 Conclusions

We proposed to evaluate the accuracy of decision trees not only by the global misclassification rate G but also by the local misclassification rate L, and designed GL-algorithm which constructs decision trees with mostly reasonable number of nodes and mostly reasonable values of the parameters G and L. Later we are planning to extend the considered technique to the case of decision tables with many-valued decisions using bi-criteria optimization algorithms described in [2]. We are planning also to extend this technique to the case of decision tables with both categorical and numerical attributes.

Acknowledgments

Research reported in this publication was supported by King Abdullah University of Science and Technology (KAUST). The authors are greatly indebted to the anonymous reviewers for useful comments.

References

1. AbouEisha, H., Amin, T., Chikalov, I., Hussain, S., Moshkov, M.: Extensions of Dynamic Programming for Combinatorial Optimization and Data Mining, Intelligent Systems Reference Library, vol. 146. Springer (2019)
2. Alsolami, F., Azad, M., Chikalov, I., Moshkov, M.: Decision and Inhibitory Trees and Rules for Decision Tables with Many-valued Decisions, Intelligent Systems Reference Library, vol. 156. Springer (2020)

3. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA (1984)
4. Lichman, M.: UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences (2013), http://archive.ics.uci.edu/ml
5. Moshkov, M.: Time complexity of decision trees. In: Peters, J.F., Skowron, A. (eds.) Trans. Rough Sets III, Lecture Notes in Computer Science, vol. 3400, pp. 244–459. Springer (2005)
6. Pawlak, Z.: Rough Sets – Theoretical Aspect of Reasoning About Data. Kluwer Academic Publishers, Dordrecht (1991)
7. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
8. Rokach, L., Maimon, O.: Data Mining with Decision Trees: Theory and Applications. World Scientific Publishing, River Edge, NJ (2008)