Quant-BnB: A Scalable Branch-and-Bound Method for Optimal Decision Trees

Haoyue Wang and Xiang Meng

Massachusetts Institute of Technology

Joint work with **Rahul Mazumder**

July, 2022

Decision Tree Model

- Data: $\{(x_1, y_1), \ldots, (x_n, y_n)\}$

  $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,p}) \in \mathbb{R}^p$

  $y_i \in \{1, 2, \ldots, C\}$ is a label
- Decision tree model
  - branch nodes split data via rules $x_{i,f} \leq t$
  - leaf nodes make predictions
- Usually fitted greedily (CART)
- Building blocks of boosting, random forest, etc.

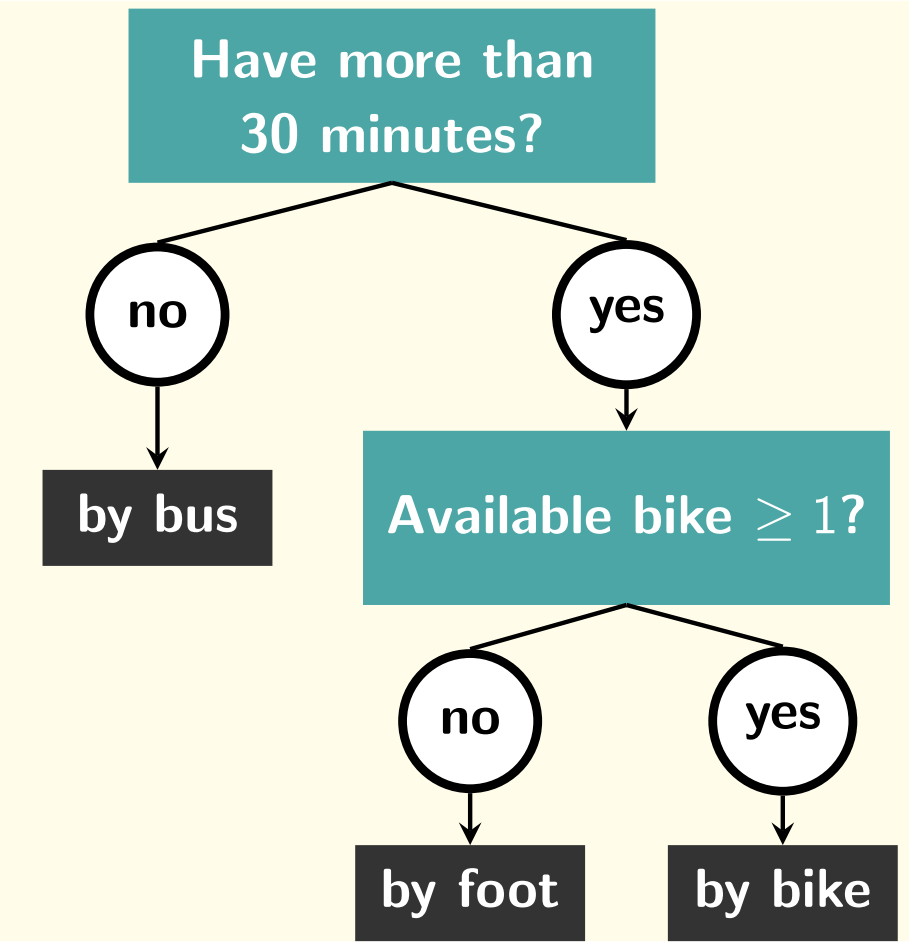

Figure: Depth-2 classification tree for commuting to work

Optimal decision tree

We focus on the optimization of a single tree:

$$\min_{T \in \mathcal{T}^d} \sum_{i=1}^{n} \mathbf{1}_{y_i \neq T(x_i)}$$

where $\mathcal{T}^d$ is the set of all binary trees with depth d.

- We focus on small d ($2 \sim 3$), large n ($10^4 \sim 10^5$), continuous data x_i.
- Brute-force search takes $O(n^d p^d)$ time.
- Existing algs: Mixed-integer programming; dynamic programming.

Quant-BnB

Quant-BnB: A Branch-and-Bound algorithm for optimal decision tree

Ideas

- Decompose the search space based on quantiles of the data.
- Novel lower bounding and upper bounding techniques.

Accuracy

- Guaranteed (training) optimality
- Improved (testing) accuracy than CART

Speed

- Efficient for small d, large n, continuous data.
- Faster than existing optimal tree algorithms

Optimizing depth-2 tree

Goal: find optimal tree $T \in \mathcal{T}^2$ with 3 branch nodes

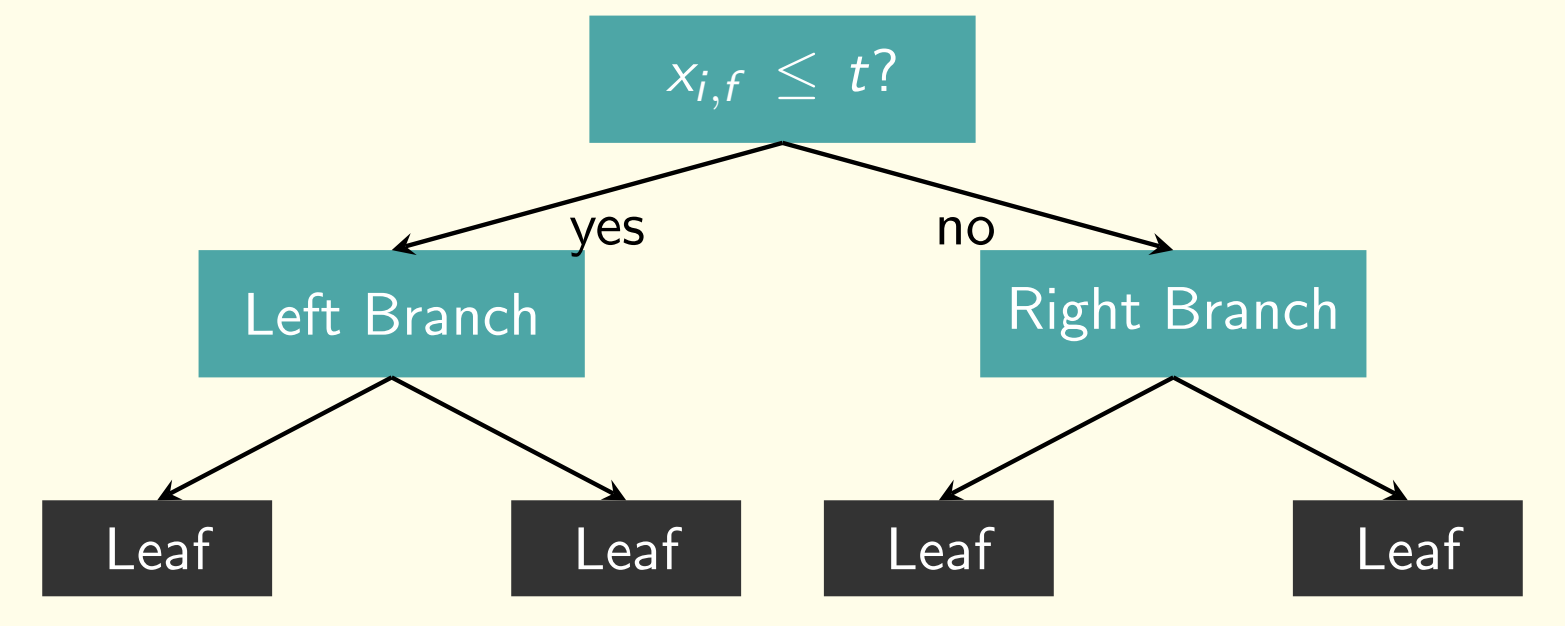

Figure: A illustration of elements in $\mathcal{T}^2$

Quantile-based Branch and Bound

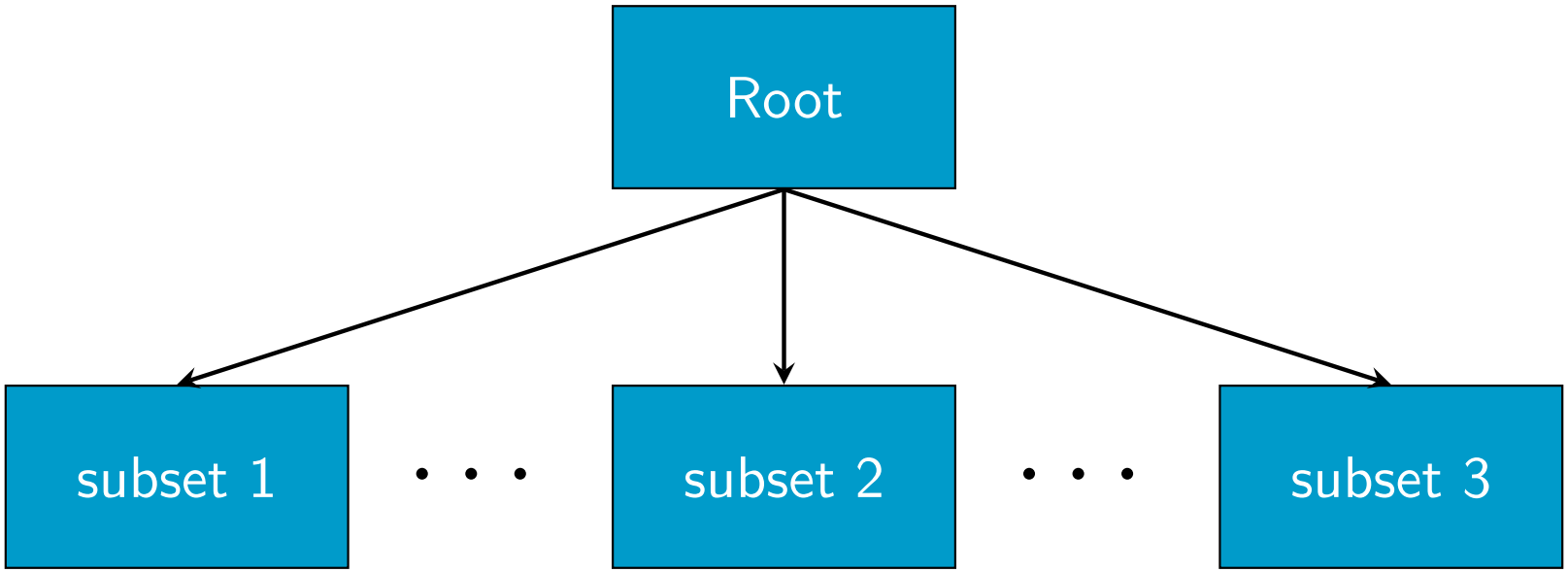

Figure: Branch and Bound framework

Quantile-based Branch and Bound

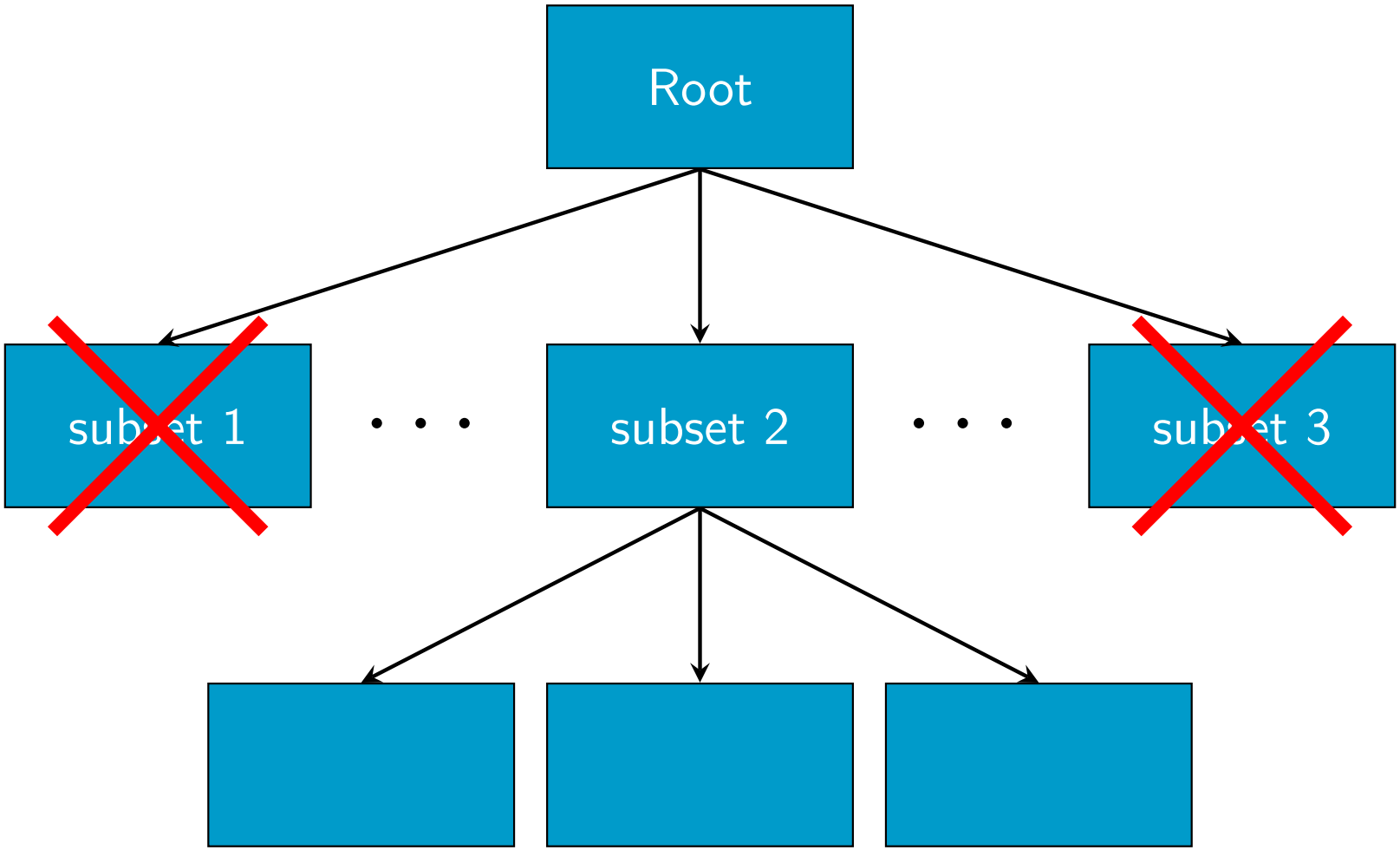

Figure: Branch and Bound framework

Quantile-based Branch and Bound

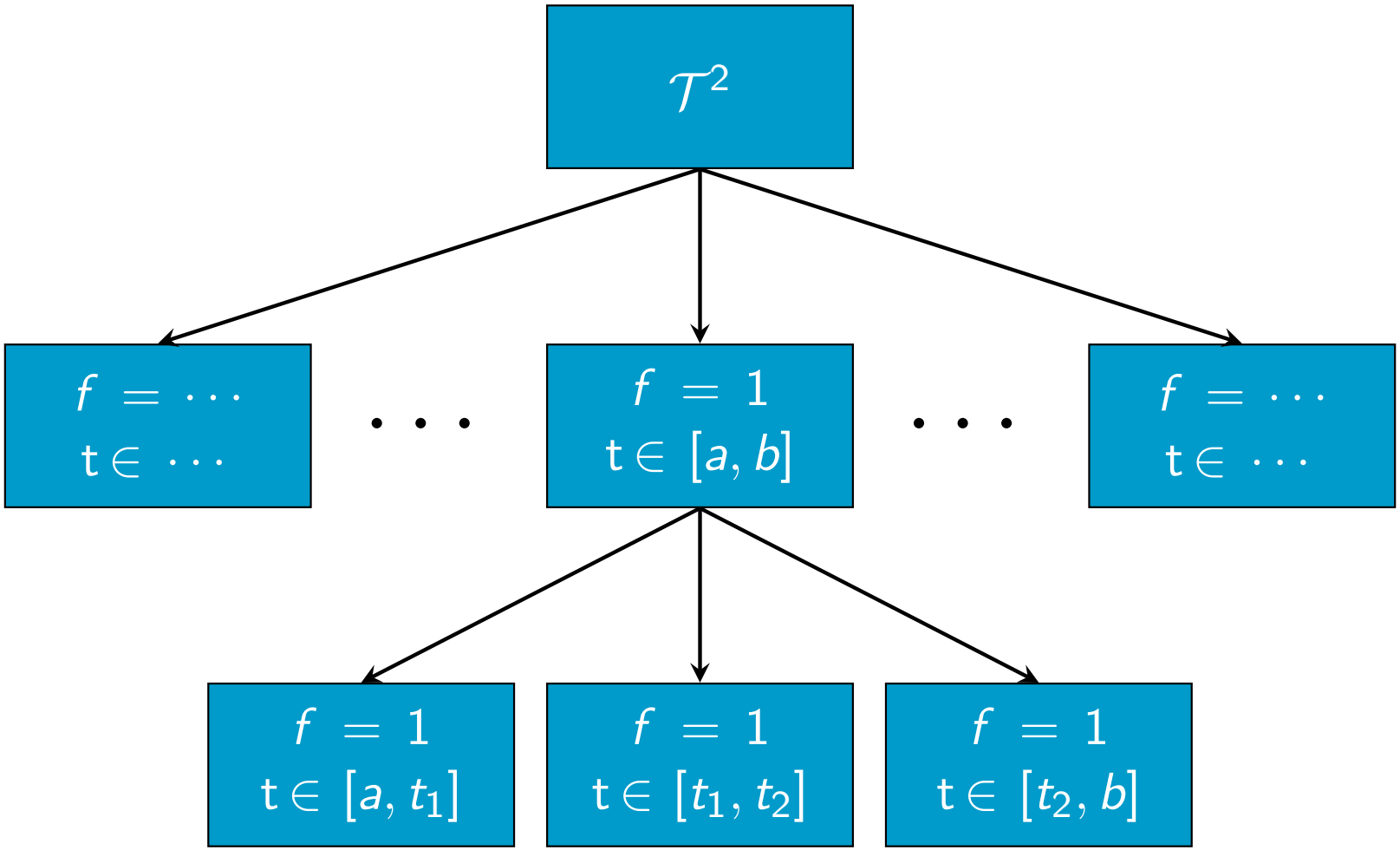

Figure: Quantile-based Branch and Bound

Optimal Classification Tree with Depth 2

Name	(n,p)	Quant-BnB	BinOCT	MurTree	DL8.5
avila	(10430,10)	**4.5**	-	OoM	3278
bank	(1097,4)	<**0.1**	2963	8.4	4.6
bean	(10888,16)	**3.4**	-	OoM	OoM
bidding	(5056,9)	**0.2**	-	345	72
eeg	(11984,14)	**2.9**	-	288	34
fault	(1552,27)	**1.6**	-	530	271
htru	(14318,8)	**1.3**	-	OoM	OoM
magic	(15216,10)	**1.0**	-	OoM	OoM
occupancy	(8143,5)	**0.3**	-	193	33
page	(4378,10)	**0.4**	-	155	84
raisin	(720,7)	**0.1**	9590	13	6.2
rice	(3048,7)	**0.4**	-	591	267
room	(8103,16)	**1.0**	-	18	14
segment	(1848,18)	**1.1**	-	389	213
skin	(196045,3)	**2.3**	-	37	16
wilt	(4339,5)	**0.2**	-	653	314

Table: Each entry denotes running time in seconds. - refers to time out (4h), OoM refers to out of memory (25GB).

Test accuracy compared with CART

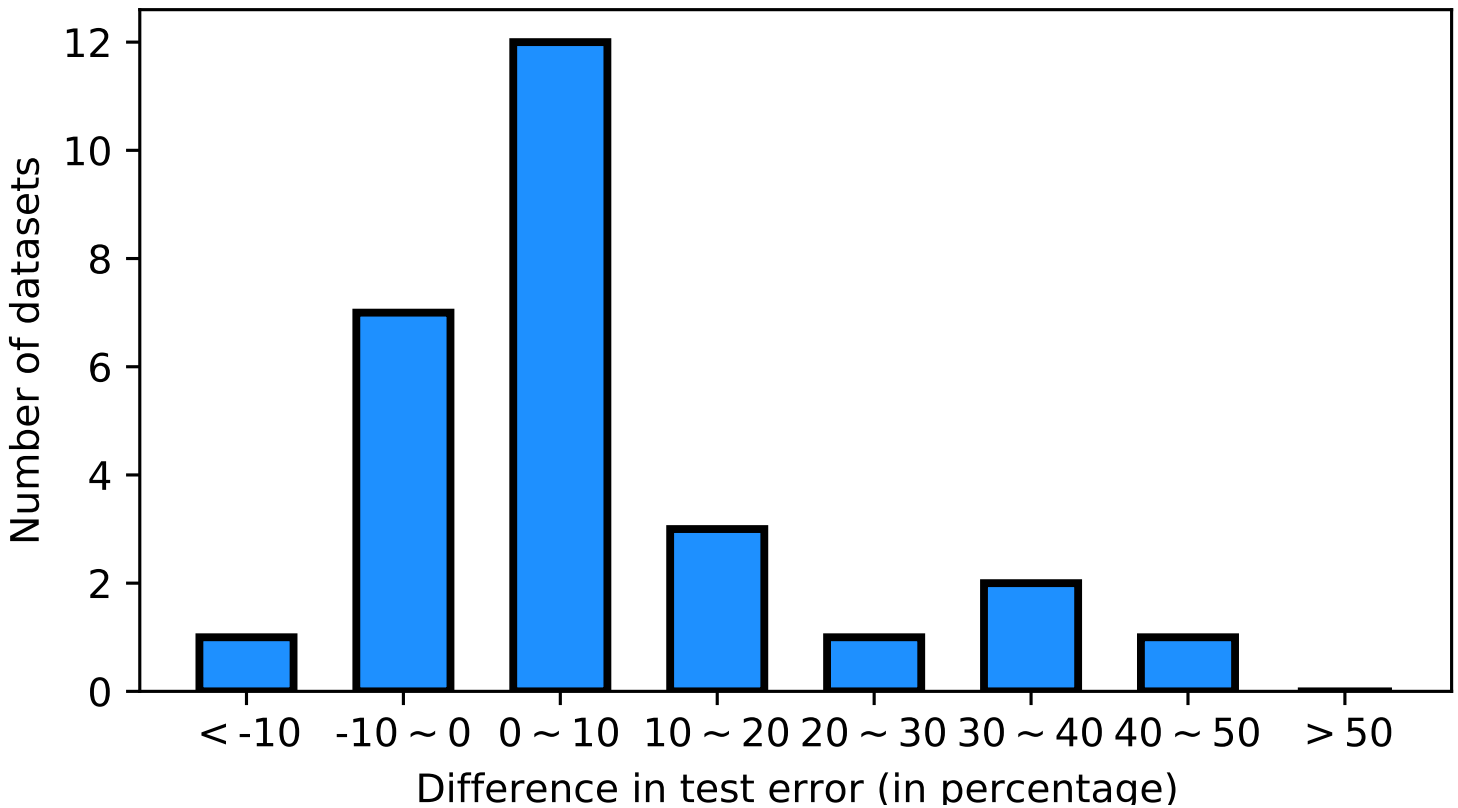

Figure: Each entry denotes the relative difference values between CART and optimal trees delivered by Quant-BnB, shown in percentages using bar charts.