Versatile Verification of Tree Ensembles – Supplement

Laurens Devos [1] **Wannes Meert** [1] **Jesse Davis** [1]

A. Proof of Theorem 2

We repeat the equations relevant to Theorem 2. The following is the single-instance constraint optimization problem:

$$\max_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{T}(\boldsymbol{x}) \quad \text{subject to} \quad \mathcal{C}(\boldsymbol{x}). \tag{3}$$

Next, the state expansion formula:

$$\begin{aligned} C([l^1_{i_1}, \ldots, l^m_{i_m}]) = \\ \{[l^1_{i_1}, \ldots, l^m_{i_m}, l^{m+1}] \mid l^{m+1} \in L^{m+1}, \\ \mathrm{box}(l^1_{i_1}, \ldots, l^m_{i_m}, l^{m+1}) \neq \emptyset\}. \end{aligned} \tag{4}$$

Finally, the definition of the scoring function $f = g + h$:

$$g(s) = \sum_{m'=1}^{m} \nu^{m'}_{i_{m'}}, \tag{5}$$

$$h(s) = \sum_{m'=m+1}^{M} h_{m'}(s), \tag{6}$$

$$\begin{aligned} h_{m'}(s) = \max\{ \; \nu^{m'}_j \mid l^{m'} \in L^{m'}, \\ \mathrm{box}(l^1_{i_1}, \ldots, l^m_{i_m}, l^{m'}) \neq \emptyset \;\}. \end{aligned} \tag{7}$$

Theorem 2. *The heuristic search in S is guaranteed to find the max-clique in $\boldsymbol{G}$ which corresponds to the optimal output configuration of the optimization problem in Equation 3.*

Proof. We show that the heuristic is consistent and thus admissible, which ensures that the search finds the optimal solution in an optimally efficient way (Dechter & Pearl, 1985).

For each state $s = [l^1_{i_1}, \ldots, l^m_{i_m}]$ and any extension $t = [l^1_{i_1}, \ldots, l^m_{i_m}, l^{m+1}_j]$, h is consistent when $h(s) \geq \nu^{m+1}_j + h(t)$. Unlike $h(t)$, $h(s)$ contains a term $h_{m+1}(s)$, which is an overestimation of ν^{m+1}_j, that is, $h_{m+1}(s) \geq \nu^{m+1}_j$. Additionally, each $h_{m'}(s) \geq h_{m'}(t), m+1 < m' \leq M$ because $\mathrm{box}(s) \supseteq \mathrm{box}(t)$, and consequently, each $\nu^{m'}_j$ considered in the max of $h_{m'}(t)$ is also considered in the max of $h_{m'}(s)$ (Equation 7). □

B. Proof of Consistent Heuristic Equation 9

We prove that $h_2(s_2) - h_1(s_1)$ is a consistent heuristic for the optimization problem in Equation 2 of the main paper (admissibility follows from consistency). First, we show that h_1 is a consistent heuristic:

Theorem B.1. *The heuristic obtained by replacing* max *by* min *in Equation 7 in the main paper, i.e.,*

$$\begin{aligned} h_1(s_1) = \sum_{m'=m+1}^{M} \min\{ \; \nu^{m'} \mid l^{m'} \in L^{m'}, \\ \mathrm{box}(l^1_{i_1}, \ldots, l^m_{i_m}, l^{m'}) \neq \emptyset \;\}, \end{aligned}$$

is a consistent heuristic for the minimization problem $\min_{\boldsymbol{x}_1} \boldsymbol{T}_1(\boldsymbol{x}_1)$.

Proof. For minimization problems, consistent means that $h_1(s_1) \leq \nu + h_1(t_1)$ for two consecutive states s_1 and t_1, and the leaf value ν added to $g_1(t_1) = g_1(s_1) + \nu$. This proof is analogous to the previous proof of Theorem 2. □

Theorem B.2. *The heuristic $h(s_1, s_2) = h_2(s_2) - h_1(s_1)$ is a consistent heuristic for the optimization problem in Equation 2 of the main paper, i.e.,*

$$\max_{\boldsymbol{x}_1,\boldsymbol{x}_2\in\mathcal{X}} \boldsymbol{T}_2(\boldsymbol{x}_2) - \boldsymbol{T}_1(\boldsymbol{x}_1) \quad \textit{subject to} \quad \mathcal{C}(\boldsymbol{x}_1, \boldsymbol{x}_2).$$

Proof. We show that $h(s_1, s_2) \geq -\nu_1 + h(t_1, s_2)$ and $h(s_1, s_2) \geq \nu_2 + h(s_1, t_2)$, for successive state pairs (s_1, t_1) and (s_2, t_2) and leaf values ν_1 and ν_2 added when transitioning from s_1 to t_1 and s_2 to t_2 respectively.

We use the fact that h_1 is a consistent heuristic with respect to the minimization of $\boldsymbol{T}_1(\boldsymbol{x}_1)$ ($h_1(s_1) \leq \nu_1 + h_1(t_1)$) and h_2 is a consistent heuristic with respect to the maximization of $\boldsymbol{T}_2(\boldsymbol{x}_2)$ ($h_2(s_2) \geq \nu_2 + h_2(t_2)$).

For the first case:

$$\begin{aligned} h(s_1, s_2) &= h_2(s_2) - h_1(s_1) \\ &\geq h_2(s_2) - (\nu_1 + h_1(t_1)) \\ &= -\nu_1 + h(t_1, s_2). \end{aligned}$$

For the second case:

$$\begin{aligned} h(s_1, s_2) &= h_2(s_2) - h_1(s_1) \\ &\geq (\nu_2 + h_2(t_2)) - h_1(s_1) \\ &= \nu_2 + h(s_1, t_2). \end{aligned}$$

[1] Department of Computer Science, KU Leuven, Leuven, Belgium. Correspondence to: Laurens Devos <laurens.devos@kuleuven.be>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

This concludes the proof. □

C. Details of Experiments

Table 1 gives an overview of the properties and parameters used for each dataset. Each model was trained on a training set consisting of 90% of all data. The remaining 10% is used as the test set. The learning rate is always determined using hyper-parameter optimization given an ensemble size and a tree depth. We train models with learning rates $0.25, 0.5, 0.75, 1.0$, pick the best value η, and then again train models with learning rates $\eta - 0.17, \eta - 0.083, \eta + 0.083, \eta + 0.17$. The best η value is determined based on the performance on the test data.

We used XGBoost version 1.2.1 (Chen & Guestrin, 2016).

C.1. Robustness Details

To find a lower bound of the l_∞ distance to the closest adversarial example, we do a binary search. We begin with a start value $\delta = \delta_{\text{start}}$ (see Table 1), and maintain lower and upper limits $\underline{\delta}$ and $\bar{\delta}$, initially 0 and δ_{start}. We proceed as follows. For a binary classification problem and for an example $\boldsymbol{x}$ with a negative predicted label $\boldsymbol{T}(\boldsymbol{x}) < 0$, solve the following optimization problem:

$$\max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \boldsymbol{T}(\hat{\boldsymbol{x}}) \;\text{ subject to }\; ||\boldsymbol{x} - \hat{\boldsymbol{x}}||_\infty < \delta.$$

Both VERITAS and MERGE compute an upper bound $\bar{b}$ on this maximum value. If $\bar{b}$ is less than zero, then we are certain that no $\hat{\boldsymbol{x}}$ can exist with a flipped label within a distance δ of $\boldsymbol{x}$. So we increase our estimate $\delta = \underline{\delta} + (\bar{\delta} - \underline{\delta})/2$, with $\underline{\delta}$ set to the last δ. If $\bar{b}$ is greater than zero, then a $\hat{\boldsymbol{x}}$ may exist with a positive label, so we decrease our estimate $\delta = \underline{\delta} - (\bar{\delta} - \underline{\delta})/2$, with $\bar{\delta}$ set to the last δ. For an instance with a positive label, the step are analogous but we minimize instead of maximize.

For a multi-classification problem like MNIST and Fashion-MNIST, we use the two-instance setting. For an example $\boldsymbol{x}$ with label l, do a binary search for all other labels $l' \neq l$, and in each step of the binary search, optimize the following:

$$\max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \boldsymbol{T}_{l'}(\hat{\boldsymbol{x}}) - \boldsymbol{T}_{l}(\hat{\boldsymbol{x}}) \;\text{ subject to }\; ||\boldsymbol{x} - \hat{\boldsymbol{x}}||_\infty < \delta,$$

i.e., optimize the difference between the weight of the classifier for label l and the weight of the classifier for l'. If this difference is positive, then a $\hat{\boldsymbol{x}}$ may exist for which the classifier is more confident in its prediction for label l' than for label l. The δ values are updated as in the binary classification case.

A (suboptimal) full solution – which VERITAS can generate when using the ARA* heuristic – can be used to update $\bar{\delta}$. The distance between the suboptimal solution and $\boldsymbol{x}$ is indeed an upper limit of the optimal δ, as the optimal distance is either smaller or the same. Using these suboptimal solutions in the search improves the convergence rate.

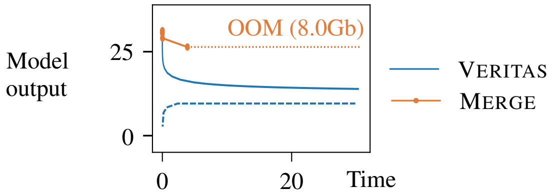

Figure 1. A typical example of how VERITAS's upper and lower bounds progress over time in comparison to MERGE's upper bound. VERITAS finds tight upper and lower bounds. Merge runs out of memory (OOM) before the timeout of 30 seconds.

For a single invocation of the optimization problem, an upper bound $\bar{b}$ is computed. Figure 1 shows a single illustrative example of how VERITAS's and MERGE's bound develop as time proceeds. Two examples of the updates by the binary search to the δ values are plotted in Figure 2.

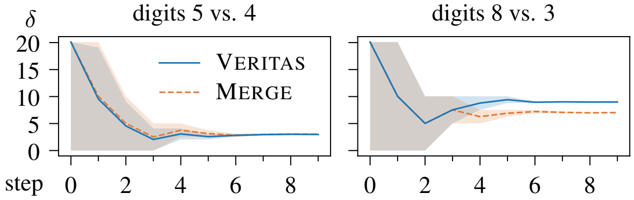

Figure 2. Illustrations of two binary search executions on two MNIST digits, a 5 and an 8. VERITAS and MERGE compute a lower bound on the distance δ to the closest adversarial example that is classified as a 4 (left) and a 3 (right). The background colors indicate the remaining binary search interval on δ. A higher $\underline{\delta}$ value is better.

References

Chen, T. and Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. ACM, 2016.

Dechter, R. and Pearl, J. Generalized best-first search strategies and the optimality of a*. *J. ACM*, 32(3):505–536, 1985.

Table 1. Overview of the properties and parameters used for each dataset. The columns are the number of rows n, the number of attributes k, the number of trees M, the maximum depth d, the learning rate η, the start δ of the robustness binary search, and the MERGE parameters T and L, and the accuracy on the test set.

Dataset	n	k	M	d	η	δ_{start}	T	L	Acc.
covtype	581 012	54	80	8	1.0	0.2	2	2	94.9%
f-mnist	70 000	784	200	8	0.33	20	2	1	91.1%
higgs	11 000 000	28	300	8	0.33	0.05	4	1	76.0%
ijcnn1	49 990	22	60	8	0.25	0.1	2	2	98.9%
mnist	70 000	784	200	8	0.33	40	2	2	98.3%
webspam	350 000	254	100	8	0.33	0.05	2	1	99.3%
mnist2v6	13 866	784	1000	4	0.25	40	4	1	99.6%