Lower-bound Estimate for Cost-sensitive Decision Trees

Mikhail V. Goubko

Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russian Federation (Tel: +7-495-334-9051; e-mail: mgoubko@mail.ru).

Abstract: While an extensive body of literature investigates problems of decision trees growing, just a few study lower-bound estimates for the expected classification cost of decision trees, especially for varying costs of tests. In this paper a new lower-bound estimate is proposed. Computation of the estimate is reduced to solving a series of set-covering problems. Computational complexity and other properties of the lower-bound estimate are investigated. The top-down algorithm of tree construction based on the proposed estimate is tested against several popular greedy cost-sensitive heuristics on a range of standard data sets from UCI Machine Learning Repository.

Keywords: knowledge-based control, learning control, classification, decision trees.

The research is supported by the grant 10-07-00129 of Russian Foundation for Basic Research.

1. INTRODUCTION

Decision trees persist among the most popular classification tools in machine learning, pattern recognition, fault detection, medical diagnostics, and situational control. Decision trees are widely used because of their simplicity, intuitiveness, ease to use and interpret.

The idea behind decision trees is that a *decision* for a *situation* is made, or a *class* for a *case* is assigned, from a series of measurements (*tests*) of observable *attributes* of the situation while the next attribute to be tested depends on the results of the previous tests. The plan of testing forms a *tree* where *leaves* are labelled with decisions, *internal nodes* are labelled with attribute names, while edges leaving an internal node are labelled with *splits* – mutually exclusive conditions on the values of the attribute being tested at that node.

Decision trees are learned from data. Typical decision tree growing algorithm takes an input of a *learning set* of labelled *examples*, i.e. vectors of attribute values accompanied with the class label, and builds a tree to classify correctly as much examples as possible. Quality of the tree is usually evaluated by *misclassification rate* on the testing set of examples.

Most of the research on decision trees construction is concentrated on growing compact, in some sense, decision trees. Classifications based on small trees are commonly believed to better generalize to new data due to the famous “Occam’s razor principle” of simple hypotheses. Thus, decision tree building becomes an optimization problem.

A number of decision tree optimization settings are known differing in permissible types of attribute values, plausible splits, and measures of the tree size. One of the classical measures is the expected length of the path in a tree.

In many applications (e.g. medical diagnostics, fault detection, etc) tests differ in cost of measuring the value of the attribute. A general blood analysis is much cheaper than a computer tomography procedure. In such a *cost-sensitive* framework a compact tree has another important advantage besides its better generalisation characteristics – it is cheaper in operation as only required tests are performed in each case. The expected cost of classification seems to be a natural optimization criterion in this framework given the decision tree correctly classifies available examples (has zero misclassification rate on the learning set). Only cost of tests is used in this paper. Other types of costs relevant to decision tree growing (*misclassification costs*, *costs of teaching*, *intervention costs*, etc) are discussed in Turney (2000).

Growing an optimal decision tree is a discrete optimization problem. Hyafil and Rivest (1976), and also Zantema and Bodlaender (2000) have shown this problem to be NP-hard. Moreover, Sieling (2008) has shown that the size of an optimal tree is hard to approximate up to any constant factor. For this reason numerous heuristic algorithms were suggested during several recent decades of finding near-optimal decision trees in different settings. Most of them employ greedy top-down tree induction. An attribute and a split for every tree node are chosen basing on the sort of information gain criterion originated to Quinlan (1979). Initially developed for equal test costs, the criterion based on information gain was then refined for cost-sensitive decision trees in a family of algorithms such as IDX (Norton, 1989), EG2 (Núñez, 1991), CS-ID3 (Tan, 1993), and many others.

Numerous experiments show good performance of these heuristics, but in any real situation the question remains open of how much extra cost is due to imperfectness of a heuristic tree growing algorithm – is it worth improving the search by looking for more sophisticated search techniques, or losses are already admissible to stop. Check of that sort is most interesting for the problems where test costs are measured in money units and are high enough.

As exact optimal tree cost is hard to compute, its cost should be approximated from below. If the lower-bound estimate exists for the cost of the optimal decision tree the answer to the above question can be obtained in the form “no more than $*X* can be economized by further improvement of a currently calculated decision tree”. Lower-bound estimates known from the literature have some limitations explained below and are hardly applicable to this problem.

In this paper a new lower-bound estimate for the expected classification cost of the optimal tree is suggested. Its calculation reduces to the solution of a number of combinatorial set-covering problems (or their linear programming relaxations). Although being NP-hard, these problems are shown to be computed efficiently in a series of experiments over real classification tasks.

The rest of the paper is organized as follows. Section 2 describes the model. The literature is reviewed in Section 3. Section 4 introduces the lower-bound estimate and Section 5 examines its computational complexity; Section 6 discusses some applications of the estimate, while Section 7 concludes.

2. THE MODEL

Consider a set of decisions (or classes) $D = \{1, \dots, d\}$, a typical class dented by *f*, and a set of attributes $M = \{1, \dots, m\}$. A typical attribute from *M* is denoted by *q*. Only categorical attributes are considered, and $k(q)$ is the cardinality of the set of values of attribute *q*.

The learning set of examples (also referred to as *cases*) *N* is also given, a typical example denoted by *w*. Example $w \in N$ is a unique vector $(a_{wq})_{q \in M}$ of attribute values, and a class label $f(w) \in D$. No noise is expected in the learning set, i.e. $f(w)$ represents a correct decision for all situations *w*. An example is also endowed with a positive number $\mu(w)$ – the probability, or frequency, of the example.

Every attribute *q* gives rise to the test or the *question* of the form “what is the value of attribute *q*?”. Different answers partition the whole set of examples *N* into the sets $S_1(q), \dots, S_{k(q)}(q)$ (some sets may be empty).

Testing different attributes incurs different *costs*. Turney (2000) distinguishes test costs depending on:

1) the true class of the case, e.g. the (expected) cost of an exercise stress test conditional on whether the patient has heart decease;
2) side-effects (the value of some hidden attribute), e.g. allergic reactions on some radiological tests;
3) an individual case;
4) prior tests performed, e.g. “group discounts” or common costs for a series of blood tests sharing the cost of collecting blood;
5) prior tests results (other attributes’ values), e.g. the cost of blood test depending on the patient’s health insurance plan.

Case-dependent test costs immediately cover the first three categories. The last two categories are reduced to case-dependent test costs by adding virtual tests that combine related questions. Also case-dependent costs cover a concealed important dependence on the correct answer of the current question. Below only case-dependent test costs t_{qw}, where $q \in M$, and $w \in N$, are considered.

Different costs t_q of tests are somewhat typical for the literature on cost-sensitive trees growing, while more complex dependencies require come clarification. Indeed, the goal of classification is to elicit a class for a certain case from the tests of its attributes, and if the cost is observable during testing, some extra information about the case (besides the answer itself) is obtained from the test. But if costs are interpreted as duration of the test, they are subject to noise that obscures the expected value t_{qw} in a single experiment. Mentioned above virtual combined tests represent another situation where case-dependent costs arise naturally.

In this paper the problem of *exact classification* is solved, thus, misclassification costs are not considered. Also, costs of testing the cases falling beyond the learning set are of no importance as only accuracy on the learning set is watched.

A decision tree $H = \langle V, E \rangle$ is a directed tree with set of nodes *V* and set of edges *E*. Internal nodes are labelled with tests, while leaves are marked by classes. For categorical attributes the edges can be associated with admissible attribute values (or, equivalently, the clauses of the form “the value of attribute *q* is *X*”).

Decision tree *H classifies* learning set *N* iff for every example $w \in N$ the path exists in the tree from the root to some leaf $v \in V$ (to generalize the tree absent edges are added and new leaves are labeled with the expected class of the parent node). There is no use to test the value of the same attribute more than once within a path, thus decision tree *H* induces a binary $m \times n$ matrix, with element e_{qw} being equal to one if test *q* belongs to the path in tree *H* for example *w*, and zero otherwise. The total cost of tests for case *w* is then written as $T(w,H) := \sum_{q \in M} t_{qw} e_{qw}$. The expected cost of classification for decision tree *H* is then obtained by averaging the costs over the training set *N*:

$$T(H) := \sum_{w \in N} \mu(w) \cdot T(w,H) = \sum_{w \in N} \mu(w) \sum_{q \in M} t_{qw} e_{qw} \; . \quad (1)$$

Decision tree *H classifies correctly* learning set *N* iff *H* classifies *N*, and cases *w*' and *w*'' share the same path in tree *H* only when they belong to the same class, i.e. $f(w') = f(w'')$.

The *problem of decision tree induction* is then to find the tree that classifies correctly learning set *N* and minimizes cost (1).

3. BRIEF LITERATURE REVIEW

The literature on growing decision trees is huge. Below only lower-bound estimates of decision tree test cost are reviewed.

Consider every test has at most $k > 1$ different outcomes and $t_{wq} = 1$ for all *w* and *q*. As *d* classes must be distinguished at leaf nodes of a decision tree, an obvious lower-bound estimate for the size of the tree is $(d-1)/(k-1)$ – the number of internal nodes in any *k*-fold tree with *d* leaves.

Using an analogy with the prefix coding problem, information theory says that the expected path length in a decision tree correctly classifying the whole learning set is bounded from below by *k*-ary entropy of the class variable (treated as a stochastic value):

$$H(D) := -\sum_{f\in D} p_f \log_k p_f \text{ , where } p_f := \sum_{\substack{w\in N:\\ f=f(w)}} \mu(w) . \qquad (2)$$

In Gubko (2008) both formulae appear as special cases of a general lower-bound estimate of the cost of the tree for a family of *homogenous cost functions* (the first – for the zero *degree of homogeneity*, the second – for the degree of unity).

Most of the following work on lower-bound estimates for decision trees also exploits the idea of entropy. Ohta and Kanaya (1991) deduce an entropy-based lower-bound estimate for the cost of an arbitrary decision tree by calculating the total cost of the tree as a weighted sum of classification costs and misclassification losses. They assume that $N = D$, all attributes have the same cardinality k, and are statistically independent.

Expected Huffman prefix code length $h(D)$ gives a slightly tighter lower bound of classification costs (Parkhomenko (2010) explains building non-binary Huffman trees). Biasizzo, Žužek, and Novak (1998) introduce a cost-sensitive version of the lower-bound estimate. They consider binary tests only. If each test $q \in M$ has an individual cost t_q, the expected cost of tests is shown to be not less than the sum of $\lfloor h(D) \rfloor$ cheapest tests' costs with an add-on of $h(D) - \lfloor h(D) \rfloor$ fraction of the cost of the next-cheapest test ($\lfloor\ \rfloor$ stands for an integer part of a number). It is notable that a similar idea of a cost-sensitive lower-bound estimate first appeared in Martin (1971), but its reasoning was poor there.

Although well-grounded and popular, entropy-based lower bounds share a common limitation. They work fine when the number of classes is large compared to the whole number of examples ($N = D$ is the best), and become too optimistic otherwise: for instance, in a binary decision problem (win or lose, valid or broken, etc) the length $h(2)$ of a code word is equal to unity irrespective of tests availability and the number of examples. Also attributes' cardinality variations are not accounted – either binary tests are assumed, or maximum cardinality is spread among all attributes. The estimate of Biasizzo-Žužek-Novak (1998) is also sensitive to cheap tests with poor information gain. These tests are never found in near-optimal trees but tangibly force down the estimate.

Bessiere, Hebrard, and O'Sullivan (2009) adhere to a quite different approach that is very similar in spirit to that adopted in this paper. The problem of optimal decision tree construction is viewed as a purely combinatorial optimization problem. The size of the tree is minimized. For every pair of examples from different classes the set of *discrepancies* is computed – the set of attributes with differing values. The minimal set of tests hitting the family of all discrepancies generated by the learning set is proved to be a lower-bound estimate of the size of the decision tree that correctly classifies this learning set (note that the minimum hitting set problem is equivalent to the set-covering problem that arises below in calculation of the lower-bound estimate). At the same time, authors limit attention to cost-insensitive binary decision problem, and use the size of the tree as an optimization criterion.

Thus, the objective of this paper is to suggest the lower-bound estimate for the cost of decision tree with case-dependent test costs. The estimate must perform well in (most interesting) situations when the number of classes is small compared to the number of examples. Also it must be less sensitive to the presence of cheap "dummy" tests.

4. LOWER-BOUND ESTIMATE

Definition 1. A subset of tests $Q \subseteq M$ *isolates* (or *classifies correctly*) case w in subset of cases $S \subseteq N$ ($w \in S$) iff sequence of tests Q assures proper decision $f(w)$ given initial uncertainty S and w is the real state of the world.

With no loss of generality suppose that $w \in S_1(q)$ for all $q \in M$. Then Q isolates w in S iff $f(w') = f(w)$ for all $w' \in \bigcap_{q\in Q} S_1(q) \bigcap S$. Note that the set of all questions M isolates any case w in N. This property assures that at least one decision tree always exists.

Definition 2. Optimal set of questions $Q(w, S) \subseteq M$ is the cheapest of the sets of questions that isolate case w in set S; Define also the *minimum cost* $t(w,S) := \sum_{w\in Q(w,S)} t_{wq}$.

By definition, cost $t(w, S)$ is the minimum cost to assure proper classification $f(w)$ given initial uncertainty S about the state of the world and w being the true state. A decision tree that classifies set N correctly induces some sequence of questions for every example $w \in N$. Obviously, this sequence isolates w in N. Thus, cost $t(w, N)$ is a lower-bound of classification cost of example w in any decision tree H that classifies set N correctly. Consequently, expected cost $T(H)$ never falls below the average minimum cost

$$T_l(N) := \sum_{w\in N} \mu(w) \cdot t(w,N) = \sum_{w\in N} \mu(w) \sum_{q\in Q(w,N)} t_{qw} \text{ ,} \qquad (3)$$

and $T_l(N)$ is the *lower-bound estimate* of the cost of the optimal tree that correctly classifies set N by virtue of family of tests M. Set of available tests M is not included in the list of arguments as for any set of cases S unrecognized after a series of tests $Q \subseteq M$ $T_l(N, M\backslash Q) = T_l(N, M)$. Only such sets of cases are considered below.

This lower-bound estimate is based on substituting the solution of the initial problem with the solution of a simpler problem. The initial problem of unknown case classification is replaced by the problem of proving the true case to a third party. Imagine you know the true case w, but your colleague does not. You prove the true case is really w by performing some available tests from M. To achieve the goal at minimum cost you should choose the tests from $Q(w, N)$. Expected cost of proof then equals exactly $T_l(N)$. It is obviously easier to classify when you know the result beforehand, and this also proves that $T_l(N)$ is the lower-bound estimate.

Definition 3. Test $q \in M$ is *essential* for set $S \subseteq N$ iff $q \in Q(w, S)$ for all $w \in S$.

At least one essential test is required for the lower-bound estimate to be reached. Moreover, if essential test q is chosen for the root of a decision tree, for the lower-bound to be reached there must exist at least one essential test for every set $S_1(q), \ldots, S_{k(q)}(q)$, and so on up to leaves of a tree. Although possible, this seems to be a very rare situation.

In some special cases the quality of the lower-bound estimate can be sufficiently low. The quality of the lower-bound estimate – the ratio $T_l(N)/T(H)$ – is proven to be at least $2/(n + 1)$ for case-insensitive tests. The value of $2/n$ can be approximated arbitrary close by the following setting. Consider learning set $N = \{1, \ldots, n\}$ of equally probable examples, set of classes $D = N$, and set M consisting of $n + 1$ questions: questions $q = 1, \ldots, n$ of the cost $2 + \varepsilon$ (where ε is a small positive constant) taking a form "is it case w or not?" for each $w \in N$, and question $q = n + 1$ of the cost n, which immediately distinguishes all cases in N. Then $Q(w, N) = \{w\}$ for every $w \in N$, and the lower-bound estimate $T_l(N) = 2 + \varepsilon$, while optimal tree H consists of the sole test $n + 1$, and has the cost $T(H) = n$. Thus, the ratio $T_l(N)/T(H) = (2 + \varepsilon)/n$ can be made arbitrary close to $2/n$. The reason, why this setting results in poor quality, is that each case w is considered separately while calculating the lower-bound estimate and all sets of tests isolating w except the cheapest one are ignored.

At the same time, in contrast to information-theory based lower-bound estimates, the proposed estimate cares for the tests availability and is applicable in situations when the number of classes is small.

The lower-bound estimate can be made tighter at the cost of m-fold computational complexity increase, as suggested in Ohta and Kanaya (1991). Every subtree of an optimal tree is also optimal, and any tree must have a test in its root. So,

$$T_l^*(N) := \min_{q\in M}\left[\sum_{w\in N}\mu(w)\cdot t_{wq} + \sum_{i=1}^{k(q)} T_l(S_i(q)) \sum_{w\in S_i(q)} \mu(w)\right] \qquad (4)$$

is a lower-bound estimate for the cost of tests of the decision tree, and $T_l^*(H)$ is never less than $T_l(H)$.

5. LOWER-BOUND ESTIMATE CALCULATION

Calculation of the lower-bound estimate $T_l(N)$ reduces to computing the optimal set of questions for all n cases. Consider a case $w \in N$ and suppose with no loss of generality that $w \in S_1(q)$ for all $q \in M$. Let $F(w) \subseteq N$ be the set of cases that share the same class with case w. Then the problem of finding $Q(w, S)$, and, consequently, $t(w, S)$, for some $S \subseteq N$ is equivalent to the problem of covering set $S \backslash F(w)$ by the family of sets $\{N \backslash S_1(q)\}_{q \in M}$, or an integer program:

Choose a binary vector $(x_q)_{q \in M}$ *to minimize* $\sum_{q\in M} t_{wq} x_q$ *given* $\sum_{q\in M} a_{qw'} x_q \geq 1$ *for all* $w' \in S \backslash F(w)$, *where* $a_{qw'}$ *is equal to zero if* $w' \in S_1(q)$, *and unity otherwise.* (5)

The set-covering problem is one of the most studied integer optimization problems (see Caprara et al (2000)). It is known to be NP-hard (and hard to approximate up to any constant factor). Nevertheless, several algorithms are tested below for the average computation time on the real classification problem.

Data to cover different problem dimensions (n and m) were generated from "Chess" data set at UCI Machine Learning Repository (archive.ics.uci.edu/ml). The data set classifies chess KRKPA7 end-games (King+Rook vs King+Pawn on a7). 3196 cases are split in two classes (1669 "won", and 1527 "nowin") by the values of one ternary and 35 binary attributes.

The initial data set was restricted to randomly chosen 9, 18, and 24 attributes by taking the expected class label. Then attributes were randomly joined to form combined tests of different cardinality (from 2 to 768), while the number of questions varied from 2 to 36. Case-sensitive test costs were randomly picked from the uniform distribution over [0, 1]. Random subset S of cases was picked and the minimum cost $t(w, S)$ was computed for 10 randomly selected cases $w \in S$ by a general-purpose branch-and-bound binary programming algorithm, which uses a continuous relaxation of integer subprograms to limit search. A group of 65000 experiments was run. See Table 1 for results.

Then integer program (5) was *relaxed* to a linear program permitting all non-negative x_q to fasten computation of the minimum cost. The solution $x^*(w,S) = (x_q^*(w,S))_{q\in M}$ of the relaxed problem gives a lower-bound estimate $t_L(w, S)$ of the minimum cost $t(w, S)$. An expected value of $t_L(w, S)$ then serves as a relaxed lower-bound estimate of the cost of correct classification of learning set N:

$$T_L(N) := \sum_{w\in N} \mu(w)\cdot t_L(w,N)\,. \qquad (6)$$

An adjusted estimate $T_L^*(N)$ is defined by analogy with (4). Lower-bound estimate $T_L(\cdot)$ obviously never exceeds $T_l(\cdot)$, but experiments show a minor fall of quality (at most 26% and 3.8% in the mean). At the same time, the computation speed gain is also minor (compare columns 1 and 2 of Table 1). The iterative active-set method is used to solve linear programs.

Table 1. Cover-set problem experiments

Algorithm	1. Binary program	2. Linear relaxation	3. Dual program
Average optimal cost	3.122	3.003	3.003
Avg. comp. time, *s*	0.1866	0.1713	0.0642
95% conf. interval for the comp. time linear regression slope*, *ns*	9.41-9.58	8.14-8.17	2.84-2.86
R^2 for the linear regression	0.98	0.97	0.85

* For Intel® Core Duo™ T7200 2 GHz.

The experiments also show that replacing a linear program with its dual one fastens computation approximately three-fold (the reason is that the number of variables m is small and the number of conditions $|S|$ is comparably large in an initial problem) – compare the slopes in Table 1. Computation time for all three algorithms exhibits a linear relation on the size $m\cdot|S|$ of the constraints matrix $A = (a_{qw})$ at high confidence level. In neither experiment a significant deviation of the computation time from the linear relation is observed.

Thus, the average computation time of the test cost' lower-bound estimate for the decision tree that classifies correctly n examples with m available tests is proportional to $n^2\cdot m$. Calculation is easily parallelized.

In the next section the lower-bound estimate is also computed for subtrees classifying some set $S \subseteq N$. If S is the set of examples undistinguished after running a series of tests $Q \subseteq M$, these questions can be excluded to fasten the calculation as they add no information and are never found in optimal sets of questions isolating a case in set S.

6. LOWER-BOUND ESTIMATE APPLICATIONS

The main goal of the new lower-bound estimate is to overcome the shortcomings of the existing estimates. Evaluation of losses of a particular heuristic decision tree is the main application. From the practical point of view it gives a rationale to accept a tree or to seek for the further improvements of the decision tree.

Lower-bound estimates are often used in branch-and-bound algorithms, but estimates (3) and (6) are too costly in calculation for such an application.

In the standard greedy algorithm of top-down induction (TDI) of the decision tree the test is chosen for the node that maximizes or minimizes a *split criterion*. A lower-bound estimate can be used to build a variant of the split criterion:

$$P(q,S) := \sum_{w \in S} \mu(w) \cdot t_{wq} + \sum_{i=1}^{k(q)} T_L(S \cap S_i(q)) \sum_{w \in S \cap S_i(q)} \mu(w) . \qquad (7)$$

The test that minimizes (7) is chosen for the node with set S of unclassified examples. Below this algorithm is referenced to as TDI+LB. Although the number of lower-bound estimate calculations is much fewer compared to that in a typical run of a branch-and-bound algorithm, TDI+LB is still too slow compared to simple cost-sensitive TDI heuristics (such as CS-ID3, IDX, and EG2), while giving doubtful gain in quality. At the same time, as the combinatorial nature of split criterion (7) crucially differs from the information-theoretic nature of the criteria used in CS-ID3, IDX, and EG2, it is interesting to compare the trees generated by these algorithms. Adjacency of the resulting trees is a good reason to believe the heuristic tree is a near-optimal one.

Note also that calculation of lower-bound estimates during TDI can be fastened sufficiently by reusing the results of previous calculations. Consider any test q from the optimal set of questions $Q(w, N)$ of some case w, or any test q for which $x^*_q(w, N) = 1$. Then it is easy to show that for all $S_i(q)$, $i = 1, \ldots, k(q)$, the equalities hold for all $q' \neq q$: $Q(w, S_i(q)) = Q(w, N)\backslash\{q\}$, and $x^*_{q'}(w, S_i(q)) = x^*_{q'}(w, N)$. That is recalculation of the minimum cost of isolating a case is required at a child node only when a non-optimal question is asked at the parent node.

A series of experiments was performed to compare TDI+LB algorithm with the other TDI heuristics. Another aim of the experiments was to check the quality of the lower-bound estimate (6) for real classification problems. Standard data sets MONK-1, MONK-2, "Cars", and "Chess" from UCI Machine Learning Repository were used. The number of classes never exceeds four in these classification problems, so the quality of entropy-based lower-bound estimates is extremely low. Two types of test costs were used in experiments. Test-sensitive costs t_q were picked from the uniform distribution over [0, 1]. Case-sensitive costs t_{qw} were calculated by adding a uniformly distributed on [0, 0.5] noise to the test-sensitive costs t_q. When calculating the split criterion for IDX, CS-ID3, and EG2, case-sensitive costs were averaged out over the set of unrecognized cases.

The results of the experiments are depicted in Table 2. Top five rows describe the data sets and the experiments. Then the average (over experiments) value is presented for the lower-bound estimate $T_L(N)$ (LB) and costs of decision trees generated by IDX, CS-ID3, and EG2 algorithms. Below the average cost of the tree generated by TDI+LB algorithm based on split criterion (7) is presented, and quality of TDI+LB is depicted in terms of the number of experiments where TDI+LB outperforms ("wins") all other algorithms, is dominated by one of them ("loses"), or leads to the same tree as the best of the other tested algorithms (a "draw"). The split criterion for EG2 algorithm has a parameter ω of the "strength of the bias towards lower cost attributes". In every experiment the bias varied from zero to 10 to obtain the best quality of EG2.

Table 2. Greedy heuristics comparison

Data set		MONK-1		MONK-2		Cars		Chess	
No of attr., m		6		6		6		36	
No of cases, n		122		169		1728		3196	
No of classes, d		2		2		4		2	
Cost type		t_q	t_{wq}	t_q	t_{wq}	t_q	t_{wq}	t_q	t_{wq}
No of trials		100	100	100	100	100	100	2	-
Avg. cost	**LB**	1.023	1.571	1.537	2.317	1.218	1.849	1.762	-
	IDX	1.404	2.188	2.030	3,138	1.451	2.221	3.246	-
	CS-ID3	1.463	2.293	2.086	3.172	1.444	2.227	3.283	-
	EG2	1.303	2.021	1.985	3.073	1.438	2.215	3.174	-
	TDI+LB	1.258	1.943	1.941	2.998	1.468	2.255	3.318	-
TDI+LB wins		63	55	95	96	20	28	1	-
TDI+LB loses		29	31	5	4	72	60	1	-
Draw		8	14	0	0	8	12	0	-

Table 2 shows that TDI+LB algorithm based on the lower-bound estimate outperforms the other algorithms on small data sets (MONK-1 and MONK-2) both in terms of the average cost and of the number of wins, while clearly losing on bigger data sets (Cars and Chess).

It is interesting to note that addition of a case-sensitive noise to costs of tests does not benefit neither the quality of the lower-bound estimate, nor the quality of TDI+LB heuristic. The changes are immaterial compared to the setting with test-sensitive costs. In all experiments the trees generated by all algorithms are notably similar, so these trees seem to be near-optimal. Under this hypothesis, the quality of the lower-bound estimate $T_L(\cdot)$ varies from the experiment to the experiment in a range from 50% to 90%.

The experiments show that "statistical" heuristics work much better starting from the hundreds of classified examples. It is expectable, as the proposed lower-bound estimate is based on the sort of "micro-description" of the classification problem.

To boost both computation speed and quality of TDI+LB, it can be combined with some information-gain based heuristic (say, EG2) replacing it when inducing the lower-level subtrees for no more than some number of examples (the boundary is determined by a constant *threshold*). Tables 3 and 4 below show that this sort of combination (denoted in the table as EG2+LB) works well for "Cars" data set and worse for "Chess" problem. Comparing the results for "Chess" with that of "Cars" shows that the lower-bound estimates (3) and (6) work considerably worse in the presence of the large number of "dummy" tests never met in near-optimal decision trees (the same issue was noticed for the cost-sensitive lower-bound estimate proposed by Biasizzo, Žužek, and Novak (1998)).

Table 3. Comparison of EG2 and EG2+LB (Cars)

Threshold		100		50		25	
Cost type		t_q	t_{wq}	t_q	t_{wq}	t_q	t_{wq}
No of trials		100	100	100	100	100	100
Avg. cost	EG2	1.412	2.186	1.492	2.219	1.482	2.247
	EG2+LB	1.425	2.192	1.489	2.217	1.481	2.246
EG2+LB wins		45	53	67	65	24	41
EG2 wins		43	27	13	5	0	0
Draw		12	20	20	30	76	59

Table 4. Comparison of EG2 and EG2+LB (Chess)

Threshold		50		25		10		5	
Cost type		t_q	t_{wq}	t_q	t_{wq}	t_q	t_{wq}	t_q	t_{wq}
No of trials		10	10	10	10	10	10	10	10
Avg. cost	EG2	1.877	3.120	2.050	2.935	2.034	3.213	2.0501	2.9349
	EG2+LB	1.901	3.128	2.057	2.942	2.036	3.217	2.0507	2.9352
EG2+LB wins		1	4	1	0	2	0	3	1
EG2 wins		9	6	9	10	7	10	7	5
Draw		0	0	0	0	1	0	0	4

7. CONCLUSION

In this paper the new lower-bound estimate is developed for the cost of the decision tree with case-dependent test costs. Unlike known estimates it performs well when the number of classes is small compared to the number of examples.

Calculation of the estimate is NP-hard in the worst case but the experiments show admissible average performance in the order of $n^2 \cdot m$ operations for *n* examples and *m* tests.

The main application of the proposed estimate is evaluation of losses of a particular heuristic decision tree algorithm. But it can also be used in split criteria of greedy algorithms of decision tree construction. Experiments on four real data sets show that these algorithms give results comparable with popular cost-sensitive heuristics – IDX, CS-ID3, EG2, and perform better on small data sets with lack of tests.

REFERENCES

Bessier, C., Hebrard, E., O'Sullivan, B. (2009). Minimising decision tree size as combinatorial optimisation. *Proc. of the 15th international conference on principles and practice of constraint programming*, 173–187.

Biasizzo, A., Žužek, A., Novak, F. (1998). Sequential diagnosis with asymmetrical tests. *The Computer Journal*, 41(3), 163–170.

Caprara, A., Toth, P., Fischetti, M. (2000). Algorithms for the Set Covering Problem. *Annals of Operations Research*, 98(1), 353– 371.

Garofalakis, M., Hyun, D. Rastogi, R., Shim, K. (2000). Efficient algorithms for constructing decision trees with constraints. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 335–339.

Gubko M.V. (2008). The search for optimal organizational hierarchies with homogeneous manager cost functions. *Automation and Remote Control*. 69(1), 89–104.

Hyafil, L., Rivest, R. (1976). Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5(1), 15–17.

Martin, W.A. (1971). *Construction of optimal sequential decision trees.* Working paper 528/71 of A. P. Sloan School of Management, MIT.

Norton, S.W. (1989). Generating better decision trees. *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, *IJCAI-89*, 800–805.

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

Ohta, S., Kanaya, F. (1991). Optimal decision tree design based on information theoretical cost bound. *IEICE Transactions*, E 74(9), 2523–2530.

Parkhomenko P.P. (2010). Questionnaires and organizational hierarchies. *Automation and Remote Control*, 71(6), 1124–1134.

Quinlan, J.R. (1979). Discovering rules by induction from large collections of examples. In D. Michie (Eds.) *Expert systems in the microelectronic age*, 168–201. Edinburgh University Press, Edinburgh.

Sieling, D. (2008) Minimization of decision trees is hard to approximate. *Journal of Computer and System Sciences*, v.74 No. 3, p.394–403.

Tan, M. (1993). Cost-sensitive learning of classification knowledge and its applications in robotics. *Machine Learning*, 13, 7–33.

Turney, P. (2000) Types of cost in inductive concept learning. *Workshop on Cost-Sensitive Learning at ICML*, 15–21.

Zantema, H., Bodlaender, H.L. (2000). Finding small equivalent decision trees is hard. *International Journal of Foundations of Computer Science*, 11(2), 343–354.