# Recap

09 April 2024 08:08

- Xgboost
  - DSMP 1
    - Intro
      - performance
      - speed
    - Reg
    - Class
    - Maths
  - DSMP 2
    - Regularization
    - Optimizations
      - missing values

# Exact Greedy Split Finding

09 April 2024 15:52

xgboost

| cgpa | package | pred | res1 |
|---|---|---|---|
| 5 | 5 | 13 | 8 |
| 6 | 13 | 13 | 0 |
| 7 | 14 | 13 | -1 |
| 8 | 9 | 13 | 4 |
| 9 | 15 | 13 | -2 |

f1 → cgpa (5)   200   4

→ cgpa 6.5   ss

8 0   ss      -1 4 2   ss

gain

best splitting

$\boxed{m1}$ + $\boxed{m2}$

avg     dt

m cols

gain = 100

$\boxed{n}$ data

$\boxed{n-1}$

cgpa < 5.5   $ss_p$   gain →   5.5

8   $ss_l$      0 -1 4 -2   $ss_r$

$\boxed{gain = ss_l + ss_r - ss_p}$

$\boxed{m(n-1)}$

100000 rows   100 cols

→ 100(99999)

exact greedy algorithm

Tradeoff

disadvantage ← advantage

slow

best splitting

↓

performance

best

# Approximate Method For Split Finding

09 April 2024 08:08

| cgpa | package |
|---|---|
| 4 | 5 |
| 5 | 13 |
| 6 | 14 |
| 7 | 9 |
| 8 | 15 |
| 9 | 4 |

(approx

3 quantiles (6)

exact greedy (5) | approx (2)

also speed up

approximations

adv: ↑ speed

disadvan: ↑ performance — increase the quantiles

quantiles

1 → approximate → basic idea

Approx

g - gradient

h → hessian

= 3

reg → $g_i = \hat{y}_i - y_i$

$G_1 = 5-3 = 2$

$G_2 = -4+1 = -3$

$G_3 = -5+6 = 1$

$H_1 = 2$

$H_2 = 2$

$H_3 = 2$

$G_1, H_1$ 1L data points $G_i$

| | cgpa | package | pred | $g_i$ | $h_i$ | |
|---|---|---|---|---|---|---|
| GH | 4 | 5 | 10 | 5 | 1 | $G_1$, $H_1$ |
| | 5 | 13 | 10 | -3 | 1 | |
| | 6 | 14 | 10 | -4 | 1 | $H_2$ |

4 6 7 8 RAM 10

$ss = \frac{(G_1+G_2+G_3)^2}{}$

| | | | | |
|---|---|---|---|---|
| 5 | 13 | 10 | | |
| 6 | 14 | 10 | -4 | 1 |
| 7 | 9 | 10 | 1 | 1 |
| 8 | 15 | 10 | -5 | 1 |
| 9 | 4 | 10 | 6 | 1 |

GH

GH.

$G_2$ $H_2$

$G_3$ $H_3$

$\boxed{\lambda = 0}$

$2 \to$ sum of gradients

if cgpa < 6.5

$$SS = \frac{(G_1 + G_2 + G_3)^2}{H_1 + H_2 + H_3}$$

$$SS = \frac{(G_1)^2}{H_1}$$

$$SS = \frac{(G_2 + G_3)^2}{H_1 + H_2}$$

gain

step 1 → No. of quantiles (bins) buckets

Step 2 → we calculate g and h for every data point

$(\hat{y}_i) = 1$

step 3 → histogram grad / hessian based on the quantiles

$$\begin{bmatrix} G_1 & G_2 & G_3 \\ H_1 & H_2 & H_3 \end{bmatrix}$$

Step 3a → store → cache memory

Step 4 → Try out splitting criteria

< 500MB parallel

cores cpu

distributed computing

100GB

if cgpa < 7.5

$$SS = \frac{(G_1 + G_2)^2}{H_1 + H_2}$$

$$SS = \frac{(G_3)^2}{H_3}$$

$g_i$ $h_i$

$G_i$ $H_i$

$\boxed{\text{fast}}$

Cache Memory →

fastest

CPU

1TB

Hard disk

ROM

storage

cheaper

8GB

RAM

faster

costlier

cache memory

fastest

6-12mb

1) reduced # splits → quantiles → histo

2) calculat $G_1 \dots G_n$ $H_1 \dots H_n$

3) pa ?

4) cache

XG Boost

xtreme

# Improvement 1 - Parallel Processing

09 April 2024 08:08

# Improvement 2 - Cache Storage

09 April 2024 15:51

Improvement 3 Quantile Sketch

09 April 2024 15:51

step 1 → [No. of quantiles (bins) bucket]

→ quantiles → 3 → 3 bins

huge (single computer) distributed

Step2 → we calculate g and h for everydata point $L(\hat{y}_i) = 1$

Step3 → histogram grad / hessian based on the quantiles

$G_1$ $G_2$ $G_3$

$H_1$ $H_2$ $H_3$

Step3a → store cache memory

Step4 → Try out splitting criteria

quantiles multiple comp

distributed computation

100GB →

RAM

5Gr

rows

A sketch is a type of data structure or algorithm that processes large data streams or datasets to create a summary or approximation, allowing for efficient computations while using limited memory. The concept is particularly useful in scenarios where it's impractical to store or process the entire dataset due to resource constraints.

A quantile sketch estimates the quantiles of a dataset without the need to store or sort the entire dataset.

approx

1. Efficiency: They enable efficient computation of approximate quantiles without sorting the entire dataset, reducing time and computational resources.
2. Scalability: Quantile sketches help XGBoost efficiently handle large datasets by summarizing data distributions compactly, which is crucial for processing big data.
3. Memory Usage: They reduce memory requirements by providing a summarized representation of the data, avoiding the need to store all individual data points.
4. Split Selection: Quantile sketches allow for effective and approximate identification of potential split points in the data, facilitating the decision-making process in tree construction.
5. Distributed Processing: In distributed environments, quantile sketches can be combined across different data subsets, enabling XGBoost to effectively approximate split points even when data is scattered across multiple machines.

5.8

5.5

5 optimization → huge datasets

quantile sketch

streaming data → data

sample → pop quantile

# Improvement 4 - Weighted Quantile Sketch

09 April 2024 15:51

- Purpose: Like a standard quantile sketch, a weighted quantile sketch provides a way to estimate quantiles. However, it accounts for weights associated with each data point, offering a more nuanced summary of the dataset.
- Usage: This is particularly useful in datasets where some observations are more important or occur more frequently than others, and this significance is represented by weights.

weighted $\boxed{\text{Quantile sketch}}$

$\boxed{\text{cgpa}}$ | package → $\boxed{\text{10 quantiles}}$ ↓ 10 bins

r1 4
r2 5 $\boxed{\text{1cr}}$ $\boxed{\text{1cr}}$
r3 6
r4 9 $\boxed{\text{10 Cr}}$
2

r10cr

1 bin → ? $\frac{10 Cr}{100}$

binning → weightage

2 5kg, 2 5kg, 2 5kg, 2 5kg, 2

(15) (2) (1) (1) (1)

Sum of weights → Sum of wei, sum of w

10 points

weighted quantile

$\boxed{\text{normal quantile}}$ classif vs weighted quantile w

weight 0.7(1-0.7)

0.7(1-0.7)
0.1(1-0.1)
0.5(1-0.5)
0.25

weight

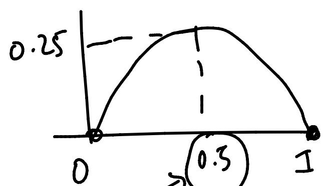
| cgpa | placement | pred | grad | $\boxed{\text{hessia}}$ |
|---|---|---|---|---|
| 3 | 1 | 0.7 | -0.3 | 0.21 |

| | | | | | |
|---|---|---|---|---|---|
| 3 | 1 | 0.7 | -0.3 | 0.21 | weight |
| 4 | 0 | 0.2 | 0.2 | 0.16 | weight |
| 6 | 0 | 0.5 | 0.1 | 0.09 | |
| 7 | 1 | 0.8 | -0.2 | 0.16 | |
| 4 | 0 | 0.3 | 0.3 | 0.21 | |
| 8 | | | | | |

min child weight → cover → sum of hessians

↓

$h_i = p_i(1-p_i)$




unsure → hessian → weightage

weighted quantile sum of hessians

| $h_i$ | | $x_i$ | $h_i$ | |
|---|---|---|---|---|
| 1 | 2 | 3 | 0.21 | → 1 pt → w |
| 1 | | 4 | 0.16 | 3 points → w |
| 1 | 2 | 6 | 0.09 | |
| 1 | | 7 | 0.16 | |
| 1 | 2 | 4 | 0.21 | 2 points → w |
| 1 | | 8 | 0.25 | |

weights quantile → 3

↓

hessians

↓

weightage of data points

# Code

09 April 2024 15:52