

Correction to: Trading Off Worst and Expected Cost in Decision Tree Problems

Aline Saettler¹  · Eduardo Laber¹ · Ferdinando Cicalese²

Received: 11 October 2017 / Accepted: 26 February 2018 / Published online: 20 March 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Correction to: Algorithmica (2017) 79:886–908
<https://doi.org/10.1007/s00453-016-0211-2>

Abstract This erratum fixes a technical problem in the paper published in Algorithmica, Volume 79, Number 3, November 2017, pp. 886–908. Theorem 1 of this paper gives upper bounds on both worst testing cost and expected testing cost of the decision tree built by Algorithm 1. Although the statement is correct, the proof presented in the paper has a problem. The proof relies on the analysis of a nonlinear program (NLP) given by Eqs. (5)–(9), which is not convex as mistakenly proved in Appendix A.2. In this erratum we present a correct proof of Theorem 1. Instead of analyzing the NLP we analyze a related linear program.

1 Notation and Algorithm 1

For the sake of self containment we present some notations and the description of Algorithm 1 introduced in [1].

The original article can be found online at <https://doi.org/10.1007/s00453-016-0211-2>.

✉ Aline Saettler
asaettler@inf.puc-rio.br

Eduardo Laber
laber@inf.puc-rio.br

Ferdinando Cicalese
ferdinando.cicalese@univr.it

¹ Departamento de Informática, PUC-Rio, Rio de Janeiro, Brazil

² Dipartimento di Informatica, Università di Verona, Verona, Italy

We define $cost(D, s)$ as the sum of the tests' cost on the path from the root of D to the leaf associated with object s . Then, the *worst testing cost* and the *expected testing cost* of D are, respectively, defined as

$$cost_W(D) = \max_{s \in S} \{cost(D, s)\} \quad \text{and} \quad cost_E(D) = \sum_{s \in S} cost(D, s)p(s). \quad (1)$$

Given a positive number j and two decision trees D_E and D_W for instance I , the procedure `CombineTrees`(D_E, D_W, j) (see Algorithm 1) constructs a new decision tree D^j for I whose worst testing cost is increased by at most j w.r.t the worst testing cost of D_W , i.e., $cost_W(D^j) \leq j + cost_W(D_W)$. Our algorithm uses the definition of a *j -replaceable node*, by which we mean a node v in D such that the total cost of the tests on the path from the root of D to v (including v) is larger than j and the cost of the path from the root of D to the parent of v is smaller than or equal to j . The procedure `Trade-Off` repeatedly uses `CombineTrees` to create several decision trees (one of these trees being D_W) with increasingly worst testing cost and chooses the one with the best expected testing cost. We will show that this way it can guarantee the best possible trade-off.

Algorithm 1 Computes trade-off tree between D_W and D_E

Procedure `CombineTrees`(D_E, D_W, j)

- 1: $D^j \leftarrow D_E$
- 2: Traverse D^j and construct $R = \{v \mid v \text{ is a } j\text{-replaceable node of } D^j\}$
- 3: **for** each $v \in R$ **do**
- 4: Replace in D^j the subtree rooted at v with D_W
- 5: **return** D^j

Procedure `Trade-Off`(D_E, D_W, C)

- 1: **for** $j = 0, \dots, C$ **do**
 - 2: $D^j \leftarrow \text{CombineTrees}(D_E, D_W, j)$
 - 3: $j^* \leftarrow \arg \min_{0 \leq j \leq C} cost_E(D^j)$
 - 4: **return** D^{j^*}
-

Proposition 1 *The decision tree D^j returned by `CombineTrees` has worst testing cost at most $j + cost_W(D_W)$.*

2 Correct Analysis of Algorithm 1

Now we analyze the decision tree $D = D^{j^*}$ output by `Trade-Off`(D_E, D_W, C), where C is an integer parameter. Notice that D is the decision tree with minimum expected testing cost among the decision trees $D^0, D^1, D^2, \dots, D^C$, where D^j is the decision tree returned by `CombineTrees`(D_E, D_W, j). It follows from the previous proposition that $cost_W(D) \leq C + cost_W(D_W)$.

The analysis of the expected testing cost of D is more involved. In order to simplify the notation we will let $W = cost_W(D_W)$. We also assume for simplicity in the

following that test costs are integers. Given a decision tree D' and an object/leaf $s \in S$ with $\text{cost}(D', s) = \kappa$ we will say that s has cost κ in D' .

Let p_i , with $i = 1, \dots, C$, be the sum of the probabilities of objects with cost i in D_E and p_{C+1} be the sum of the probabilities of the objects with cost larger than C in D_E . Clearly:

$$\text{cost}_E(D_E) \geq \sum_{i=1}^{C+1} p_i \cdot i.$$

Furthermore, for $j = 0, \dots, C$, we have that:

$$\text{cost}_E(D^j) \leq \sum_{i=1}^j p_i \cdot i + \left((j + W) \sum_{i=j+1}^{C+1} p_i \right)$$

because the objects whose cost in D_E is larger than j have cost at most $j + W$ in D^j . Moreover, for a probability distribution $\mathbf{q} = (q_1, \dots, q_{C+1})$, let

$$f(\mathbf{q}) = \min_{j=0, \dots, C} \left\{ \frac{\sum_{i=1}^j q_i \cdot i + (j + W) \sum_{i=j+1}^{C+1} q_i}{\sum_{i=1}^{C+1} q_i \cdot i} \right\},$$

and let $\mathbf{p} = (p_1, \dots, p_{C+1})$. Thus, we have

$$\frac{\text{cost}_E(D)}{\text{cost}_E(D_E)} = \min_{j=0, \dots, C} \left\{ \frac{\text{cost}_E(D^j)}{\text{cost}_E(D_E)} \right\} \leq f(\mathbf{p}) \leq \max_{\mathbf{q} \in \mathcal{P}} f(\mathbf{q}), \tag{2}$$

where $\mathcal{P} = \{(q_1, q_2, \dots, q_{C+1}) \mid \sum_{i=1}^{C+1} q_i = 1 \text{ and } q_1, q_2, \dots, q_{C+1} \geq 0\}$. The next lemma gives the exact value of $\max_{\mathbf{q} \in \mathcal{P}} f(\mathbf{q})$.

Lemma 1 Let $\mathbf{p}^* = (p_i^*, \dots, p_{C+1}^*)$, with

$$p_i^* = \begin{cases} \frac{(W-1)^{i-1}}{W^i}, & i = 1, \dots, C \\ \frac{(W-1)^C}{W^C}, & i = C + 1. \end{cases} \tag{3}$$

We have that $\mathbf{p}^* \in \mathcal{P}$ and

$$f(\mathbf{p}^*) = \frac{1}{\left(1 - \left(\frac{W-1}{W}\right)^{C+1}\right)} = \max_{\mathbf{q} \in \mathcal{P}} f(\mathbf{q}).$$

Proof The statements $\mathbf{p}^* \in \mathcal{P}$ and

$$f(\mathbf{p}^*) = \frac{1}{\left(1 - \left(\frac{W-1}{W}\right)^{C+1}\right)}$$

can be verified through simple calculations (see Appendix A).

To prove that $f(\mathbf{p}^*) = \max_{\mathbf{q} \in \mathcal{P}} f(\mathbf{q})$, we first show that

$$\max_{\mathbf{q} \in \mathcal{P}} f(\mathbf{q}) = \max_{\mathbf{q}'} \min_{j=0, \dots, C} \left\{ \sum_{i=1}^j q'_i \cdot i + (j + W) \sum_{i=j+1}^{C+1} q'_i \right\} \text{ s. t.} \tag{4}$$

$$\sum_{i=1}^{C+1} i \cdot q'_i = 1 \tag{5}$$

$$q'_i \geq 0, \quad i = 1, \dots, C + 1. \tag{6}$$

In fact, let $\mathbf{r} = \operatorname{argmax}_{\mathbf{q} \in \mathcal{P}} f(\mathbf{q})$ and let $K = \sum_{i=1}^{C+1} i \cdot r_i$. Then, \mathbf{r}/K is a solution for the problem defined by Eqs. (4)–(6) and its objective value is $f(\mathbf{r})$. Conversely, let \mathbf{r}' be the optimal solution of the problem defined by Eqs. (4)–(6) and let z' be the corresponding objective function value. Moreover, let

$$K' = \frac{1}{\sum_{i=1}^{C+1} r'_i}.$$

Then, $K'\mathbf{r}' \in \mathcal{P}$ and $f(K'\mathbf{r}') = z'$,

Therefore, we can analyze the optimum value of the optimization problem defined by Eqs. (4)–(6). This problem can be formulated as a linear program as follows:

$$\max z \quad \text{s. t.} \tag{7}$$

$$z - \sum_{i=1}^j i \cdot q_i - (j + W) \left(\sum_{i=j+1}^{C+1} q_i \right) \leq 0, \quad j = 0, \dots, C \tag{8}$$

$$\sum_{i=1}^{C+1} i \cdot q_i = 1 \tag{9}$$

$$q_i \geq 0, \quad i = 1, \dots, C + 1. \tag{10}$$

Thus, to show that

$$\max_{\mathbf{q} \in \mathcal{P}} f(\mathbf{q}) = \frac{1}{\left(1 - \left(\frac{W-1}{W}\right)^{C+1}\right)},$$

it suffices to exhibit a feasible solution for the dual of the above LP with objective value $\frac{1}{\left(1 - \left(\frac{W-1}{W}\right)^{C+1}\right)}$.

The dual of the LP defined by Eqs. (7)–(9) is given by

$$\min \lambda_E \quad \text{s. t.} \tag{11}$$

$$k \cdot \lambda_E - \sum_{j=0}^{k-1} (j + W)\lambda_j - \sum_{j=k}^C k\lambda_j \geq 0, \quad k = 1, \dots, C + 1 \tag{12}$$

$$\sum_{j=0}^C \lambda_j \geq 1 \tag{13}$$

$$\lambda_j \geq 0, \quad j = 0, \dots, C. \tag{14}$$

It is possible to show (see Appendix B for calculations) that $\lambda^* = (\lambda_E^*, \lambda_0^*, \dots, \lambda_C^*)$, where

$$\lambda_j^* = \frac{W^j(W - 1)^{C-j}}{W^{C+1} - (W - 1)^{C+1}},$$

for $j = 0, \dots, C$ and $\lambda_E^* = \frac{1}{\left(1 - \left(\frac{W-1}{W}\right)^{C+1}\right)}$ is a feasible solution for the dual problem,

which establishes the lemma. □

Thus, by setting $C = \lfloor \rho W \rfloor$ we get the following theorem.

Theorem 1 Fix an instance I of the decision tree optimization problem and let D_E be a decision tree such that $cost_E(D_E) = OPT_E(I)$. For every $\rho > 0$ there exists a decision tree D such that

$$cost_W(D) \leq (1 + \rho)OPT_W(I) \quad \text{and} \quad cost_E(D) \leq \left(\frac{1}{1 - e^{-\rho}}\right) OPT_E(I).$$

Proof Let $W = OPT_W(I)$ and $C = \lfloor \rho W \rfloor$. Let D_W be a decision tree such that $cost_W(D_W) = W$. It follows from the analysis above that the decision tree D output by Trade-Off(D_E, D_W, C) has worst testing cost at most $C + W < (1 + \rho)W$ and expected testing cost smaller than

$$\frac{1}{\left(1 - \left(\frac{W-1}{W}\right)^{C+1}\right)} OPT_E(I) \leq \left(\frac{1}{1 - \left(\frac{W-1}{W}\right)^{\rho W}}\right) OPT_E(I) \leq \left(\frac{1}{1 - e^{-\rho}}\right) OPT_E(I).$$

A Calculations for \mathbf{p}^* in Lemma 1

In order to verify that $\mathbf{p}^* \in \mathcal{P}$ it suffices to use the fact that p_0^*, \dots, p_C^* form a geometric progression with ratio $(W - 1)/W$.

To show that

$$f(\mathbf{p}^*) = \frac{1}{1 - \frac{(W-1)^{C+1}}{W^{C+1}}},$$

it is enough to use the following identities

$$\begin{aligned} \sum_{i=1}^{C+1} i \cdot p_i^* &= W - (C + W) \frac{(W - 1)^C}{W^C} + (C + 1) \frac{(W - 1)^C}{W^C} \\ \sum_{i=1}^j i \cdot p_i^* &= W - (j + W) \frac{(W - 1)^j}{W^j} \\ \sum_{i=j+1}^{C+1} p_i^* &= \frac{(W - 1)^j}{W^j}. \end{aligned}$$

B Calculations with the Dual Solution in Lemma 1

In this section we prove that $\lambda^* = (\lambda_E^*, \lambda_0^*, \lambda_1^*, \dots, \lambda_C^*)$ is a feasible solution for the dual problem given by Eqs. (11)–(14).

To verify that constraint (13) is satisfied we use the fact that $\lambda_0^*, \lambda_1^*, \dots, \lambda_C^*$ form a geometric sequence. In addition, to verify that constraint (12) holds we use the following identities:

$$W \sum_{j=0}^{k-1} \lambda_j^* = \frac{(W - 1)^{C+1-k} \cdot W^{k+1} - W(W - 1)^{C+1}}{W^{C+1} - (W - 1)^{C+1}} \tag{15}$$

$$k \sum_{j=k}^C \lambda_j^* = \frac{k \cdot W^{C+1} - kW^k(W - 1)^{C-k+1}}{W^{C+1} - (W - 1)^{C+1}}, \tag{16}$$

$$\sum_{j=0}^{k-1} j \lambda_j^* = \frac{kW^k \cdot (W - 1)^{C-k+1} - W^{k+1} \cdot (W - 1)^{C+1-k} + W(W - 1)^{C+1}}{W^{C+1} - (W - 1)^{C+1}} \tag{17}$$

and

$$k\lambda_E^* = \frac{kW^{C+1}}{W^{C+1} - (W - 1)^{C+1}}.$$

Reference

1. Saettler, A., Laber, E., Cicalese, F.: Trading of worst and expected cost in decision tree problems. *Algorithmica* **79**(3), 886–908 (2017)