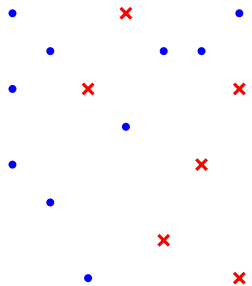
On Computing Optimal Tree Ensembles

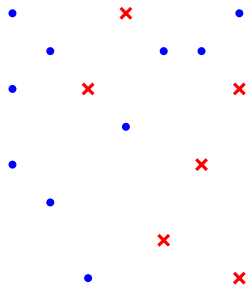
Christian Komusiewicz[1], Pascal Kunz[2,3], Frank Sommer[1], and Manuel Sorge[4]

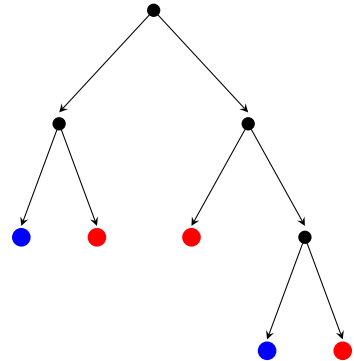
[1]Friedrich-Schiller-Universität Jena, Germany
[2]TU Berlin, Germany
[3]HU Berlin, Germany
[4]TU Wien, Austria

ICML 2023

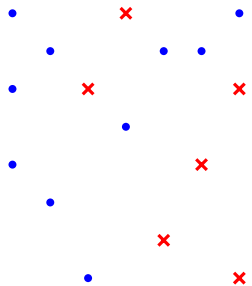


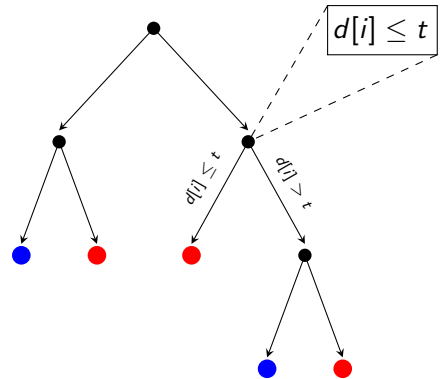
Input: Training data set of n points in d dimensions and two classes red and blue.
Task: Compute a decision tree of minimal size that classifies each example correctly.
(no misclassifications allowed)



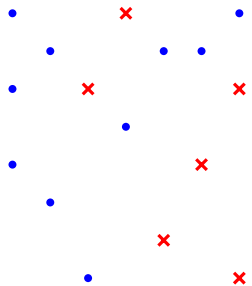


Input: Training data set of n points in d dimensions and two classes red and blue.
Task: Compute a decision tree of minimal size that classifies each example correctly. (no misclassifications allowed)

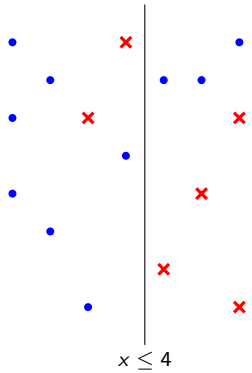


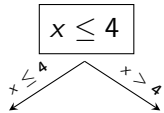


Input: Training data set of n points in d dimensions and two classes red and blue.
Task: Compute a decision tree of minimal size that classifies each example correctly. (no misclassifications allowed)



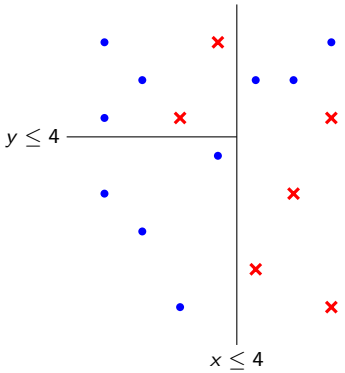
Input: Training data set of n points in d dimensions and two classes red and blue.
Task: Compute a decision tree of minimal size that classifies each example correctly.
(no misclassifications allowed)

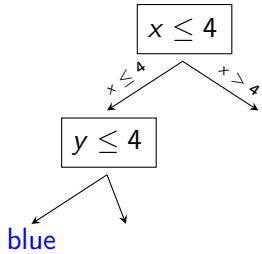




Input: Training data set of n points in d dimensions and two classes red and blue.

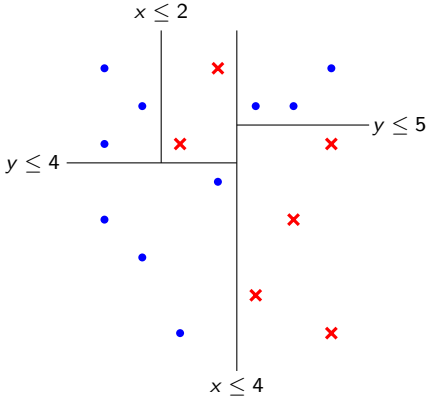
Task: Compute a decision tree of minimal size that classifies each example correctly. (no misclassifications allowed)

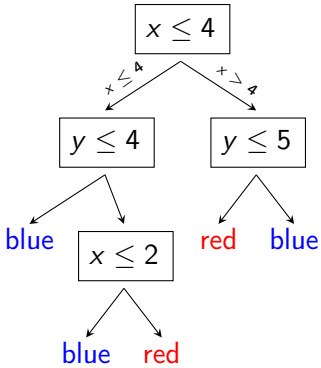




Input: Training data set of n points in d dimensions and two classes red and blue.

Task: Compute a decision tree of minimal size that classifies each example correctly. (no misclassifications allowed)





Input: Training data set of n points in d dimensions and two classes red and blue.

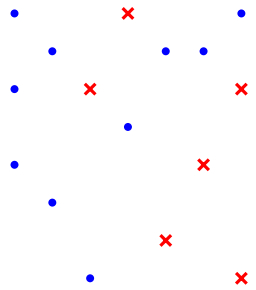
Task: Compute a decision tree of minimal size that classifies each example correctly. (no misclassifications allowed)

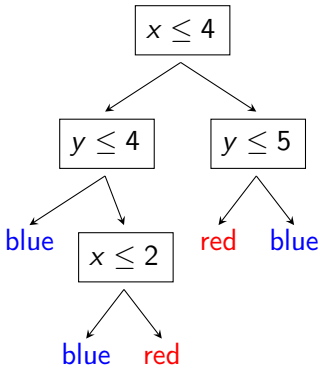
Theorem (*Hyafil and Rivest, Inf. Process. Lett. 1976*)

Computing an optimal decision tree is NP-hard.

Theorem (*Hyafil and Rivest, Inf. Process. Lett. 1976*)

Computing an optimal decision tree is NP-hard.



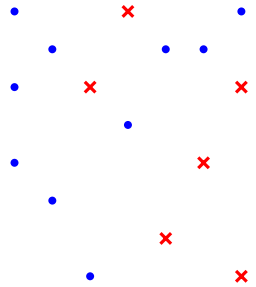


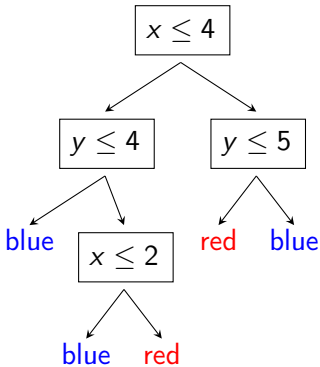
Parameters

s: Size of decision tree (# inner nodes) **4**

Theorem (*Hyafil and Rivest, Inf. Process. Lett. 1976*)

Computing an optimal decision tree is NP-hard.





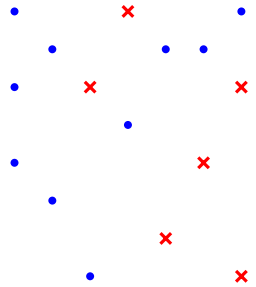
Parameters

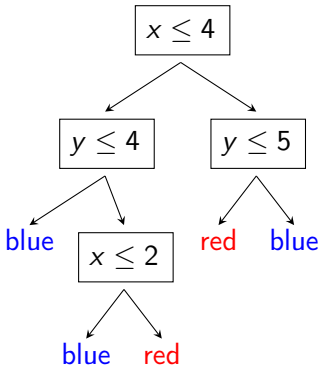
s: Size of decision tree (# inner nodes) **4**

D: Maximal domain size (maximal # different values in one dimension) **8**

Theorem (*Hyafil and Rivest, Inf. Process. Lett. 1976*)

Computing an optimal decision tree is NP-hard.





Parameters

s: Size of decision tree (# inner nodes)	**4**
D: Maximal domain size (maximal # different values in one dimension)	**8**
$\boldsymbol{\delta}$: Maximal # of dimensions in which a red and a blue example differ	**2**

Computing Optimal Decision Trees

Parameters

- s: Size of decision tree (# inner nodes)
- D: Maximal domain size (maximal # different values in one dimension)
- δ: Maximal # of dimensions in which a red and a blue example differ

Computing Optimal Decision Trees

Parameters

s: Size of decision tree (# inner nodes)
D: Maximal domain size (maximal # different values in one dimension)
δ: Maximal # of dimensions in which a red and a blue example differ

Theorem (*Ordyniak and Szeider, AAAI 2021*)

An optimal decision tree can be computed in $\Omega(\delta^s \cdot (D^s 2\delta)^s \cdot 2^{s^2})$ time.

Computing Optimal Decision Trees

Parameters

- s: Size of decision tree (# inner nodes)
- D: Maximal domain size (maximal # different values in one dimension)
- δ: Maximal # of dimensions in which a red and a blue example differ

Theorem (*Ordyniak and Szeider, AAAI 2021*)

An optimal decision tree can be computed in $\Omega(\delta^s \cdot (D^s 2\delta)^s \cdot 2^{s^2})$ time.

Theorem (*Our Work*)

An optimal decision tree can be computed in $\mathcal{O}((6 \cdot \delta D s)^s sn)$ time.

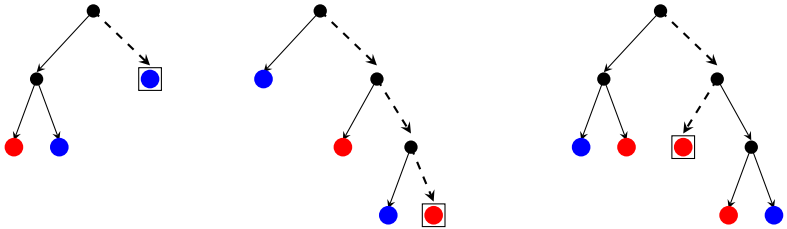
Outlook

Our algorithm with running time $\mathcal{O}((6 \cdot \delta Ds)^s sn)$ also works for Tree Ensembles!

Outlook

Our algorithm with running time $\mathcal{O}((6 \cdot \delta Ds)^s sn)$ also works for Tree Ensembles!

Tree Ensembles: Majority vote of ℓ decision trees



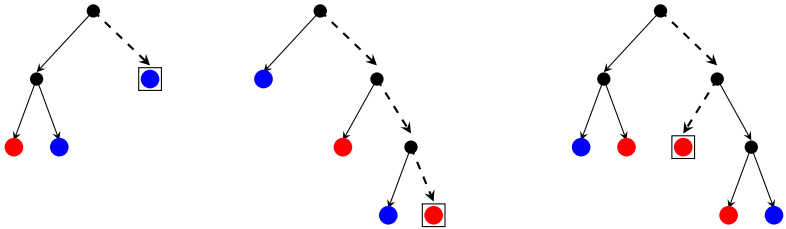
$\ell = 3$

example classified as red

Outlook

Our algorithm with running time $\mathcal{O}((6 \cdot \delta Ds)^s sn)$ also works for Tree Ensembles!

Tree Ensembles: Majority vote of ℓ decision trees



$\ell = 3$

example classified as red

Future Work

- Algorithm engineering for our algorithm for decision trees
- Study variant which allows misclassifications