LightGBM

Ensemble Learning: Light GBM

Pilsung Kang
School of Industrial Management Engineering
Korea University

<Boosting>

① Adaboost

② GBM

↓ scale up, Efficiency

③ XGBoost

④ Light GBM

⑤ Cat GBM

Light GBM

Ke et al. (2017)

- Motivation
  - ✓ Conventional GBM need to, for every feature, scan all the data instances to estimate the information gain of all the possible split points

    ↳ XGBoost에서는 전체 스캔을 최적화

    Light GBM: 전체 스캔 X

- Idea
  - ✓ To reduce the number of data instances and the number of features
    - Gradient-based One-Side Sampling (GOSS)
      - Data instances with different gradients play different roles in the computation of information gain
      - Keep instances with large gradients and randomly drop instances with small gradients

      Gradient가 높은 Instance만 상위 %기준 Sampling
    - Exclusive Feature Bundling (EFB)
      - In a sparse feature space, many features are (almost) exclusive, i.e., they rarely take nonzero values simultaneously (ex: one-hot encoding)

        ↳ 하나의 객체에 대해 특정한 두개의 변수가 동시에 영향을 가질 확률↓
      - Bundling these exclusive features does not degenerate the performance

      Risk : Exclusive하지 않은 케이스 존재 가능

Light GBM

- Gradient-based One-sided Sampling (GOSS)

XGBoost

Bucket기준 Split Point 탐색

Algorithm 1: Histogram-based Algorithm

Input: I: training data, d: max depth
Input: m: feature dimension
$nodeSet \leftarrow \{0\}$ ▷ tree nodes in current level
$rowSet \leftarrow \{\{0, 1, 2, ...\}\}$ ▷ data indices in tree nodes
for $i = 1$ ***to*** d **do**
  for $node$ ***in*** $nodeSet$ **do**
    usedRows $\leftarrow rowSet[node]$
    for $k = 1$ ***to*** m **do**
      $H \leftarrow$ new Histogram()
      ▷ Build histogram
      for j ***in*** $usedRows$ **do**
        bin $\leftarrow I$.f[k][j].bin
        H[bin].y $\leftarrow H$[bin].y + I.y[j]
        H[bin].n $\leftarrow H$[bin].n + 1
      Find the best split on histogram H.
      ...
  Update $rowSet$ and $nodeSet$ according to the best split points.
  ...

LightGBM

Algorithm 2: Gradient-based One-Side Sampling

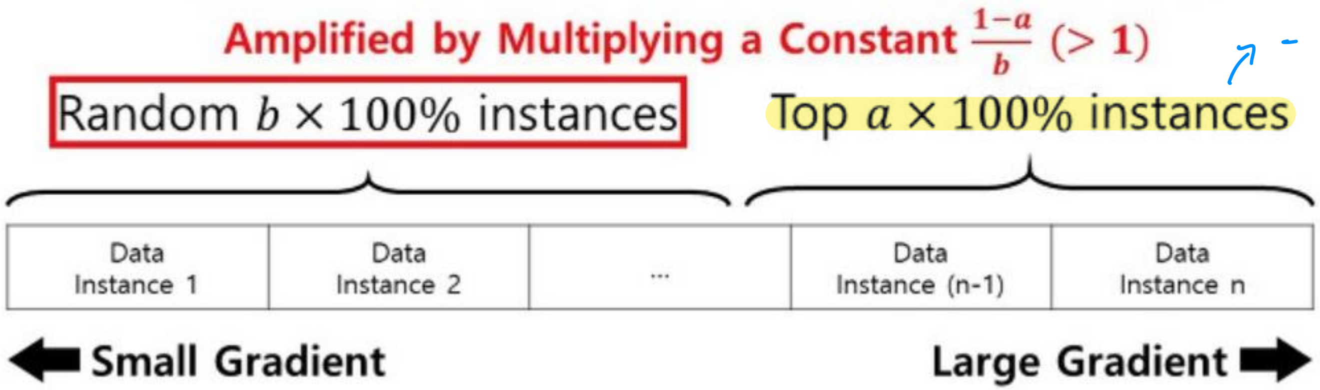
Input: I: training data, d: iterations
Input: a: sampling ratio of large gradient data
Input: b: sampling ratio of small gradient data
Input: $loss$: loss function, L: weak learner
models $\leftarrow$ {}, fact $\leftarrow \frac{1-a}{b}$
topN $\leftarrow$ a $\times$ len(I) , randN $\leftarrow$ b $\times$ len(I)
for $i = 1$ ***to*** d **do**
  preds $\leftarrow$ models.predict(I)
  g $\leftarrow loss$(I, preds), w $\leftarrow$ {1,1,...}
  sorted $\leftarrow$ GetSortedIndices(abs(g))
  topSet $\leftarrow$ sorted[1:topN]
  randSet $\leftarrow$ RandomPick(sorted[topN:len(I)], randN)
  usedSet $\leftarrow$ topSet + randSet
  w[randSet] $\times$= fact ▷ Assign weight $fact$ to the small gradient data.
  newModel $\leftarrow$ L(I[usedSet], $-$ g[usedSet], w[usedSet])
  models.append(newModel)

Light GBM

- Gradient-based One-sided Sampling (GOSS)

- Gradient가 높은 Instance

- 모두 사용



https://cdm98.tistory.com/m/31

$\frac{1-a}{b}$

e.g.

a : 상위 10%.

b : 하위 90%.

$\frac{1-a}{b} = \frac{0.9}{0.9} = 1$

a : 0.05

b : 0.5

$\frac{0.95}{0.5} = 19 > 1$

Light GBM

- Exclusive Feature Bundling (EFB)

$\{x_1, x_4, x_7\} \rightarrow y_1$

$\{x_2, x_5, x_6, x_9\} \rightarrow y_2$

Bundling이 되어야하는 변수들을 이용하여 하나의 변수로 값을 표현하고자 하는 과정

① 현재 존재하는 feature set에 대해 어떠한 feature를 하나의 번들로 정할 것인지

Algorithm 3: Greedy Bundling

Input: F: features, K: max conflict count
Construct graph G
searchOrder $\leftarrow$ G.sortByDegree()
bundles $\leftarrow$ { }, bundlesConflict $\leftarrow$ { }
for *i in searchOrder* **do**
  needNew $\leftarrow$ True
  for j = 1 *to len(bundles)* **do**
    cnt $\leftarrow$ ConflictCnt(bundles[j],F[i])
    if *cnt + bundlesConflict[i]* $\leq K$ **then**
      bundles[j].add(F[i]), needNew $\leftarrow$ False
      break
  if *needNew* **then**
    Add $F[i]$ as a new bundle to *bundles*
Output: *bundles*

②

Algorithm 4: Merge Exclusive Features

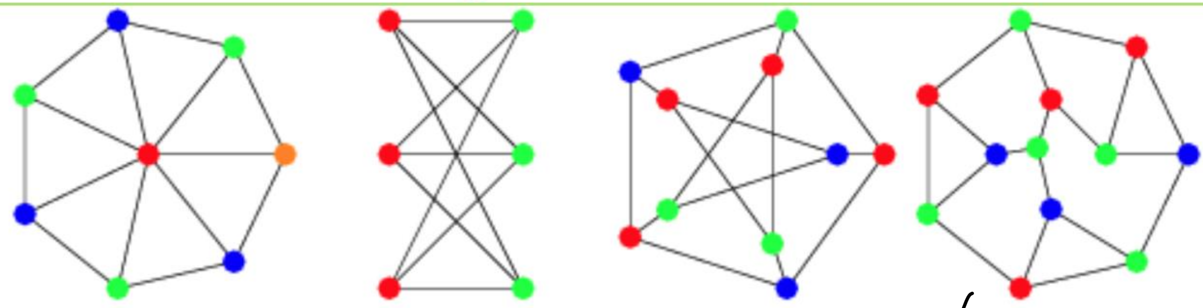
Input: *numData*: number of data
Input: F: One bundle of exclusive features
binRanges $\leftarrow$ {0}, totalBin $\leftarrow$ 0
for *f in F* **do**
  totalBin += f.numBin
  binRanges.append(totalBin)
newBin $\leftarrow$ new Bin(numData)
for i = 1 *to numData* **do**
  newBin[i] $\leftarrow$ 0
  for j = 1 *to len(F)* **do**
    if *F[j].bin[i]* $\neq$ *0* **then**
      newBin[i] $\leftarrow$ F[j].bin[i] + binRanges[j]
Output: *newBin*, *binRanges*

Light GBM

- Exclusive Feature Bundling (EFB)
  - ✓ Can be formulated as a Graph coloring problem
    - ▪ Construct a Graph (V, E)
      - V: feature
      - E: total conflicts between features

↑: 중복↑ → 선택 X ⇒ Conflict이 없는 경우,
하나의 bundle로 묶음.

Minimum Vertex Coloring



: 1st Bundle

: 2nd Bundle

: 3rd Bundle

: 4th Bundle

'Reduction'

8 → 4 bundle

2 Bundle

3 Bundle

3 Bandle

Light GBM

- Exclusive Feature Bundling (EFB)
  - ✓ Greedy bundling example

Instances \ features

	x_1	x_2	x_3	x_4	x_5
I_1	1	1	0	0	1
I_2	0	0	1	1	1
I_3	1	2	0	0	2
I_4	0	0	2	3	1
I_5	2	1	0	0	3
I_6	3	3	0	0	1
I_7	0	0	3	0	2
I_8	1	2	3	4	3
I_9	1	0	1	0	0
I_{10}	2	3	0	0	2

Edge의 강도 = Conflict의 정도 → 동시에 0이 아닌 객체의 수로 결정

	x_1	x_2	x_3	x_4	x_5
x_1	-	6	2	1	6
x_2	6	-	1	1	6
x_3	2	1	-	3	4
x_4	1	1	3	-	3
x_5	6	6	4	3	-
	15	14	10	8	19

	x_5	x_1	x_2	x_3	x_4
d	19	15	14	10	8

degree → greedy Method는 시작점이 필요

시작점은 degree를 활용

Light GBM

- Exclusive Feature Bundling (EFB)
  - ✓ Greedy bundling example (cut-off = 0.2)

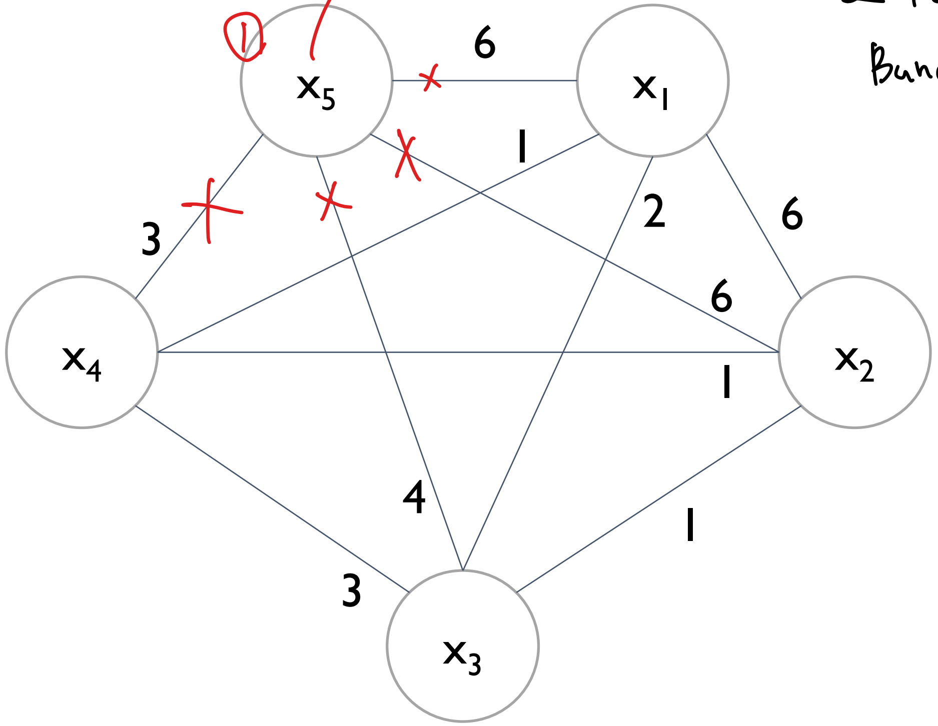
Hyperparameter

N=10 → 10×0.2 = 2

0이 아닌 Value가 2 이상일 때 Bunding을 진행 X

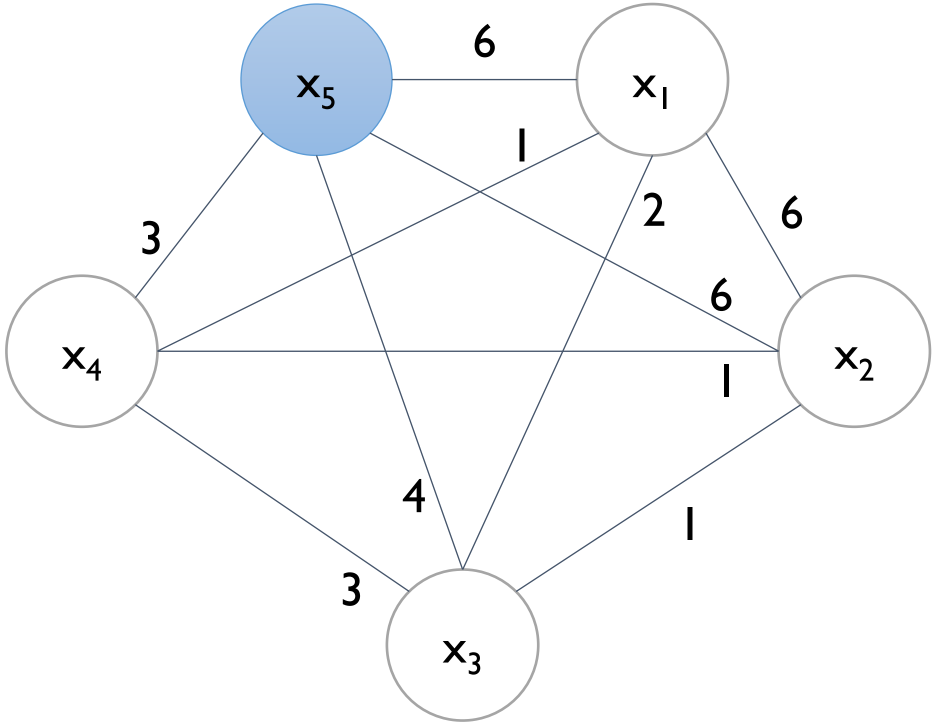
↳ 거리가 짧아야하기 때문

degree가 가장 높음



Light GBM

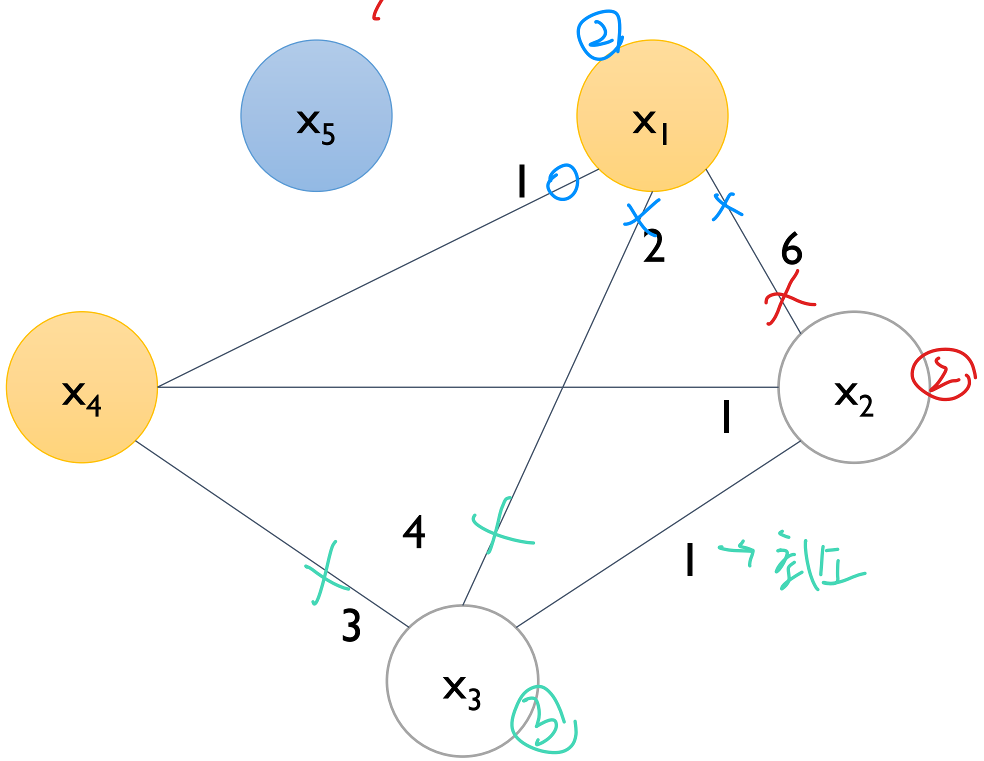
- Exclusive Feature Bundling (EFB)
  - ✓ Greedy bundling example (cut-off = 0.2)



Light GBM

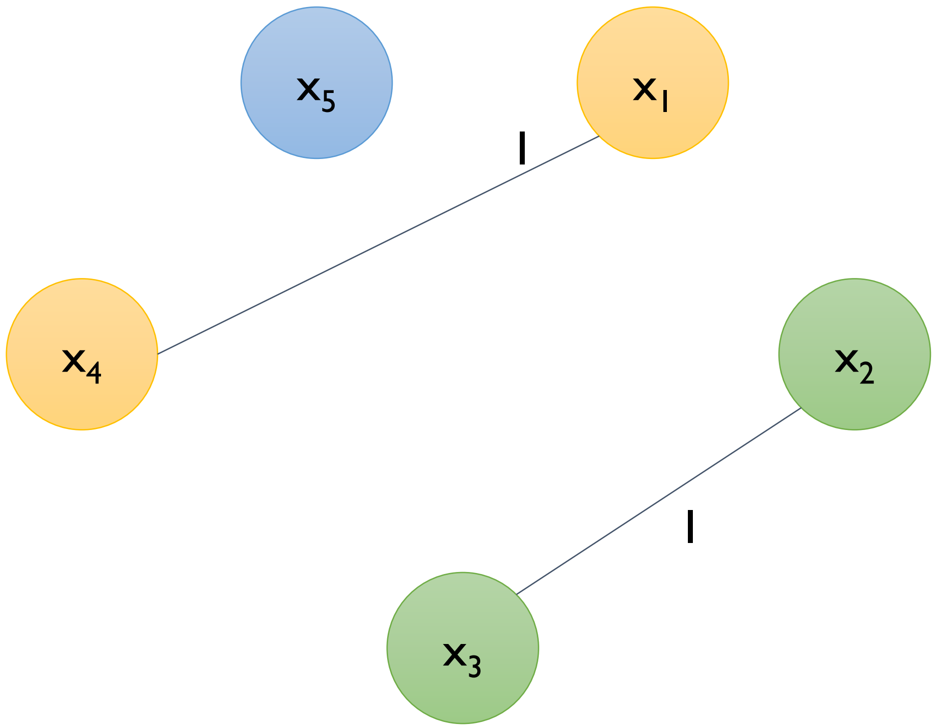
- Exclusive Feature Bundling (EFB)
  - ✓ Greedy bundling example (cut-off = 0.2)

Cutoff 0.2기준 x5는 모두 2이상이므로, '고립'



Light GBM

- Exclusive Feature Bundling (EFB)
  - ✓ Greedy bundling example (cut-off = 0.2)



X_1, X_2, X_3, X_4, X_5

$\Downarrow$

(X_1, X_4)

(X_2, X_3) → 3개의 Bundle 형성

(X_5)

Greed Bundling의 효과

Light GBM

- Exclusive Feature Bundling (EFB)
  - ✓ Greedy bundling example (cut-off = 0.2)

Merge를 위해
Column 조정

	x_1	x_2	x_3	x_4	x_5
I_1	1	1	0	0	1
I_2	0	0	1	1	1
I_3	1	2	0	0	2
I_4	0	0	2	3	1
I_5	2	1	0	0	3
I_6	3	3	0	0	1
I_7	0	0	3	0	2
I_8	1	2	3	4	3
I_9	1	0	1	0	0
I_{10}	2	3	0	0	2

	x_5	x_1	x_4	x_2	x_3
I_1	1	1	0	1	0
I_2	1	0	1	0	1
I_3	2	1	0	2	0
I_4	1	0	3	0	2
I_5	3	2	0	1	0
I_6	1	3	0	3	0
I_7	2	0	0	0	3
I_8	3	1	4	2	3
I_9	0	1	0	0	1
I_{10}	2	2	0	3	0

Light GBM

- Exclusive Feature Bundling (EFB)

  ✓ Exclusive feature merging

  Bunding을 하기 위한 대상이 되는 변수에 원래 기준이 되는 변수의 최대값을 더한다

  - Add offsets to the original values of the features

x_1 기준 $\qquad$ x_2 기준

	x_5	x_1	x_4	x_2	x_3
I_1	1	1	0	1	0
I_2	1	0	1	0	1
I_3	2	1	0	2	0
I_4	1	0	3	0	2
I_5	3	2	0	1	0
I_6	1	3	0	3	0
I_7	2	0	0	0	3
I_8	3	1	4	2	3
I_9	0	1	0	0	1
I_{10}	2	2	0	3	0
Min		0	0	0	0
Max		3	4	3	3

	x_5	x_{14}	x_{23}
I_1	1	1	1
I_2	1	4	4
I_3	2	1	2
I_4	1	6	5
I_5	3	2	1
I_6	1	3	3
I_7	2	0	6
I_8	3	1	2
I_9	0	1	4
I_{10}	2	2	3

Add the offset 3 to the nonzero values of x_4

Add the offset 3 to the nonzero values of x_3

Conflict시, 기준변수 값 활용

Conflict: Use the value of x_1

Conflict: Use the value of x_2

데이터의 손실이 발생

Light GBM

- Experiments
  - ✓ Dataset description

Table 1: Datasets used in the experiments.

Name	$\#data$	$\#feature$	Description	Task	Metric
Allstate	12 M	4228	Sparse	Binary classification	AUC
Flight Delay	10 M	700	Sparse	Binary classification	AUC
LETOR	2M	136	Dense	Ranking	NDCG [4]
KDD10	19M	29M	Sparse	Binary classification	AUC
KDD12	119M	54M	Sparse	Binary classification	AUC

  - ✓ Training time

	xgb_exa	xgb_his	lgb_baseline	EFB_only	**LightGBM**
Allstate	10.85	2.63	6.07	0.71	**0.28**
Flight Delay	5.94	1.05	1.39	0.27	**0.22**
LETOR	5.55	0.63	0.49	0.46	**0.31**
KDD10	108.27	OOM	39.85	6.33	**2.85**
KDD12	191.99	OOM	168.26	20.23	**12.67**

Light GBM

- Experiments
  - ✓ Overall accuracy

	xgb_exa	xgb_his	lgb_baseline	SGB	**LightGBM**
Allstate	0.6070	0.6089	0.6093	0.6064 ± 7e-4	**0.6093 ± 9e-5**
Flight Delay	0.7601	0.7840	0.7847	0.7780 ± 8e-4	**0.7846 ± 4e-5**
LETOR	0.4977	0.4982	0.5277	0.5239 ± 6e-4	**0.5275 ± 5e-4**
KDD10	0.7796	OOM	0.78735	0.7759 ± 3e-4	**0.78732 ± 1e-4**
KDD12	0.7029	OOM	0.7049	0.6989 ± 8e-4	**0.7051 ± 5e-5**

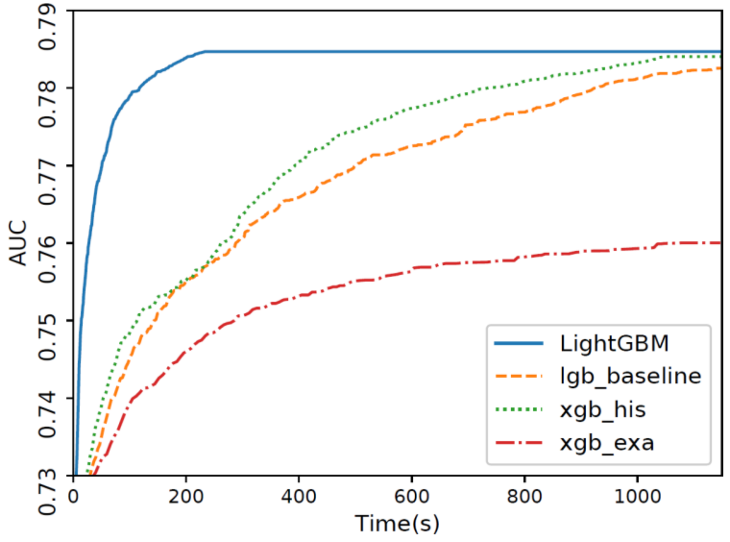


Figure 1: Time-AUC curve on Flight Delay.

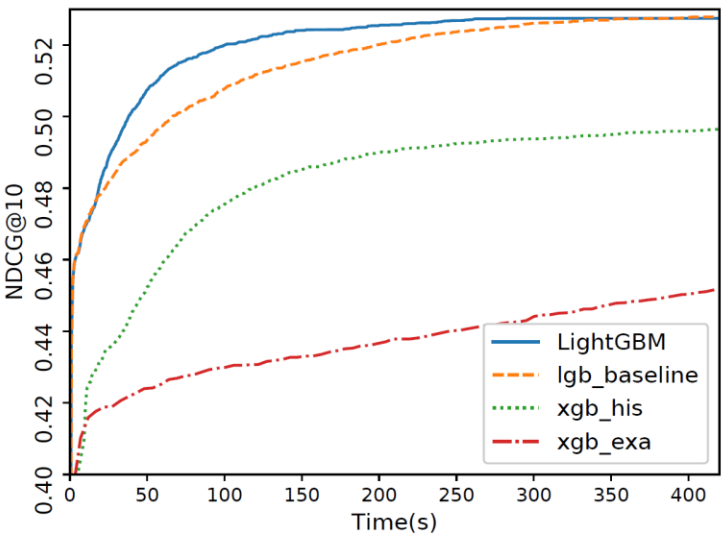


Figure 2: Time-NDCG curve on LETOR.


Q & A