Learning And Mining from DatA
www.lamda.nju.edu.cn

Fast Provably Robust Decision Trees and Boosting

Jun-Qi Guo, Ming-Zhuo Teng, Wei Gao and Zhi-Hua Zhou

Nanjing University

LAMDA
Learning And Mining from DatA
www.lamda.nju.edu.cn

About this work

Much attention has been paid to Robust decision trees and ensembles

Previous methods: high comp. or no guarantee of provable robustness

In this work, we propose:

✓ **Fast Provably Robust Decision Tree (FPRDT)**

- A tradeoff between local and global optimization
- the smallest computational complexity

✓ **Provably Robust AdaBoost (PRAdaBoost)**

- the smallest computational complexity

Methods	Comp. complexity	Prov. robustness
Our FPRDT	$O(n \log n)$	✓
Our PRAdaBoost	$O(n \log n)$	✓
ROCT (Vos & Verwer, 2021b)	$O(\exp(n))$	✓
TREANT (Calzavara et al., 2020)	$O(n^2)$	✓
PRB (Andriushchenko & Hein, 2019)	$O(n^2)$	✓
GROOT (Vos & Verwer, 2021a)	$O(n \log n)$	✗
RIGDT-heuristic (Chen et al., 2019a)	$O(n \log n)$	✗
RGBDT (Chen et al., 2019a)	$O(n \log n)$	✗
AdvBoost (Kantchelian et al., 2016)	$O(n \log n)$	✗

Outline

- **Introduction**
- Our work
  - Robust decision trees
  - Robust AdaBoost
- Experiments
- Conclusion

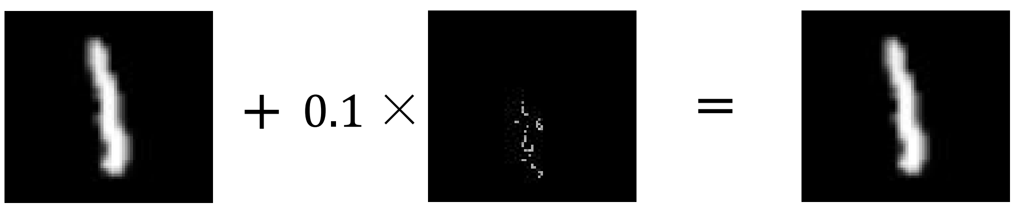
Decision tree and ensembles

Learning And Mining from DatA
www.lamda.nju.edu.cn

Decision trees and ensembles are important learning algorithms

- Decision trees [Quinlan et al., 1986; Safavian et al., 1991; Maimon et al., 2014]
- AdaBoost [Freund et al., 1996; Ratsch et al., 2001; Bartlett et al., 2006]
- Random forests [Breiman, 2001; Biau, 2012; Athey et al., 2019]
- ……

Those models are **vulnerable** to adversarial perturbations:



Prediction: “1”
Loss: 0

Adversarial
perturbation

Prediction: “7”
Loss: 1

Previous robust decision trees and ensembles

- Global robust decision tree [Vos et al., 2021b]
- Boosting method [Andriushchenko et al., 2019; Chen et al., 2019]

high comp. complexity

- Local robust decision tree [Chen et al., 2019; Vos et al., 2021a]
- Random forests based method [Vos et al., 2021a]

no guarantee of provable robust.

- ……

Can we learn robust decision trees and Boosting with

- low computational complexity
- guarantees of provable robustness

Outline

- Introduction
- **Our work**
  - **Robust decision trees**
  - Robust AdaBoost
- Experiments
- Conclusion

Adversarial robust learning

Adversarial robust learning aims to optimize

$$\min_{h \in \mathcal{H}} \sum_{i=1}^{n} \max_{||\boldsymbol{z}_i - \boldsymbol{x}_i||_\infty < \epsilon} l(h(\boldsymbol{z}_i), y_i)$$

- training data $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$
- learning model $h \in \mathcal{H}$
- loss function $l(\cdot\,,\cdot) \to R$

Our work: decision tree model h

the 0/1 loss $l(h(\boldsymbol{z}_i), y_i) = \mathbb{I}[h(\boldsymbol{z}_i) \neq y_i]$

Objective loss for robust decision tree

Learning And Mining from DatA
www.lamda.nju.edu.cn

- A decision tree has m leaf nodes
- Outputs $v_j = h(\boldsymbol{z})$ for instance $\boldsymbol{z}$, which is belong to the j-th leaf

Our objective loss for decision tree is given by

$$\sum_{i=1}^{n} \max_{||\boldsymbol{z}_i - \boldsymbol{x}_i||_\infty < \epsilon} l(h(\boldsymbol{z}_i), y_i) = \sum_{i=1}^{n} \underbrace{\max_{j \in [m]}}_{\text{Traversing all nodes}} \left\{ \mathbb{I}\left[v_j \neq y_i\right] \cdot \underbrace{\mathrm{Fall}(i, j)}_{\text{Is the } i\text{-th instance belong to the } j\text{-th leaf}} \right\}$$

Main idea: potential predictions of $\boldsymbol{x}_i$ can be solved by traversing all nodes for decision tree

Decomposition of objective loss at a split

- Split the p-th leaf to leaves l and r
- Let v_l and v_r denote the output for two leaves, respectively

We can decompose the objective loss at the split on the p-th leaf as

$$\max_{j \in \{l,r\} \cup [m] \setminus p} \{\mathbb{I}[v_j \neq y_i] \mathrm{Fall}(i,j)\}$$

$$= \max\{\underbrace{\mathbb{I}[v_l \neq y_i]\mathrm{Fall}(i,l)}_{\substack{\text{loss for left child}\\ \text{(easy)}}}, \underbrace{\mathbb{I}[v_r \neq y_i]\mathrm{Fall}(i,r)}_{\substack{\text{loss for right child}\\ \text{(easy)}}}, \underbrace{\max_{j \in [m] \setminus \{p\}} \{\mathbb{I}[v_j \neq y_i]\mathrm{Fall}(i,j)\}}_{\substack{\text{maximal loss for other leaves}\\ \text{(hard)}}}\}$$

For simplicity, denote by $K_{ip} = \max_{j \in [m] \setminus \{p\}} \{\mathbb{I}[v_j \neq y_i] \mathrm{Fall}(i,j)\}$

Calculate the maximal loss K_{ip}

- Split the p-th leaf to leaves l and r
- K_{ip}: the maximal loss for other leaves
- E_i: the sum of loss for all leaves

We determine K_{ip} by the E_i minus the loss for p-th leaf:

$$\underbrace{K_{ip}}_{\text{maximal loss for other leaves}} = 1 \quad \leftrightarrow \quad \underbrace{E_i}_{\text{sum of loss for all leaves}} - \underbrace{\mathbb{I}[v_p \neq y_i]\mathrm{Fall}(i,p)}_{\text{loss for } p\text{-th leaf}} \geq 1$$

E_i can be updated efficiently after each split.

Solution

Split the p-th leaf to leaves l and r

Let v_l and v_r denote the output for two leaves, respectively

Let t and η denote the split feature and threshold, respectively

The solution steps:

- Traverse all feature t
- Traverse threshold η from sorted potential splits
- Fixed t' and η', by traverse $v_l, v_r \in \{0,1\}$, we solve

$$(v_l', v_r') = \operatorname*{argmin}_{v_l, v_r \in \{0,1\}} L(t', \eta', v_l, v_r)$$

- Update the optimal solution

Fast Provably Robust Decision Trees

Algorithm 2 FPRDT-recursive

Input: Dataset S_n, split leaf p, perturbation size ϵ, previous optimal loss $\hat{\mathcal{L}}^{\text{pre}}$
Output: None
Initialize $\hat{\mathcal{L}}^* \leftarrow +\infty$
Compute loss K_{ip} ($i \in [n]$) according to Eqn. (3) — calculate the maximum loss among other leaves
for $t' = 1$ **to** d **do**
  Compute $W_{t'}$ according to Eqn. (4) — traverse potential split features and sorted thresholds
  for η' **in** sorted($W_{t'}$) **do**
    Compute $\mathbb{I}[\mathcal{E}_{il} \neq \emptyset]$ and $\mathbb{I}[\mathcal{E}_{ir} \neq \emptyset]$ by Eqns. (5)-(6) — compute the corresponding optimal loss
    Solve v_l' and v_r' from Eqn. (7) with optimal loss $\hat{\mathcal{L}}'$
    if $\hat{\mathcal{L}}' < \hat{\mathcal{L}}^*$ **then** — update the minimum loss and the optimal solution
      $t^* = t', \eta^* = \eta', v_l^* = v_l', v_r^* = v_r', \hat{\mathcal{L}}^* = \hat{\mathcal{L}}'$
    end if
  end for
end for
if $\hat{\mathcal{L}}^* < \hat{\mathcal{L}}^{\text{pre}}$ **then**
  Split leaf p via $\{t^*, \eta^*, v_l^*, v_r^*\}$, and obtain children l, r
  Update E_i ($i \in [n]$) according to Eqn. (8)
  FPRDT-recursive(S_n, l, ϵ, $\hat{\mathcal{L}}^*$)
  FPRDT-recursive(S_n, r, ϵ, $\hat{\mathcal{L}}^*$)
end if

Computation complexity $\mathcal{O}(n\log n)$

- sort the potential thresholds $\mathcal{O}(n\log n)$
- solve minimum $\mathcal{O}(n)$

Outline

- Introduction
- **Our Work**
  - Robust Decision Trees
  - **Robust AdaBoost**
- Experiments
- Conclusion

Upper bound on robust AdaBoost

Learning And Mining from DatA
www.lamda.nju.edu.cn

AdaBoost essentially optimizes the exponential loss

For robust AdaBoost, it is an NP-hard problem to optimize the adversarial exponential loss [Kantchelian et al., 2013]

We consider the upper bound on adversarial exponential loss as

$$\max_{||z_i - x_i||_\infty < \epsilon} \exp\left(\sum_{j=1}^{m} - y_j \alpha_j h_j(z_i)\right) \leq \prod_{j=1}^{m} \max_{||z_i - x_i||_\infty < \epsilon} \{\exp(- y_i \alpha_j h_j(z_i))\}$$

Main idea: relax the robust problem to several robust sub-problems for each base learner

Main idea of our fast provably robust AdaBoost

- We use our fast provably robust decision trees (FPRDT) as base learner by minimizing the weighted adversarial 0/1 loss
- Robust AdaBoost updates the instance weights by

$$w_{t+1,i} = w_{t,i} \max_{||z_i - x_i||_\infty < \epsilon} \{\exp(-y_i \alpha_t h_t(z_i))\}$$

Complexity $\mathcal{O}(n\log n)$: update the weights $\mathcal{O}(n)$
train the base learner $\mathcal{O}(n\log n)$

Convergence analysis for PRAdaBoost

Theorem Let $H(x)$ be the final classifier of PRAdaBoost with error ϵ_t for each iteration $t \in [T]$. We have

$$\frac{1}{n}\sum_{i=1}^{n} \min_{||z_i - x_i||_\infty < \varepsilon} \mathbb{I}[H(z_i) \neq y_i] \leq \exp\left(-2\sum_{t=1}^{T}(0.5-\epsilon_t)^2\right)$$

As AdaBoost, empirical adversarial 0/1 error of PRAdaBoost has the exponential decrease with the iteration number *T*

Outline

- Introduction
- Our Work
  - Robust Decision Trees
  - Robust AdaBoost
- **Experiments**
- Conclusion

Benchmark datasets

Dataset (Perb.)	# Inst.	# Feat.	Dataset (Perb.)	# Inst.	# Feat.
ionos (.2)	351	34	cifar10:0v5 (.1)	12,000	3,072
breast (.3)	683	9	cifar10:0v6 (.1)	12,000	3,072
diabet (.05)	768	8	cifar10:4v8 (.1)	12,000	3,072
bank (.1)	1,372	4	mnist2v6 (.4)	13,866	784
Japan3v4 (.1)	3,087	14	mnist3v8 (.4)	13,966	784
har1v2 (.1)	3,266	561	F-mnist2v5 (.2)	14,000	784
spam (.05)	4,601	57	F-mnist3v4 (.2)	14,000	784
GesDvP (.01)	4,838	32	F-mnist7v9 (.2)	14,000	784
wine (.05)	6,497	11	mnist1v7 (.4)	15,170	784

- Number of instances: 351 ~ 15170
- Number of features: 4 ~ 3072
- Perturbation rate follows previous work

Compared methods

LAMDA
Learning And Mining from DatA
www.lamda.nju.edu.cn

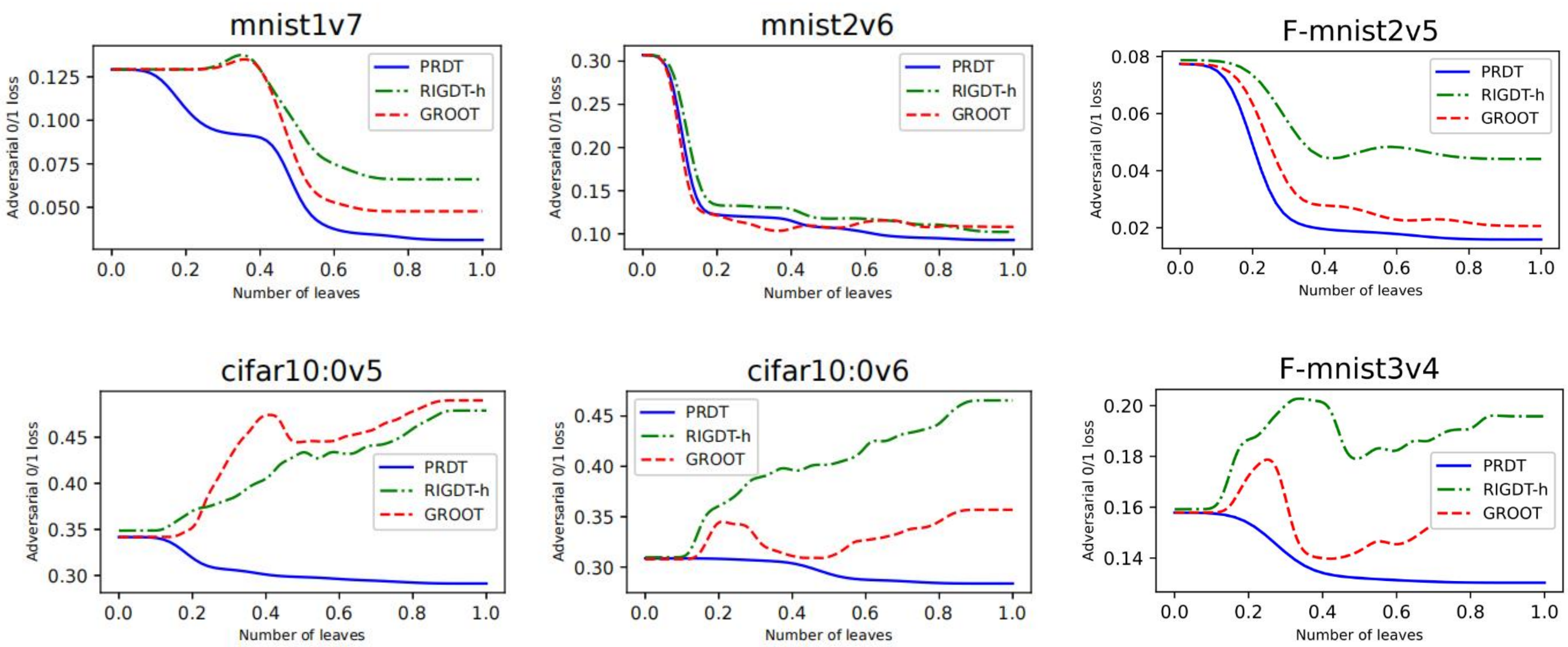
Robust decision trees

- **RIGBT-h**: Robust tree via adversarial information gain [Chen et al., 2019]
- **TREANT**: Robust tree with additional constraints [Calzavara et al., 2020]
- **GROOT**: Robust tree via adversarial Gini impurity [Vos et al., 2021a]
- **ROCT**: Robust tree with global optimization [Vos et al., 2021b]
- **PRB tree**: Robust tree based on adv. exp-loss [Andriushchenko et al., 2021]

Robust tree ensembles:

- **RGBDT**: Robust boosting via approximating adversarial loss [Chen et al., 2019]
- **RIGDT forest**: Random forest with base learner RIGBT-h [Vos et al., 2021b]
- **GROOT forest**: Random forest with base learner GROOT [Vos et al., 2021b]
- **PRBoosting**: Robust GBDT by upper bound of exp-loss [Andriushchenko et al., 2021]

Comparisons on training adversarial error



Our approach: **continuously decreases** the training adversarial errors

Unprovable methods: **may increase** the training adversarial errors

Accuracy comparisons for our FPRDT

Learning And Mining from DatA
www.lamda.nju.edu.cn

Dataset	FPRDT	Decision tree	RIGDT-h	GROOT	TREANT	ROCT	PRB tree
ionos	.7954 ± .0302	.3100 ± .0549•	.7015 ± .0874•	.7829 ± .0325•	.7232 ± .0434•	.7897 ± .0330•	.7601 ± .0346•
breast	.8765 ± .0290	.2501 ± .0457•	.8381 ± .0317•	.8744 ± .0273	.8334 ± .0318•	.8735 ± .0256	.8706 ± .0258•
diabet	.6674 ± .0258	.6333 ± .0346•	.5695 ± .0640•	.6481 ± .0352•	.6695 ± .0228	.6552 ± .0244•	.6633 ± .0333
bank	.6577 ± .0311	.6333 ± .0346•	.4685 ± .0713•	.5410 ± .0440•	.6092 ± .0259•	.6539 ± .0167	.6299 ± .0328•
Japan3v4	.6673 ± .0129	.5751 ± .0377•	.5638 ± .0337•	.5829 ± .0468•	N/A	.6671 ± .0170	.5954 ± .0112•
har1v2	.8044 ± .0249	.2316 ± .0434•	.7074 ± .0257•	.8058 ± .0198	N/A	.7786 ± .0165•	.7383 ± .0148•
spam	.7404 ± .0124	.0006 ± .0012•	.4669 ± .1019•	.7231 ± .0188•	N/A	.4813 ± .0469•	.6968 ± .0110•
GesDvP	.7301 ± .0174	.4783 ± .1390•	.5483 ± .1035•	.7164 ± .0180•	N/A	.7071 ± .0126•	.7014 ± .0190•
wine	.6364 ± .0062	.3515 ± .1430•	.4032 ± .0375•	.6373 ± .0073	.6388 ± .0079∘	.6117 ± .0343•	.6299 ± .0080•
cifar10:0v5	.6878 ± .0177	.2960 ± .0598•	.3469 ± .0457•	.4847 ± .0608•	N/A	.6639 ± .0113•	.6501 ± .0106•
cifar10:0v6	.6883 ± .0102	.5878 ± .0568•	.4771 ± .0168•	.5555 ± .0495•	N/A	.6684 ± .0077•	.6833 ± .0051•
cifar10:4v8	.6613 ± .0117	.2561 ± .0784•	.4882 ± .0468•	.4727 ± .0195•	N/A	.6319 ± .0114•	.6698 ± .0093∘
mnist2v6	.8954 ± .0025	.0178 ± .0320•	.8850 ± .0079•	.8725 ± .0110•	N/A	.7842 ± .0222•	.8385 ± .0673•
mnist3v8	.8527 ± .0058	.0028 ± .0071•	.8102 ± .0102•	.7575 ± .0185•	N/A	.7565 ± .0280•	.8030 ± .0235•
F-mnist2v5	.9780 ± .0027	.3214 ± .2143•	.9446 ± .0080•	.9714 ± .0054•	N/A	.9394 ± .0128•	.9761 ± .0025•
F-mnist3v4	.8652 ± .0056	.0166 ± .0521•	.7928 ± .0129•	.8193 ± .0104•	N/A	.8271 ± .0168•	.8407 ± .0052•
F-mnist7v9	.8760 ± .0058	.2930 ± .1554•	.8100 ± .0108•	.8291 ± .0115•	N/A	.8376 ± .0188•	.8399 ± .0106•
mnist1v7	.9633 ± .0034	.0036 ± .0081•	.9325 ± .0076•	.9457 ± .0052•	N/A	.8937 ± .0095•	.9196 ± .0264•
Average	.7802 ± .1090	.2922 ± .2172	.6530 ± .1877	.7233 ± .1494	—	.7342 ± .1134	.7503 ± .1075
Win/Tie/Loss		**18/0/0**	**18/0/0**	**15/3/0**	**16/1/1**	**15/3/0**	**16/1/1**

Our FPRDT: significantly better than other decision trees

- unprovable methods make local optimizations
- provable methods do not obtain good result due to high complexity

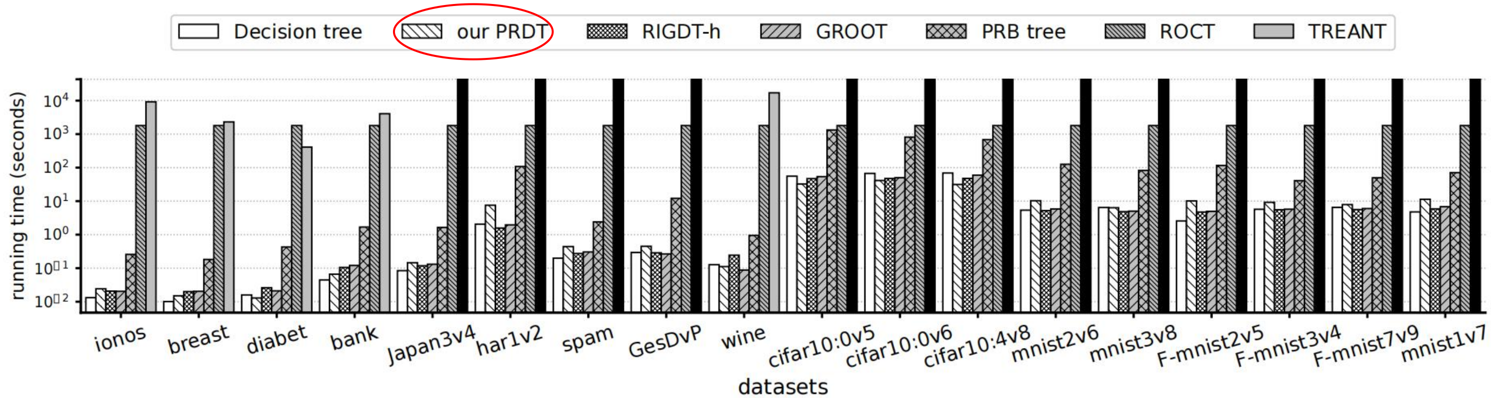
Accuracy comparisons for our PRAdaBoost

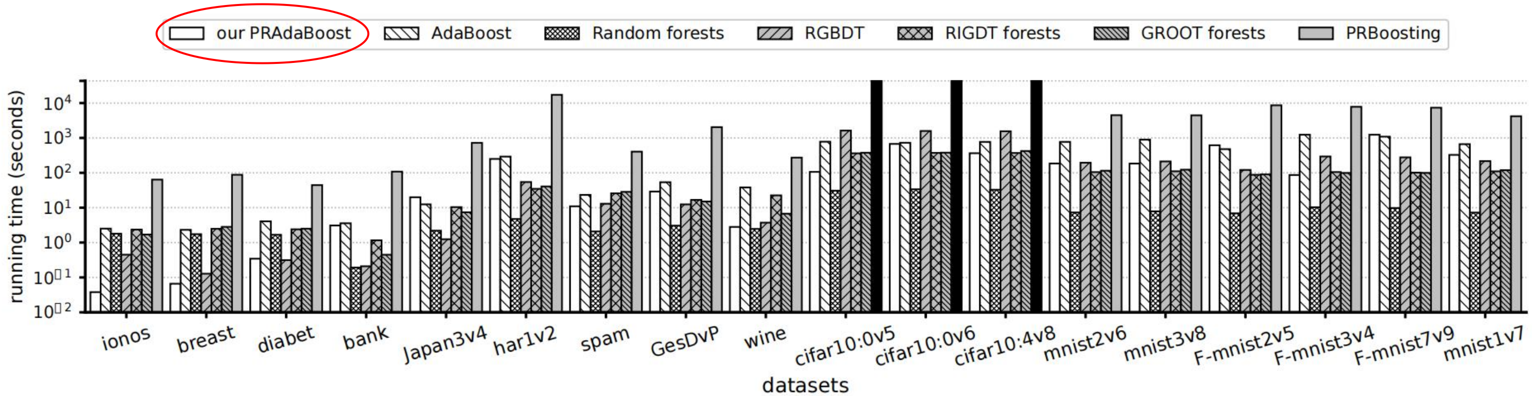
Dataset	PRAdaBoost	AdaBoost	Random forests	RGBDT	RIGDT forests	GROOT forests	PRBoosting
ionos	$.7960 \pm .0329$	$.0321 \pm .0137\bullet$	$.1122 \pm .0353\bullet$	$.5276 \pm .0472\bullet$	$.6565 \pm .0414\bullet$	$.7869 \pm .0346$	$.7755 \pm .0370$
breast	$.8793 \pm .0263$	$.0732 \pm .0156\bullet$	$.2167 \pm .0193\bullet$	$.7034 \pm .0648\bullet$	$.8437 \pm .0280\bullet$	$.8838 \pm .0251$	$.8694 \pm .0282\bullet$
diabet	$.6635 \pm .0281$	$.1352 \pm .0156\bullet$	$.4523 \pm .0411\bullet$	$.4560 \pm .0371\bullet$	$.5999 \pm .0371\bullet$	$.6578 \pm .0210\bullet$	$.6276 \pm .0325\bullet$
bank	$.6680 \pm .0361$	$.4019 \pm .0619\bullet$	$.5087 \pm .0292\bullet$	$.4427 \pm .0333\bullet$	$.5087 \pm .0292\bullet$	$.6407 \pm .0308\bullet$	$.6195 \pm .0403\bullet$
Japan3v4	$.6816 \pm .0165$	$.4913 \pm .0231\bullet$	$.5187 \pm .0184\bullet$	$.5885 \pm .0099\bullet$	$.6039 \pm .0171\bullet$	$.6580 \pm .0160\bullet$	$.6874 \pm .0167$
har1v2	$.8601 \pm .0162$	$.0092 \pm .0065\bullet$	$.8326 \pm .0137\bullet$	$.6417 \pm .0165\bullet$	$.4998 \pm .0212\bullet$	$.7917 \pm .0169\bullet$	$.8653 \pm .0130$
spam	$.7540 \pm .0129$	$.0000 \pm .0000\bullet$	$.0000 \pm .0000\bullet$	$.4888 \pm .0349\bullet$	$.6212 \pm .0202\bullet$	$.7495 \pm .0130\bullet$	$.7312 \pm .0101\bullet$
GestDvP	$.7315 \pm .0165$	$.1031 \pm .0079\bullet$	$.1887 \pm .0086\bullet$	$.2420 \pm .0090\bullet$	$.6459 \pm .0118\bullet$	$.7314 \pm .0127$	$.7318 \pm .0179$
wine	$.6397 \pm .0069$	$.0002 \pm .0004\bullet$	$.0910 \pm .0112\bullet$	$.1068 \pm .0118\bullet$	$.4161 \pm .0129\bullet$	$.6329 \pm .0008\bullet$	$.6356 \pm .0066$
cifar10:0v5	$.6906 \pm .0159$	$.0083 \pm .0035\bullet$	$.3015 \pm .0058\bullet$	$.3137 \pm .0095\bullet$	$.4413 \pm .0094\bullet$	$.5262 \pm .0123\bullet$	N/A
cifar10:0v6	$.6958 \pm .0092$	$.0556 \pm .0041\bullet$	$.3678 \pm .0103\bullet$	$.3446 \pm .0091\bullet$	$.5199 \pm .0082\bullet$	$.5604 \pm .0098\bullet$	N/A
cifar10:4v8	$.6710 \pm .0096$	$.0019 \pm .0014\bullet$	$.2956 \pm .0074\bullet$	$.2707 \pm .0103\bullet$	$.4614 \pm .0074\bullet$	$.4983 \pm .0068\bullet$	N/A
mnist2v6	$.9437 \pm .0046$	$.0000 \pm .0000\bullet$	$.0000 \pm .0000\bullet$	$.8442 \pm .0266\bullet$	$.8999 \pm .0062\bullet$	$.9249 \pm .0045\bullet$	$.9365 \pm .0044\bullet$
mnist3v8	$.8829 \pm .0106$	$.0000 \pm .0000\bullet$	$.0000 \pm .0000\bullet$	$.7155 \pm .0147\bullet$	$.7756 \pm .0081\bullet$	$.8228 \pm .0057\bullet$	$.8680 \pm .0082\bullet$
F-mnist2v5	$.9823 \pm .0028$	$.3852 \pm .0552\bullet$	$.4561 \pm .0041\bullet$	$.9645 \pm .0050\bullet$	$.9654 \pm .0040\bullet$	$.9791 \pm .0029\bullet$	$.9843 \pm .0018\circ$
F-mnist3v4	$.8674 \pm .0055$	$.0000 \pm .0000\bullet$	$.0441 \pm .0311\bullet$	$.7172 \pm .0131\bullet$	$.8063 \pm .0064\bullet$	$.8392 \pm .0063\bullet$	$.8640 \pm .0054$
F-mnist7v9	$.8691 \pm .0066$	$.0000 \pm .0000\bullet$	$.1357 \pm .0358\bullet$	$.7401 \pm .0164\bullet$	$.8282 \pm .0067\bullet$	$.8359 \pm .0068\bullet$	$.8779 \pm .0069\circ$
mnist1v7	$.9752 \pm .0031$	$.0000 \pm .0000\bullet$	$.0000 \pm .0000\bullet$	$.7897 \pm .1170\bullet$	$.9600 \pm .0032\bullet$	$.9668 \pm .0031\bullet$	$.9781 \pm .0028$
Average	$.7918 \pm .1134$	$.0943 \pm .1545$	$.2512 \pm .2279$	$.5499 \pm .2276$	$.6697 \pm .1763$	$.7492 \pm .1422$	—
Win/Tie/Loss		**18/0/0**	**18/0/0**	**18/0/0**	**18/0/0**	**15/3/0**	**9/7/2**

Our PRAdaBoost: significantly better than other unprovable methods

comparable with PRBoosting yet with smaller time

Running time





Our approaches: faster than other provable robust methods

comparable with unprovable robust methods

Conclusion

In this work, we propose

- **Fast Provably Robust Decision Tree (FPRDT)**
  adversarial 0/1 loss, smallest computational complexity
- **Provably Robust AdaBoost (PRAdaBoost)**
  upper bound on adversarial exp. loss, smallest comp. complexity

Future work: other adversarial losses for robust decision trees

Thanks!