




Ensemble Learning

담당자: 박무성



목차

1. XG Boost
2. Light Gradient Boost
3. Cat boost

목차

1. XG Boost
2. Light Gradient Boost
3. Cat boost

1. XG Boost의 직관적 이해

XG Boost는 Gradient Boost Machine이다!

-> GBM의 약점을 보완하고 더 빠른 학습을 위해 발전시킨 형태

Split Finding Algorithm

: “GBM처럼 하나하나 다 따져보지말고, 데이터를 분할해서 더 빠르게 최적의 분할 지점을 찾자”

Sparsity-Aware Split Algorithm

: “결측치, 0을 효과적으로 처리하자”

1. XG Boost의 직관적 이해

부스팅의 아이디어 : weak model을 boosting 시켜서 strong model로 제작

Weak Model : Decision Tree -> Bagging -> Strong Model : Random Forest

Weak Model : Linear Classifier -> Boosting -> Strong Model : Ada Boost

1. XG Boost의 직관적 이해

약한 분류기로 Decision Tree 사용

기존의 Gradient Boosting Machine

: 모든 경우의 수를 탐색하는거라서 뚱뚱하고 무거워

병렬처리가 불가능한 Sequential 방식이다보니

시간이 너무 오래 걸림

Decision Tree 복습

: Exact greedy algorithm

(**purity**를 높이는 가능한 모든 경우의 수 탐색)

Weak Model인 Decision Tree가 작동하는 방식을
약간 바꿔볼까...?

1. XG Boost의 직관적 이해

기존의 방식

: Basic exact greedy algorithm

“decision tree에서 최적의 분할점을 찾기위해 모든 경우의 수를 시도해보자“

Value																			
Lable	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

But,

만약 데이터가 메모리에 다 들어가지 않는다면...? (로딩되지 않는다면?)

하나하나 다 찾아봐야해서 병렬처리가 불가능

1. XG Boost의 직관적 이해

XG Boost의 아이디어

: “데이터를 분할해서 따로따로 최적의 분할 지점을 찾자”

For speed

Algorithm 2: Approximate Algorithm for Split Finding

for $k = 1$ **to** m **do**

 Propose $S_k = \{s_{k1}, s_{k2}, \dots, s_{kl}\}$ by percentiles on feature k .
 Proposal can be done per tree (global), or per split(local).

end

for $k = 1$ **to** m **do**

$G_{kv} \leftarrow \sum_{j \in \{j | s_{k,v} \geq \mathbf{x}_{jk} > s_{k,v-1}\}} g_j$
 $H_{kv} \leftarrow \sum_{j \in \{j | s_{k,v} \geq \mathbf{x}_{jk} > s_{k,v-1}\}} h_j$

end

Follow same step as in previous section to find max score only among proposed splits.

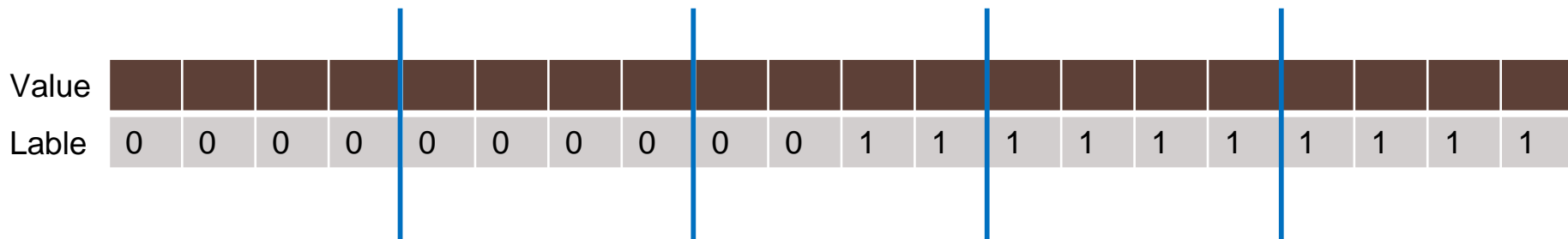
1. XG Boost의 직관적 이해

XG Boost의 아이디어

: “데이터를 분할해서 따로따로 최적의 분할 지점을 찾자”

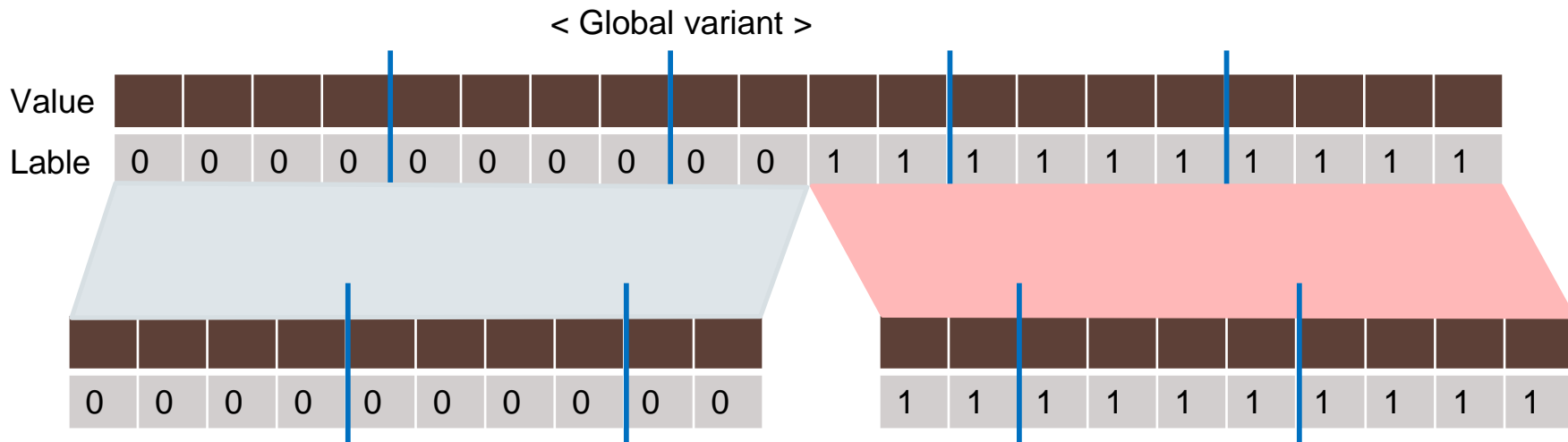
1) Value를 sort하자

2) Percentile에 따라서 n개의 bucket으로 전체 데이터를 나누자



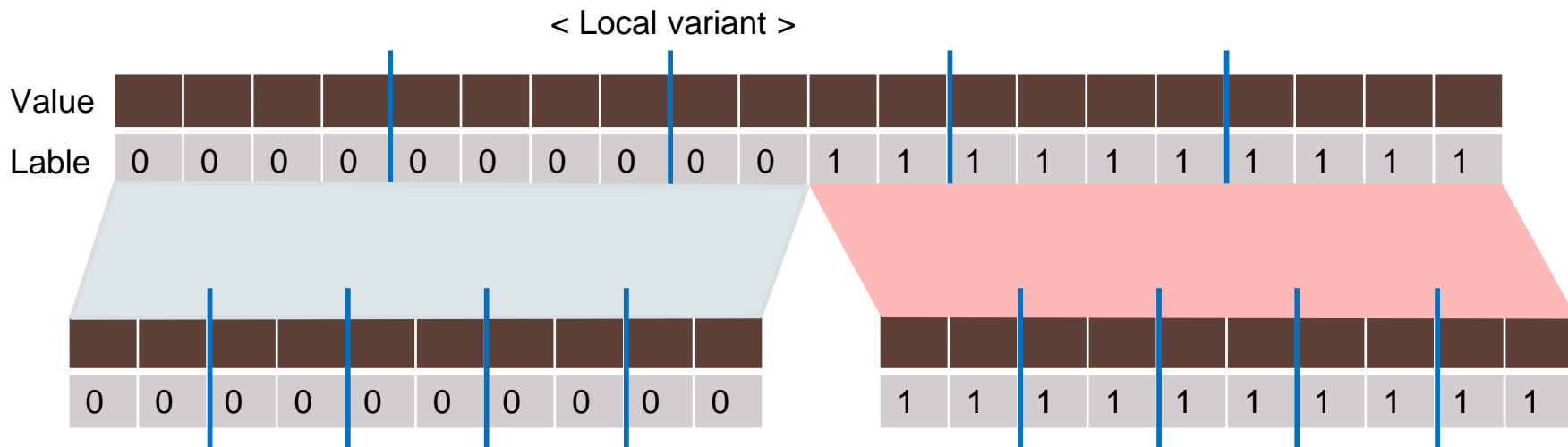
1. XG Boost의 직관적 이해

XG Boost의 아이디어 : “데이터를 분할해서 따로따로 최적의 분할 지점을 찾자”



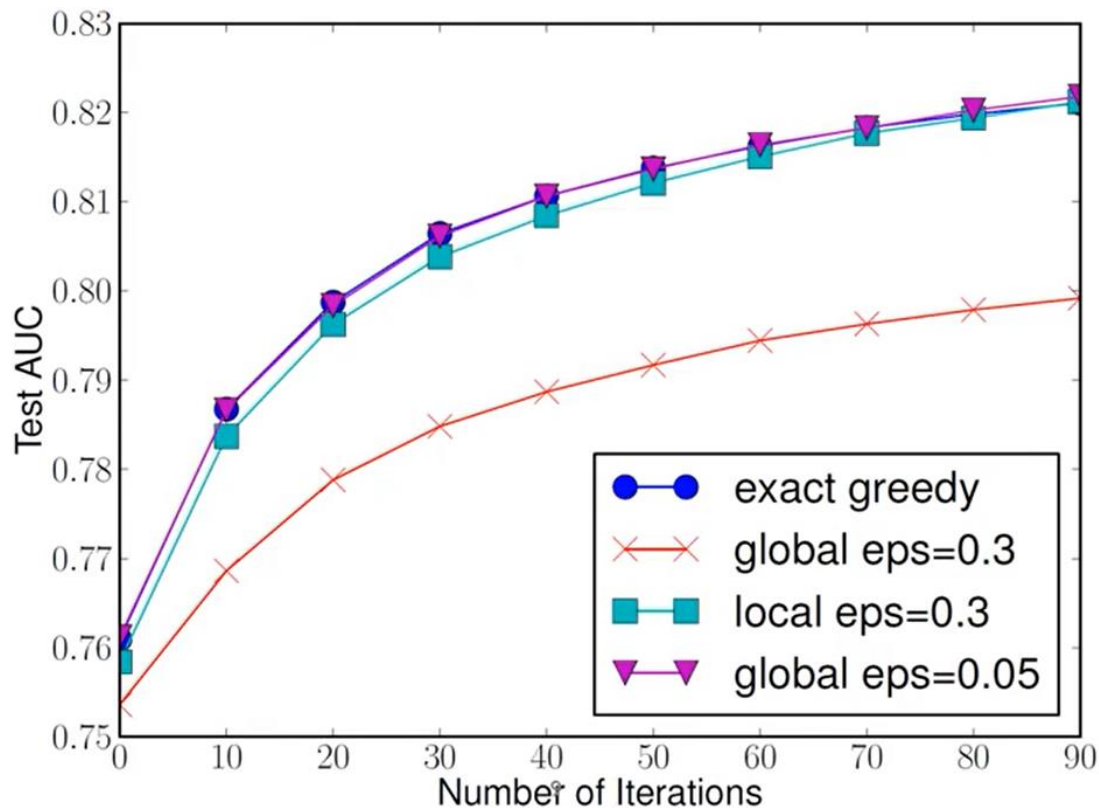
1. XG Boost의 직관적 이해

XG Boost의 아이디어 : “데이터를 분할해서 따로따로 최적의 분할 지점을 찾자”



1. XG Boost의 직관적 이해

성능 비교



Bucket의 수 : $1/\xi$

Point. Global variant를 사용할거라면 hyper parameter ξ 를 더 작게 잡아야한다.

1. XG Boost의 직관적 이해

결측치 or '0'을 효율적으로 처리

- 실제 데이터는 missing value가 많을 수 밖에 없음
- '0'으로 인코딩된 경우도 많음

1. XG Boost의 직관적 이해

결측치 or '0'을 효율적으로 처리

: default direction으로 보내버리자!

Missing value go to right

Value	1.8		1.1	0.7		2.3	0.3		1.5	3.3	0.2	0.9
Lable	1	0	1	0	0	0	0	0	1	1	0	0

Value	0.2		0.3	0.7		0.9	1.1		1.5	1.8	2.3	3.3
Lable	0	0	0	0	0	0	1	0	1	1	1	1

Value	0.2	0.3	0.7	0.9	1.1	1.5	1.8	2.3	3.3			
Lable	0	0	0	0	1	1	1	1	1	0	0	0

1. XG Boost의 직관적 이해

결측치 or '0'을 효율적으로 처리

: default direction으로 보내버리자!

Missing value go to left

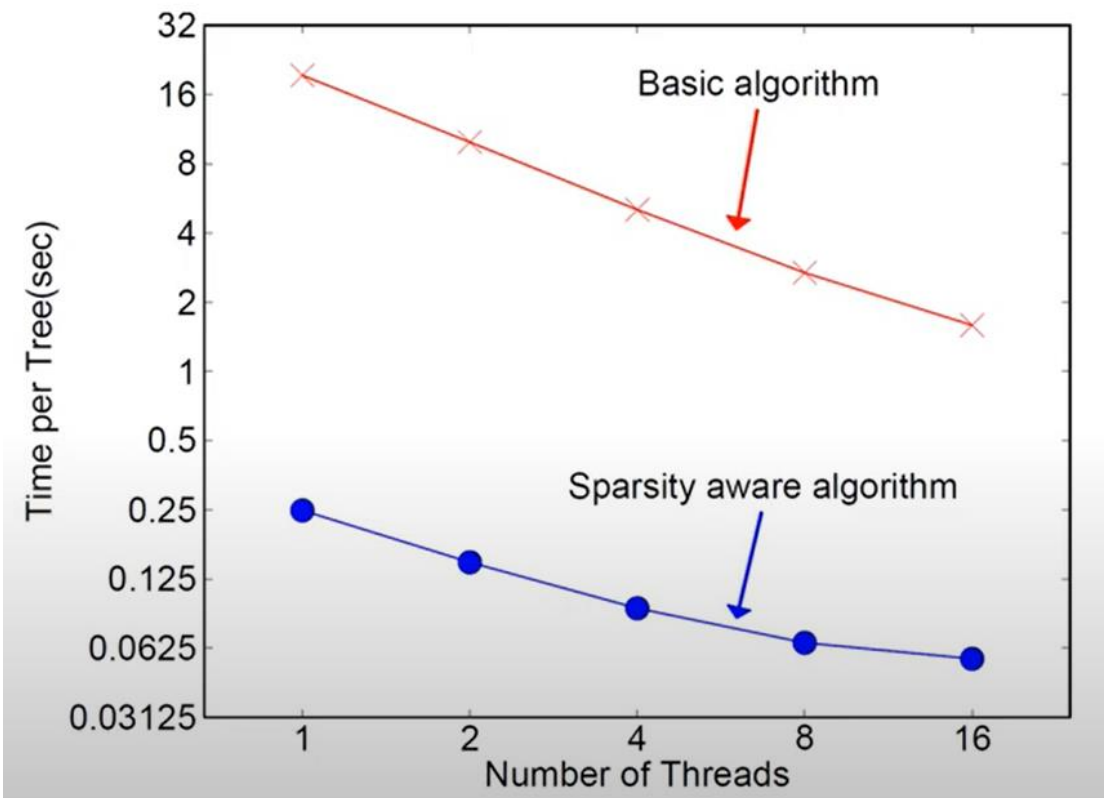
Value	1.8		1.1	0.7		2.3	0.3		1.5	3.3	0.2	0.9
Lable	1	0	1	0	0	0	0	0	1	1	0	0

Value	0.2		0.3	0.7		0.9	1.1		1.5	1.8	2.3	3.3
Lable	0	0	0	0	0	0	1	0	1	1	1	1

Value				0.2	0.3	0.7	0.9	1.1	1.5	1.8	2.3	3.3
Lable	0	0	0	0	0	0	0	1	1	1	1	1

1. XG Boost의 직관적 이해

성능 비교



1. XG Boost의 직관적 이해

자료 구조 및 하드웨어 처리

과적합 규제

Ensemble 첫번째 세션 복습

목차

1. XG Boost
2. Light Gradient Boost
3. Cat boost

2. Light GBM의 직관적 이해

Idea : 뚱뚱하고 시간이 오래걸리는 GBM을 효율적으로 사용할 순 없을까?

- Gradient based One-Side Sampling (GOSS)

“많이 틀린것만 다시 학습하자”

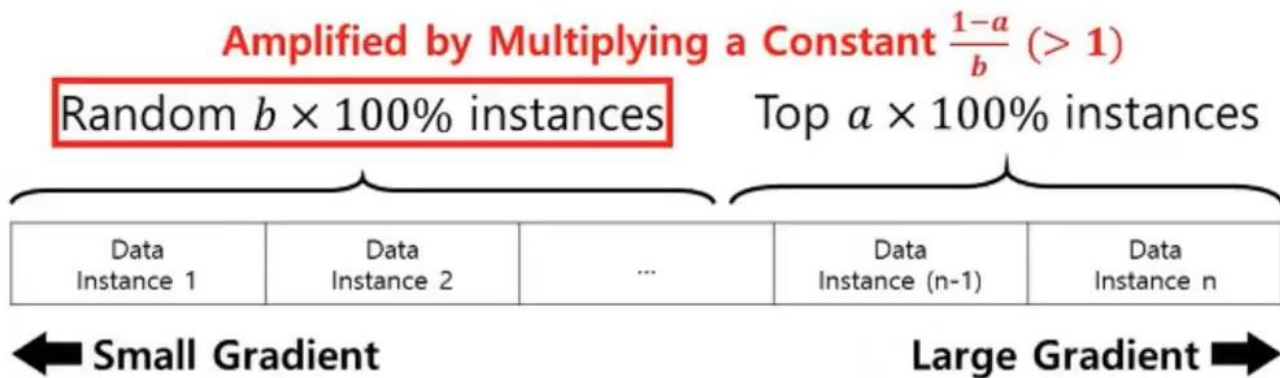
- Exclusive Feature Bundling (EFB)

“ 정보를 잃지않는 선에서 Feature끼리 묶어버리자”

2. Light GBM

- Gradient based One-Side Sampling (GOSS)

“많이 틀린것만 다시 학습하자”



2. Light GBM

- Exclusive Feature Bundling (EFB)

“ 정보를 잃지않는 선에서 Feature끼리 묶어버리자”

Feature A는 0

Feature B는 정수 k 라면 , 둘을 묶어서 하나의 feature로 만들면 정보 손실이 적지않을까?

2. Light GBM

- Exclusive Feature Bundling (EFB)

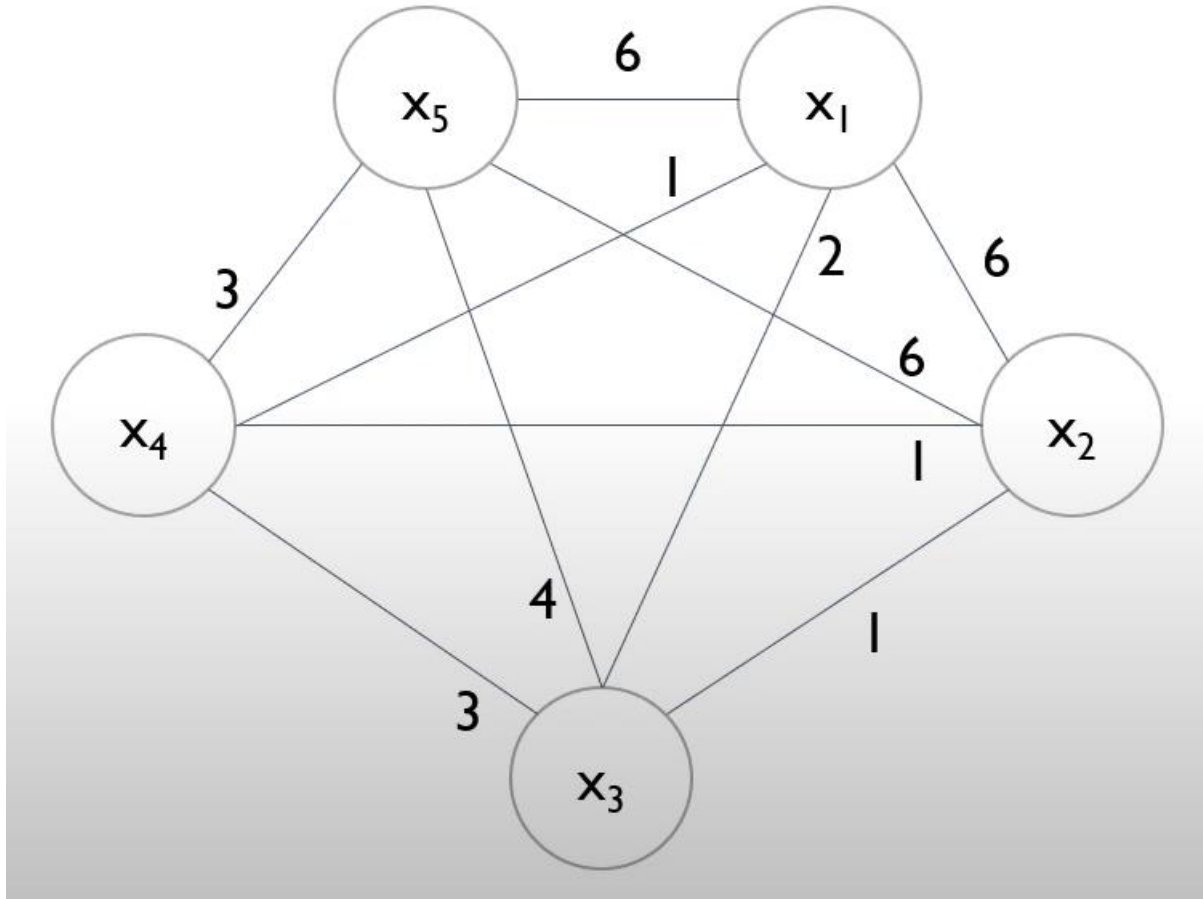
“ 정보를 잃지않는 선에서 Feature끼리 묶어버리자”

	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_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	-

	x_5	x_1	x_2	x_3	x_4
d	19	15	14	10	8

2. Light GBM




2. Light GBM

- Exclusive Feature Bundling (EFB)

“ 정보를 잃지않는 선에서 Feature끼리 묶어버리자”

	x_1	x_2	x_3	x_4	x_5
l_1	1	1	0	0	1
l_2	0	0	1	1	1
l_3	1	2	0	0	2
l_4	0	0	2	3	1
l_5	2	1	0	0	3
l_6	3	3	0	0	1
l_7	0	0	3	0	2
l_8	1	2	3	4	3
l_9	1	0	1	0	0
l_{10}	2	3	0	0	2



	x_5	x_1	x_4	x_2	x_3
l_1	1	1	0	1	0
l_2	1	0	1	0	1
l_3	2	1	0	2	0
l_4	1	0	3	0	2
l_5	3	2	0	1	0
l_6	1	3	0	3	0
l_7	2	0	0	0	3
l_8	3	1	4	2	3
l_9	0	1	0	0	1
l_{10}	2	2	0	3	0

2. Light GBM

- Exclusive Feature Bundling (EFB)

“ 정보를 잃지않는 선에서 Feature끼리 묶어버리자”

	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

	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

2. Light GBM

Idea : 뚱뚱하고 시간이 오래걸리는 GBM을 효율적으로 사용할 순 없을까?

- Gradient based One-Side Sampling (GOSS)

“많이 틀린것만 다시 학습하자”

- Exclusive Feature Bundling (EFB)

“ 정보를 잃지않는 선에서 Feature끼리 묶어버리자”