Machine Learning, 19, 29–43 (1995)
© 1995 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Monotonicity Maintenance in Information-Theoretic Machine Learning Algorithms

ARIE BEN-DAVID msariebd@pluto.mscc.huji.ac.il
Management Information Systems, School of Business Administration, The Hebrew University of Jerusalem, Mount Scopus, Jerusalem 91905, Israel

Editor: Paul Utgoff

Abstract. Decision trees that are based on information-theory are useful paradigms for learning from examples. However, in some real-world applications, known information-theoretic methods frequently generate non-monotonic decision trees, in which objects with better attribute values are sometimes classified to lower classes than objects with inferior values. This property is undesirable for problem solving in many application domains, such as credit scoring and insurance premium determination, where monotonicity of subsequent classifications is important. An attribute-selection metric is proposed here that takes both the error as well as monotonicity into account while building decision trees. The metric is empirically shown capable of significantly reducing the degree of non-monotonicity of decision trees without sacrificing their inductive accuracy.

Keywords: information theory, monotonic decision trees, consistency, accuracy, monotonic classification problems

1 Introduction

Suppose a college admissions committee decides to use decision trees to determine whom to admit based on standardized test scores and grades. For reasons such as fairness and liability, the college would not want to use a decision tree that admits an applicant with certain scores, and then rejects another who scores as high or higher on each measure. Similarly, a life insurance company would not wish to rely on a decision tree that quotes a young and healthy applicant a higher premium rate than one that has been quoted to an old and unhealthy person.

The classifications in both the school admissions and the life insurance premium problems are required to be monotonic with respect to the attribute values. These problems are, therefore, called *monotonic classification (MOC) problems.* MOC problems are important because they are very common, and deal with many aspects of our life. In addition to the examples given above, MOC problems include, among others, credit scoring (Carter, 1987), consumer choice (Jacoby, 1974), school and transportation selection, investment decisions, referee and editorial decisions (Larichev, 1988), employee selection, lecturer evaluation, and certain social workers decisions (Ben-David, 1992). The examples that are used to construct decision trees for real-world MOC problems are frequently non-monotonic with respect to each other, in particular, when the examples are taken from past human decisions (Jacoby, 1974; Hayes-Roth, 1983).

Ideally, decision trees for MOC problems should be monotonic, regardless of whether their training sets are monotonic or not. Unfortunately, information-theoretic top-down-induction decision tree (TDIDT) algorithms (Quinlan, 1986), that use entropy as the criterion

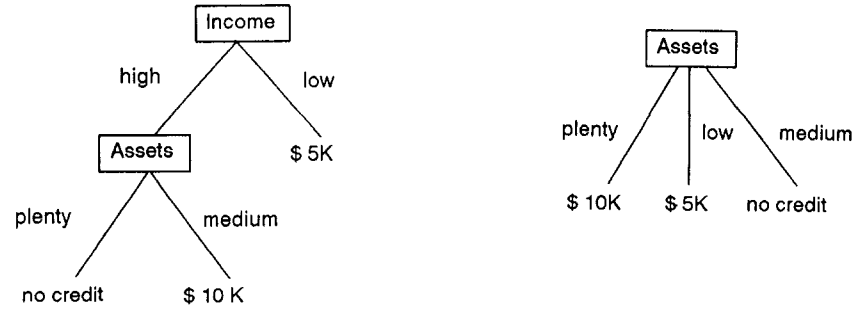

Figure 1. Non-monotonic decision trees.

for attribute selection may produce non-monotonic decision trees. Figure 1 illustrates two such cases of simplified credit scoring decision trees. The attributes are *assets*, and *income + assets* respectively. The decisions are shown at the leaves. It is quite evident that the decision tree of *assets* does not make much sense. A client with low assets is authorized a $ 5K line of credit, while one with more assets is refused. It is easy to show that the *income + assets* decision tree suffers from similar anomalies.

TDIDT algorithms that use the E-score as their attribute selection metric do not consider the order within the attribute values and among the classes. The same observation applies to Mantaras's distance-between-partitions measure (Mantaras, 1991), and to Nunez's background knowledge (Nunez, 1991). Consequently, TDIDT algorithms are not well adapted to deal with MOC problems. The above shortcoming is shared by some other well known learning from examples paradigms: Feed-forward neural networks (Rumelhart, 1986), most of Michalski's AQ family of models (Michalski, 1983), CN2 (Clark & Niblett, 1989), and Fisher's COBWEB (1987), suffer from the same limitation while dealing with MOC problems.

However, decision trees are also required to provide acceptable inductive accuracy. Unfortunately, in real-world cases, these two goals often conflict. Clearly, the tradeoff between monotonic classifications and inductive accuracy is domain dependent. Legal requirements, if applicable, push toward monotonicity of classifications (see above). Human-related considerations also motivate the use of monotonic decision trees, since end-users often consider non-monotonic classifications of MOC problems as unacceptable (see also Larichev & Moshkovich, 1988; Ben-David, 1992).

This paper presents a metric that can improve the monotonicity of decision trees, with little, or no loss of accuracy. The metric's properties are empirically studied on five real-world MOC problems.

2 Monotonicity and decision trees

We begin with a few formal definitions.

DEFINITION 1. Let $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ denote two instances in the same problem domain, described by attributes 1 through n. All the attribute values, $x_i s$ and $y_i s$, are assumed to be ordinal (i.e., ordered) or numeric. An order between X and

Table 1. A monotonic training set.

Example	Income	Assets	Credit history	Class
#1	high	plenty	good	$ 10K
#2	high	low	bad	$ 5K
#3	low	medium	bad	no credit

Y is defined as follows:

$$\begin{array}{lll} X = Y & \text{if } x_i = y_i & \forall i = 1, 2, \ldots, n \\ X > Y & \text{if } x_i > y_i & \forall i = 1, 2, \ldots, n \\ X \geq Y & \text{if } x_i > y_i \quad \text{OR} \quad x_i = y_i & \forall = 1, 2, \ldots, n \\ X < Y & \text{if } x_i < y_i & \forall i = 1, 2, \ldots, n \\ X \leq Y & \text{if } x_i < y_i \quad \text{OR} \quad x_i = y_i & \forall i = 1, 2, \ldots, n \end{array}$$

The non-monotonic relation between attribute-class pairs is defined now:

DEFINITION 2. Let (X, C_x) and (Y, C_y) represent two attribute-class pairs, where X and Y are attribute values as in Definition (1). The classes of X and Y are denoted by C_x and C_y respectively. The attribute-class pairs (X, C_x) and (Y, C_y) are non-monotonic with respect to each other if:

$$\begin{array}{ll} X \leq Y \wedge C_x > C_y & \text{OR} \\ X \geq Y \wedge C_x < C_y & \text{OR} \\ X = Y \wedge C_x \neq C_y & \end{array}$$

The monotonic relation between attribute-class pairs can now be defined as:

DEFINITION 3. Two attribute-class pairs (X, C_x) and (Y, C_y) are monotonic with respect to each other if they do not meet any of the conditions set forth in Definition (2).

The definition of monotonicity between attribute-class pairs can be extended to attribute-test/answer-node paths in decision trees.

DEFINITION 4. Let (P, C_p) and (Q, C_q) be two attribute-test/answer-node paths in the same decision tree, were P and Q are attribute-tests, and C_p and C_q are answer-nodes. The paths (P, C_p) and (Q, C_q) are monotonic with respect to each other if they do not comply with any of the conditions set forth in Definition (2).

DEFINITION 5. A decision tree is *monotonic* if all its attribute-test/answer-node pairs are monotonic with respect to each other.

To illustrate a basic problem that frequently occurs while building decision trees for MOC problems, consider the following example: Credit worthiness is determined by considering *income* level, *assets*, and *credit history*. There are only three examples in our simplified case, and they are shown in Table 1.

It is easy to show that all the examples of Table 1 are monotonic with respect to each other. We apply now the ID3 algorithm to the examples of Table 1. The E-score (i.e., entropy) of income, $E(income) = 0.667$ bits $(2/3 * 1 + 1/3 * 0)$. The entropy vanishes on assets, $E(assets) = 0$. Although the examples in the training set were all monotonic with respect to each other, the resulting decision tree, shown on the right side of Fig. 1, is non-monotonic. It is, however, unambiguous as far as information theory is concerned. We formalize the above observation in the following proposition:

PROPOSITION 1. *A training set in which all the examples are monotonic with respect to each other is not guaranteed to generate monotonic decision trees via information-theoretic TDIDT algorithms that use entropy for attribute selection.*

PROOF. The example of Table 1.

Clearly, statistical outlier-detection techniques may be employed on non-monotonic examples. However, these methods are not guaranteed to be effective, since non-monotonic examples are not necessarily outliers. Note that even if those methods could have always ended with monotonic training sets, we have just seen that information-theoretic metrics cannot generally guarantee the generation of monotonic decision trees from monotonic training sets.

3 Building accurate monotonic decision trees

While information-theoretic metrics attempt to minimize the error without regard to monotonicity, other known algorithms, such as matrix-based methods, and the OLM (Ben-David, 1989; 1992), result in purely monotonic decision trees without regard to error. The main disadvantage of both currently known TDIDT algorithms and monotonicity-oriented methods for solving MOC problems stems from their bias toward a single goal. In most real-world MOC applications, however, tradeoffs between accuracy and monotonicity do exist. This section proposes a metric that allows such tradeoffs.

TDIDT algorithms are quite well known, and will not be reiterated here. Matrix-based methods represent relations among k branches of a decision tree by a $k \times k$ symmetric matrix M. The m_{ij} element of M is 1 if branch i is non-monotonic with respect to branch j, and 0 otherwise. Each row (column) is associated with a counter, in which the sum of the respective row (column) is recorded. Beginning with those branches that are non-monotonic with respect to most of the other branches, their rows and columns are deleted, and the counters are updated. The branch pruning repeats until either all row (column) counters are zero (i.e., the tree is monotonic), or the matrix M becomes of size 1×1. The matrix M is called a *non-monotonicity matrix.*

Another algorithm that can be used for generating monotonic decision trees is the Ordinal Learning Model (OLM) (Ben-David et al., 1989, 1992). The OLM picks a branch of a decision tree at random and declares it monotonic. It later picks a second branch at random. If the second branch is monotonic with respect to the first, it is also declared monotonic. Otherwise, the second branch is discarded. The monotonicity checks continue for all the branches. Each branch has to be monotonic with respect to its predecessors that already have been declared monotonic. Otherwise, it is rejected. This simple conflict resolution

strategy is relatively fast, and it will be used here later in the Empirical Results section. Other, more complex, conflict resolution methods are discussed in the above mentioned publications as well as in the references therein.

Unlike the above single goal models, a new metric is proposed here that takes into account both error and monotonicity considerations. We first define a measure of non-monotonicity of decision trees:

DEFINITION 6. A *non-monotonicity index* is the ratio between the actual number of non-monotonic branch pairs of a decision tree, and the maximum number of pairs that could have been non-monotonic with respect to each other in the same tree.

To find the non-monotonicity index of a given decision tree with k branches, construct a $k \times k$ non-monotonicity matrix M as discussed earlier. The sum of M's entries is denoted W.

$$W = \sum_{i=1}^{k} \sum_{j=1}^{k} m_{ij}$$

At most $(k^2 - k)$ entries of M may be labeled non-monotonic (a branch cannot be non-monotonic with respect to itself). The non-monotonicity index of a decision tree with attribute tests $a_1, a_2, \ldots, a_v$ is defined as:

$$I_{a_1,a_2,\ldots,a_v} = \frac{W_{a_1,a_2,\ldots,a_v}}{k^2_{a_1,a_2,\ldots,a_v} - k_{a_1,a_2,\ldots,a_v}}$$

Consider, again, the *assets* decision trees of Fig. 1. There is one non-monotonic pair of branches in the *assets* tree:

(*low*, \$ 5K), (*medium*, *no credit*).

The non-monotonicity matrix is always symmetric, hence $W_{\text{assets}} = 2$. The number of branches is $k_{\text{assets}} = 3$. Therefore, the non-monotonicity index of the *assets* decision tree is

$$I_{\text{assets}} = \frac{2}{3^2 - 3} = 0.333$$

The *income + assets* decision tree of Fig. 1 has two pairs of inconsistent branches:

(*high*, *plenty*, *no credit*), (*high*, *medium*, \$ 10K), and
(*high*, *plenty*, *no credit*), (*low*, ⟨*don't care*⟩, \$ 5K),

Note that if the ⟨don't care⟩ is replaced by *low*, the latter pair is clearly non-monotonic with respect to each other. Therefore

$$I_{\text{income+assets}} = \frac{2+2}{3^2 - 3} = 0.667$$

The non-monotonicity index is used in the following definition:

DEFINITION 7. The *order-ambiguity-score* of a decision tree is defined in terms of the non-monotonicity index.

$$A_{a_1,a_2,\ldots,a_v} = \begin{cases} 0 & \text{if } I_{a_1,a_2,\ldots,a_v} = 0 \\ -(\log_2 I_{a_1,a_2,\ldots,a_v})^{-1} & \text{otherwise} \end{cases}$$

The order-ambiguity-score is added to the E-score as follows:

DEFINITION 8. The *total-ambiguity-score* is the sum of the E-score, as defined in the ID3 algorithm, and the order-ambiguity-score.

$$T_{a_1,a_2,\ldots,a_v} = E_{a_1,a_2,\ldots,a_v} + A_{a_1,a_2,\ldots,a_v}$$

The metric that is proposed here selects the attribute with the lowest total-ambiguity-score. The total-ambiguity-score has some desirable properties for MOC problems. Unlike ID3's E-score, it considers both the error as well as monotonicity. The value of the order-ambiguity-score increases with the non-monotonicity index. The use of a logarithmic scale for the definition of the order-ambiguity-score is only natural, since the total-ambiguity-score is defined as the sum of the E-score (which is logarithmic), and the order-ambiguity-score.

The above definition of the total-ambiguity-score does not imply that monotonicity considerations necessarily dominate the tree building procedure. Rather, the value of the order-ambiguity-score is lower than 1 for non-monotonicity indices lower than 0.50. In realistic MOC problems, such as those to be studied in the next section, the values of the non-monotonicity indices are substantially lower than 0.5, and one has to verify that the values of the order-ambiguity-scores are not too low relative to the E-scores.

An effective way of expressing tradeoffs between entropy and monotonicity can be achieved by introducing an additional parameter to the total-ambiguity-score.

$$T_{a_1,a_2,\ldots,a_v} = E_{a_1,a_2,\ldots,a_v} + RA_{a_1,a_2,\ldots,a_v}$$

The parameter R expresses the relative importance of monotonicity relative to inductive accuracy in a given problem. When $R = 0$, the total-ambiguity-score uses only its E-score component as the hill-climbing guide. If R has a very high value, monotonicity considerations dominate the building of the decision tree. Several iterations with different values of R on a sample of the training set may be helpful for determining an appropriate value for R.

To illustrate how the proposed metric works, we apply now the ID3 algorithm to the data of Table 1, using the total-ambiguity-score instead of the E-score as the attribute selection metric. We choose $R = 2$, to express the relative importance of monotonicity versus ambiguity, and trace the computation:

$$\begin{aligned} E_{\text{income}} &= 0.667 \text{ bits} \\ I_{\text{income}} &= A_{\text{income}} = 0 \\ T_{\text{income}} &= 0.667\ (0.667 + 2 * 0) \end{aligned}$$

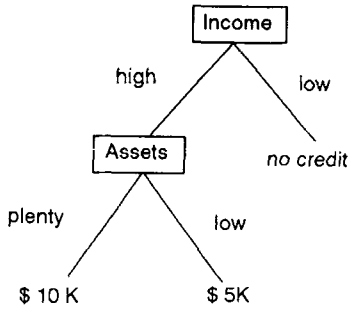

Figure 2. MID's decision tree.

$$
\begin{aligned}
E_{\text{assets}} &= 0 \text{ bits} \\
I_{\text{assets}} &= 0.333\ (2/(3^2-3)) \\
A_{\text{assets}} &= 0.630\ (-\log_2 0.333)^{-1} \\
T_{\text{assets}} &= 1.260\ (0+2*0.630)
\end{aligned}
$$

$$
\begin{aligned}
E_{\text{credit history}} &= 0.667 \text{ bits} \\
I_{\text{credit history}} &= A_{\text{credit history}} = 0 \\
T_{\text{credit history}} &= 0.667.
\end{aligned}
$$

The total-ambiguity-scores of *income* and *credit history* are the lowest, and we assume that the attribute *income* is selected as the first attribute-test.

The total-ambiguity-score of *income* + *assets* is checked now:

$$
\begin{aligned}
E_{\text{income+assets}} &= 0 \text{ bits} \\
I_{\text{income+assets}} &= A_{\text{income+assets}} = 0 \\
T_{\text{income+assets}} &= 0.
\end{aligned}
$$

Unlike the decision tree that has been obtained earlier by applying ID3 with the E-score metric on the same training set, the decision tree that results of applying ID3 with the total-ambiguity-score is monotonic, and its E-score vanishes as well. Figure 2 shows the resulting decision tree. In order to distinguish between the original version of ID3 that uses the E-score and the monotonicity-oriented version of ID3 that uses the total-ambiguity-score as its metric, the latter will be called here MID.

The additional computation of the total-ambiguity-score, relative to ID3's E-score, stems from the monotonicity checks, and the calculation of the order-ambiguity-scores. It can be shown that in the worst case, the number of monotonicity checks is $O(d^2n^2)$, where d denotes the number of attributes, and n is the number of examples in the training set. The number of order-ambiguity-score calculations is identical to the number of the E-score calculations in ID3.

An empirical evaluation of the effectiveness of the total-ambiguity-score metric is given in the next section.

4 Empirical results

Two key questions that arise with respect to the total-ambiguity-score metric are examined here using five real-world MOC problems:

Table 2. Details of the data sets.

	MOD	ERA	EFE	ESL	LEV
No. of attributes	5	4	8	4	4
Number of possible values:					
Attributes	6	7	5	10	5
Classes	2	7	5	10	5
Number of examples	121	125	124	122	125

Table 3. Main results.

Method/Domain		MOD	ERA	EFE	ESL	LEV
ID3:	MAE	0.244	1.002	0.638	0.641	0.696
	Non-monotonicity (%)	14.757	8.230	9.389	3.056	4.246
MID ($R = 1$):	MAE	0.245	0.992	0.670	0.638	0.701
	Non-monotonicity (%)	14.089	7.403	8.591	2.694	4.002
MID ($R = 10$):	MAE	0.231	0.998	0.647	0.618	0.670
	Non-monotonicity (%)	13.174	7.144	8.156	2.414	3.401
MID ($R = 100$):	MAE	0.233	0.998	0.668	0.635	0.670
	Non-monotonicity (%)	13.174	7.144	7.771	2.414	3.401
MID ($R = 1000$):	MAE	0.233	0.998	0.668	0.635	0.670
	Non-monotonicity (%)	13.174	7.144	7.771	2.414	3.401
ID3 + OLM:	MAE	0.267	1.410	0.502	0.637	0.752
	Non-monotonicity (%)	0.000	0.000	0.000	0.000	0.000

A. Does the proposed metric succeed in generating decision trees that have significantly lower non-monotonicity indices when compared with decision trees that are generated using the E-score metric?
B. Does the proposed metric bring about any significant loss of classification accuracy relative to the E-score?

We begin by introducing the problem domains.

Moody's Bond Rating (*MOD*): Includes ratings of bonds according to several key financial ratios. The bonds are partitioned to two groups: 'Good' bonds with Moody's AAA, AA, and A ratings, and 'risky' bonds with lower Moody's ratings.

Employee Rejection/Acceptance (*ERA*): The data set includes attributes of hypothetical applicants for a job, and evaluations of Business Administration students regarding their qualifications.

Examination Form Evaluation (*EFE*): Includes attribute values of proposed matriculation examinations, and experts' judgements about their quality.

Employee Selection (*ESL*): This data set includes actual attribute values of applicants for an industrial opening, and judgments of recruiting experts about their qualifications for these jobs.

Lecturers Evaluation (*LEV*): Includes attribute values of hypothetical lecturers, and opinions of Business Administration students about their teaching qualifications.

All the above data sets involved actual human decisions, and they all included non-monotonic examples. All the attribute and class values were integers. The examples for ERA, ESL, and LEV were randomly selected from larger data sets, such that the training

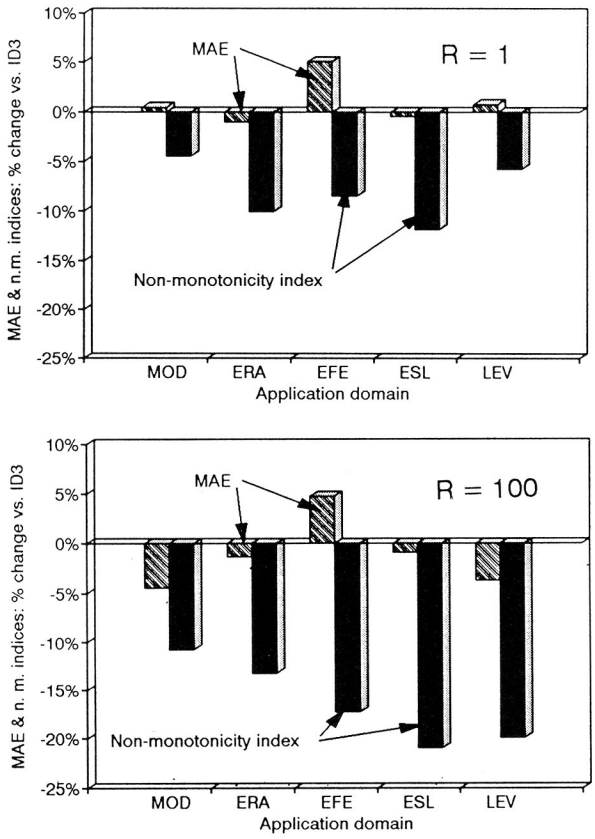

Figure 3. MAEs and non-monotonicity indices; MID ($R = 1, 100$) vs. ID3.

sets and holdout samples had the same size in all the application domains. This strategy was adopted in order to reduce the number of variables in the experiment.

More details about the data sets are shown in Table 2.

The experiment was conducted as follows: Each data set was randomly partitioned into a training set and a holdout sample on a 50%–50% basis. This procedure was repeated ten times (at random) for each data set. The total-ambiguity-score was used by ID3 with four values of R ($R = 1, 10, 100$, and 1000), and the inductive capabilities of the resulting decision trees were tested using the respective holdout samples. The ID3 algorithm was similarly applied using the E-score. The non-monotonic branches of ID3's decision trees were later removed using the OLM. The latter experiment is labeled ID3 + OLM. Table 3 summarizes the main findings. Table 3 shows the mean absolute error (MAE) for the holdout samples (i.e., |predicted class value—true class value|). The non-monotonicity indices of Definition (6) are shown in percents rather than as fractions (i.e., $100 * I$). All the empirical results that are reported in this paper are averages of the 10-fold validation method discussed above. Appendix A details the relevant statistics, as well as the mean square error (MSE).

Figure 3 shows the MAEs of the total-ambiguity-score's decision trees (MID, $R = 1, 100$) relative to those of ID3. The non-monotonicity indices of the former relative to ID3

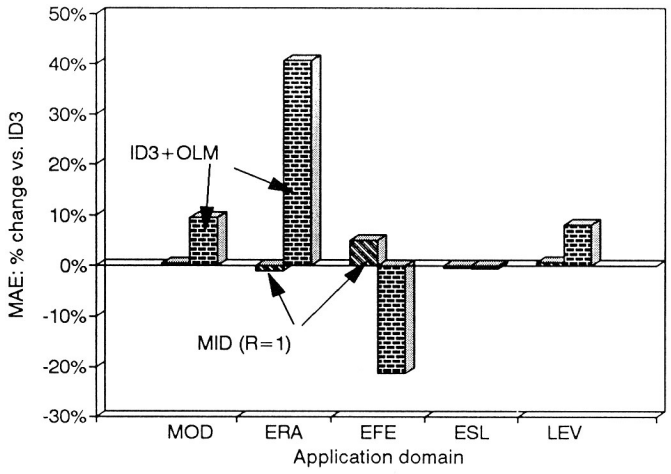

Figure 4. MAEs of ID3 + OLM and MID ($R = 1$) vs. ID3.

are also shown. All the results are expressed as percents relative to ID3. Figure 3 clearly shows that the total-ambiguity-score resulted in decision trees with lower non-monotonicity indices than the E-score. A significant statistical difference (at a confidence interval of 0.95) has been observed between MID's and ID3's non-monotonicity indices in MOD, EFE, ESL, and LEV ($R = 10$, 100, and 1000). This observation also applies to ERA, but within a slightly lower confidence interval. More importantly, no (statistically) significant difference between the MAEs of the two metrics has been observed in any application domain.

The MAEs of ID3 + OLM and MID ($R = 1$) are similarly shown in Fig. 4. In three problem domains (MOD, ERA, and LEV) the MAEs of ID3 + OLM deteriorated relative to the corresponding MAEs of ID3. This deterioration was statistically significant (at a confidence interval of 0.95) in ERA and LEV, and insignificant in MOD. In two application domains (EFE and ESL), the MAEs improved. However, only in EFE was this improvement statistically significant. The observation that ID3 + OLM's MAEs deteriorated in some application domains and improved in other domains (when compared with ID3) is not surprising, since the OLM does not consider the error during its operation.

Figure 5 shows another important property of ID3 + OLM. The variances of ID3 + OLM's MAEs increased substantially (relative to the variances of ID3's MAEs) in two application domains, ERA and ESL. Both these differences were statistically significant. MID's variances of MEAs, on the other hand, were relatively close to those of ID3. This observation is also explained by the fact that the OLM does not consider accuracy during its operation.

5 Conclusions and further research

It has been argued that monotonicity of classifications is a very important consideration while solving MOC problems. Unfortunately, current TDIDT attribute selection measures do not take monotonicity into account. They may result in non-monotonic decision trees, even when all the examples in the training set are monotonic with respect to each other.

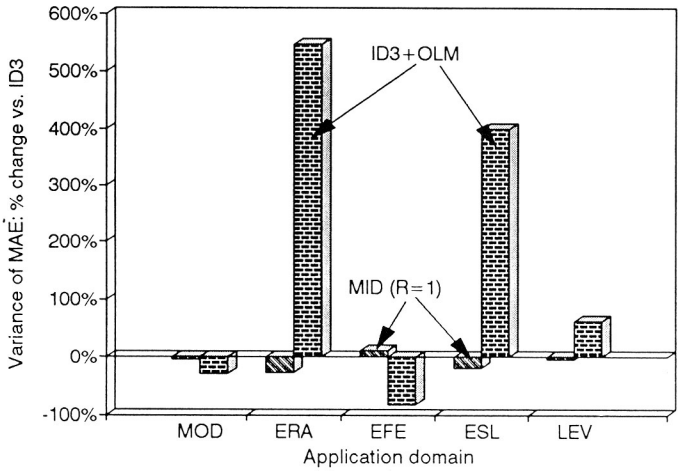

Figure 5. Variances of MAEs; ID3 + OLM and MID ($R = 1$) vs. ID3.

Unlike the E-score, the total-ambiguity-score metric that has been proposed here considers both monotonicity as well as inductive accuracy.

Using five real-world problem domains, it has been shown that the total-ambiguity-score generates decision trees with significantly lower non-monotonicity indices than those that are generated by the E-score metric. More importantly, the former achieves this goal without a significant deterioration of the inductive accuracy (again, when compared with ID3's E-score).

Although the discussion has been focused on extensions to Quinlan's well known E-score attribute selection measure, it is also pertinent to other attribute selection metrics, such as Quinlan's gain-ratio, or Mantaras's distance-based attribute selection metric. TDIDT algorithms that were not studied here, such as C4.5 (Quinlan, 1987), ID4 (Schlimmer & Fisher, 1986), and ID5R (Utgoff, 1989), may also be adapted to deal with MOC problems using a similar approach.

Since MOC problems are so common in human daily life, it is worthwhile to address some interesting open questions in future research: For example, the order-ambiguity score, as defined here, does not take into account the severity of non-monotonic conflicts. A measure that considers the severity of these conflicts may be helpful for some applications. The effects of windowing on the performance of the proposed metric are also of interest. Also, it is worthwhile to investigate whether the inclusion of the order within the attributes and classes in background knowledge can provide better results than those that were obtained via the total-ambiguity-score metric.

Appendix A

General explanations

Appendix A shows detailed results by application. The statistic T tests the hypothesis that the average value shown to its left significantly differs from the one obtained via ID3.

Table A1. MOD.

Method	Avg	T	Var	F
MSE:				
ID3	0.244		0.002	
MID ($R = 1$)	0.245	0.406	0.002	0.98
MID ($R = 10$)	0.231	1.008	0.002	0.91
MID ($R = 100$)	0.233	0.877	0.002	0.82
MID ($R = 1000$)	0.233	0.877	0.002	0.82
ID3 + OLM	0.267	1.091	0.001	0.70
MAE:				
ID3	0.244		0.002	
MID ($R = 1$)	0.245	0.406	0.002	0.98
MID ($R = 10$)	0.231	1.008	0.002	0.91
MID ($R = 100$)	0.233	0.877	0.002	0.82
MID ($R = 1000$)	0.233	0.877	0.002	0.82
ID3 + OLM	0.267	1.091	0.001	0.70
Non-monotonicity (%):				
ID3	14.757		3.613	
MID ($R = 1$)	14.089	1.617	1.517	0.420
MID ($R = 10$)	13.174	2.591	1.621	0.449
MID ($R = 100$)	13.174	2.591	1.621	0.449
MID ($R = 1000$)	13.174	2.591	1.621	0.449
ID3 + OLM	0.000	24.550	0.000	0.000

Table A2. ERA.

Method	Avg	T	Var	F
MSE:				
ID3	1.621		0.048	
MID ($R = 1$)	1.618	0.205	0.038	0.787
MID ($R = 10$)	1.628	0.284	0.045	0.929
MID ($R = 100$)	1.628	0.284	0.045	0.929
MID ($R = 1000$)	1.628	0.284	0.045	0.929
ID3 + OLM	2.957	4.708	0.798	16.501
MAE:				
ID3	1.002		0.009	
MID ($R = 1$)	0.992	1.285	0.007	0.724
MID ($R = 10$)	0.998	0.404	0.007	0.772
MID ($R = 100$)	0.998	0.404	0.007	0.772
MID ($R = 1000$)	0.998	0.404	0.007	0.772
ID3 + OLM	1.410	5.268	0.061	6.444
Non-monotonicity (%):				
ID3	8.230		5.083	
MID ($R = 1$)	7.403	1.456	2.057	0.405
MID ($R = 10$)	7.144	1.738	1.351	0.266
MID ($R = 100$)	7.144	1.738	1.351	0.266
MID ($R = 1000$)	7.144	1.738	1.351	0.266
ID3 + OLM	0.000	11.554	0.000	0.000

Table A3. EFE.

Method	Avg	*T*	Var	*F*
MSE:				
ID3	1.134		0.118	
MID ($R = 1$)	1.182	0.908	0.128	1.092
MID ($R = 10$)	1.104	0.346	0.111	0.941
MID ($R = 100$)	1.138	0.038	0.170	1.443
MID ($R = 1000$)	1.138	0.038	0.170	1.443
ID3 + OLM	0.769	3.805	0.013	0.107
MAE:				
ID3	0.638		0.015	
MID ($R = 1$)	0.670	1.794	0.017	1.112
MID ($R = 10$)	0.647	0.290	0.015	0.981
MID ($R = 100$)	0.668	0.969	0.027	1.764
MID ($R = 1000$)	0.668	0.969	0.027	1.764
ID3 + OLM	0.502	3.987	0.003	0.170
Non-monotonicity (%):				
ID3	9.389		4.466	
MID ($R = 1$)	8.591	1.543	2.412	0.540
MID ($R = 10$)	8.156	2.053	2.974	0.666
MID ($R = 100$)	7.771	2.347	3.084	0.690
MID ($R = 1000$)	7.771	2.347	3.084	0.690
ID3 + OLM	0.000	14.049	0.000	0.000

Table A4. ESL.

Method	Avg	*T*	Var	*F*
MSE:				
ID3	0.902		0.015	
MID ($R = 1$)	0.911	1.174	0.015	1.025
MID ($R = 10$)	0.856	1.325	0.019	1.271
MID ($R = 100$)	0.901	0.016	0.038	2.615
MID ($R = 1000$)	0.901	0.016	0.038	2.615
ID3 + OLM	0.923	0.235	0.102	6.962
MAE:				
ID3	0.641		0.005	
MID ($R = 1$)	0.638	0.389	0.004	0.790
MID ($R = 10$)	0.618	1.038	0.006	1.397
MID ($R = 100$)	0.635	0.271	0.009	1.880
MID ($R = 1000$)	0.635	0.271	0.009	1.880
ID3 + OLM	0.637	0.104	0.023	4.972
Non-monotonicity (%):				
ID3	3.056		0.311	
MID ($R = 1$)	2.694	1.578	0.596	1.915
MID ($R = 10$)	2.414	2.317	0.461	1.483
MID ($R = 100$)	2.414	2.317	0.461	1.483
MID ($R = 1000$)	2.414	2.317	0.461	1.483
ID3 + OLM	0.000	17.324	0.000	0.000

Table A5. LEV.

Method	Avg	T	Var	F
MSE:				
ID3	0.958		0.053	
MID ($R = 1$)	0.984	0.813	0.050	0.940
MID ($R = 10$)	0.914	1.106	0.022	0.405
MID ($R = 100$)	0.914	1.106	0.022	0.405
MID ($R = 1000$)	0.914	1.106	0.022	0.405
ID3 + OLM	1.079	2.101	0.096	1.809
MAE:				
ID3	0.696		0.011	
MID ($R = 1$)	0.701	0.549	0.011	0.949
MID ($R = 10$)	0.670	1.841	0.006	0.547
MID ($R = 100$)	0.670	1.841	0.006	0.547
MID ($R = 1000$)	0.670	1.841	0.006	0.547
ID3 + OLM	0.752	2.252	0.018	1.617
Non-monotonicity (%):				
ID3	4.246		1.926	
MID ($R = 1$)	4.002	1.579	1.845	0.958
MID ($R = 10$)	3.401	2.583	0.772	0.401
MID ($R = 100$)	3.401	2.583	0.772	0.401
MID ($R = 1000$)	3.153	2.299	0.770	0.400
ID3 + OLM	0.000	9.675	0.000	0.000

For example, the MAE of the MOD application domain (see Table A1) is 0.233 for MID ($R = 100$), and the respective MAE of ID3 is 0.244. The T statistic, 0.877, indicates that the difference between these two means is (statistically) insignificant within a confidence interval of 0.95. The statistic F is written similarly. It tests whether the differences between the variances are significant. The comparison is done against the respective variance that was obtained by applying ID3.

Acknowledgments

I am deeply grateful to Sholom Weiss and to the anonymous referees for their helpful comments an earlier draft. This work would not have been possible without the kind assistance of the people and the organizations who contributed the empirical data: Dr. Yoav Ganzach of the Hebrew University of Jerusalem (ERA), Mr. Uri Ben-Shalom of the Israeli Ministry of Education (EFE), and Mr. Eli Fishof, the director of Pilat, Inc. (ESL).

This research was funded, in part, by the Recanati Fund.

References

Ben-David, A., Sterling, L., & Pao, Y.H. (1989). Learning and classification of monotonic ordinal concepts. *Computational Intelligence*, 5(1), 45–49.

Ben-David, A. (1992). Automatic generation of symbolic multiattribute ordinal knowledge-based DSSs: Methodology and applications. *Decision Sciences*, 23(6), 1357–1372.

Carter, C., & Catlett, J. (1987). Assessing credit card applications using machine learning. *IEEE Expert*, Fall 1987, 71–79.

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, 3, 261–283.

Hayes-Roth, F., Waterman, D.A., & Lenat, D.B. (1983). *Building expert systems*, Addison-Wesley.

Jacoby, J., Speller D.E., & Berning, C.K. (1974). Information load: Replication and extension. *Journal of Consumer Research*, 1, 33–42.

Larichev, O.I., Moshkovich, H.M., & Rebrik, S.B. (1988). Systematic research into human behavior in multiattribute object classification problems. *Acta Psychologica*, 68, 171–182.

Mantaras, R.L. (1991). A distance-based attribute selection measure for decision tree induction. *Machine Learning*, 6, 81–92.

Michalski, R.S. (1983). A theory and methodology of inductive learning. In *Machine learning: An artificial intelligence approach*, R.S. Michalski, J. Carbonell, & T. Mitchell, (Eds.), Morgan Kaufmann Publishing Co., CA.

Nunez, M. (1991). The use of background knowledge in decision tree induction. *Machine Learning*, 6, 231–250.

Quinlan, J.R. (1983). Learning efficient classification procedures and their applications to chess endgames. In *Machine learning: An artificial intelligence approach*, R.S. Michalski, J. Carbonell, & T. Mitchell, (Eds.), Tioga Publishing Co., Palo Acto, CA.

Quinlan, J.R. (1986). The effect of noise on concept learning. In *Machine learning: An artificial intelligence approach*, R.S. Michalski, J. Carbonell, & T. Mitchell, (Eds.), Morgan Kaufmann, CA.

Schlimmer, J.C. & Fisher, D.H. (1986). A case study of incremental concept induction. *Proceedings of the Fifth National Conference on Artificial Intelligence*, Morgan Kaufmann, CA, 496–501.

Utgoff, P.E. (1989). Incremental induction of decision trees. *Machine Learning*, 4, 161–186.

Received July 3, 1993
Accepted October 4, 1993
Final Manuscript May 11, 1994