Robustness Verification of Tree-based Models
Hongge Chen* (MIT), Huan Zhang* (UCLA), Si Si (Google), Yang Li (Google), Duane Boning (MIT), and Cho-Jui Hsieh (UCLA) (*Equal Contribution)

Source code (XGBoost compatible!): https://github.com/chenhongge/treeVerification

Introduction

Robustness Verification problem:

\[ f^* = \min f(x + \delta) \]
\[ \|\delta\|_\infty \leq \epsilon \]

We compute a lower bound of \( f^* \) and improve it iteratively.

- We verify the robustness for tree based models (include GBDT, random forest, etc)
- Cast as a max-clique enumeration problem on a multi-partite graph with bounded boxicity.
- up to 100X faster than exact verification, small gap to \( f^* \)

Verifying your XGBoost model today!
https://github.com/chenhongge/treeVerification

Single Tree Verification

Insight: decision tree nodes partition the feature space using boxes, whose boundaries can be tracked. The partitions can be generated in linear time.

Exact verification of a single tree is easy!

But how to verify a tree ensemble?

- Naive: consider the worst case of each tree, and add worst case together (loose bound, but very fast)
- Ours: consider multiple trees together using graph theory (much tighter)

Tree Ensemble Verification

If \( f(x) > 0 \), we are safe as long as the largest sum of leaf values in all reachable regions is > 0.

Theorem: Each K-clique of the graph represents a reachable output for \( x \) after perturbation.

Efficient Multi-level Verification

Finding K-cliques on all K trees can be expensive.
We can group K trees to M groups and find (K/M)-cliques inside each group, and use naive bounds between groups up to 100X faster than exact verification, small gap to \( f^* \)

Experiments

The average \( \epsilon \) distance of Cheng's attack [1] and our verification method. The distance is normalized with distance by MILP [2]. Numbers close to 1 indicate a tighter bound.

The average running time of Cheng's attack and our verification method. The running time is normalized with MILP's running time.

References:

Unlike the minimax based adversarial training on deep training, [3] uses a similar maximin robust optimization formulation but can be verified. Compared to DNNs, tree based models are more verifiable (by MILP based exact verification [2] and this work) as tight and fast verification methods are available.