LightGBM: A Highly Efficient Gradient Boosting Decision Tree

Nov 13, 2023

Seungeun Lee

① | Introduction

② | GBDT

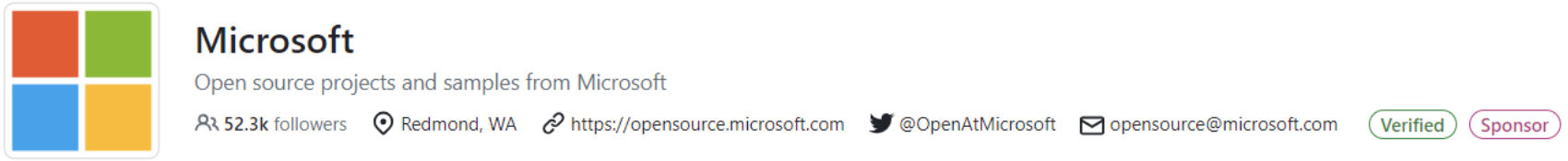
③ | GOSS

④ | EFB

⑤ | Details

Introduction

- https://lightgbm.readthedocs.io/en/stable/
- https://github.com/microsoft/LightGBM
- Microsoft



- Literally, “LIGHT” model – fast and convenient

- e.g.) for some (tremendously) large dataset(s), it may not be possible to use XGBoost (OOM errors, time-related issues, …) – then try LightGBM

Introduction

- Many engineering optimizations have been adopted in XGBoost
- The efficiency and scalability are still unsatisfactory (especially for large datasets)
- Need to scan all data instances to estimate all possible split points

- GOSS: exclude a significant proportion of data instances with small gradients
- Data instances with larger gradients play a more important role
- **Reducing # of data instances (samples)**

Introduction

- EFB: bundle mutually exclusive features (i.e., they rarely take nonzero values simultaneously)
- **Reducing # of features**

- Experiments: speeds up the training process (up to over 20 times), similar accuracy

① | Introduction

② | GBDT

③ | GOSS

④ | EFB

⑤ | Details

GBDT

- Gradient Boosting Decision Tree

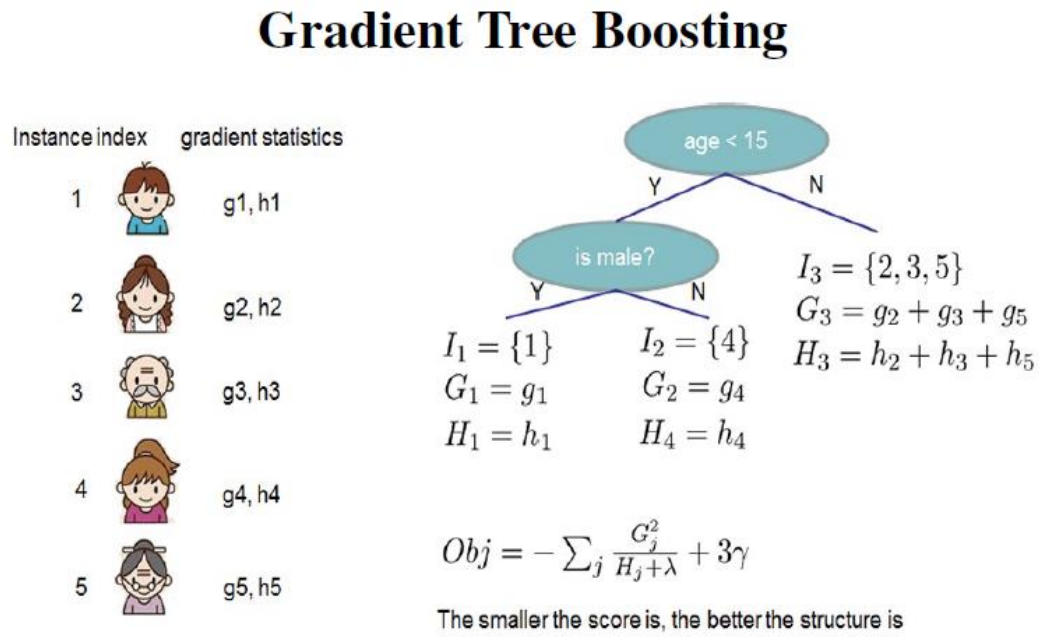


Figure 2: Structure Score Calculation. We only need to sum up the gradient and second order gradient statistics on each leaf, then apply the scoring formula to get the quality score.

① | Introduction

② | GBDT

③ | GOSS

④ | EFB

⑤ | Details

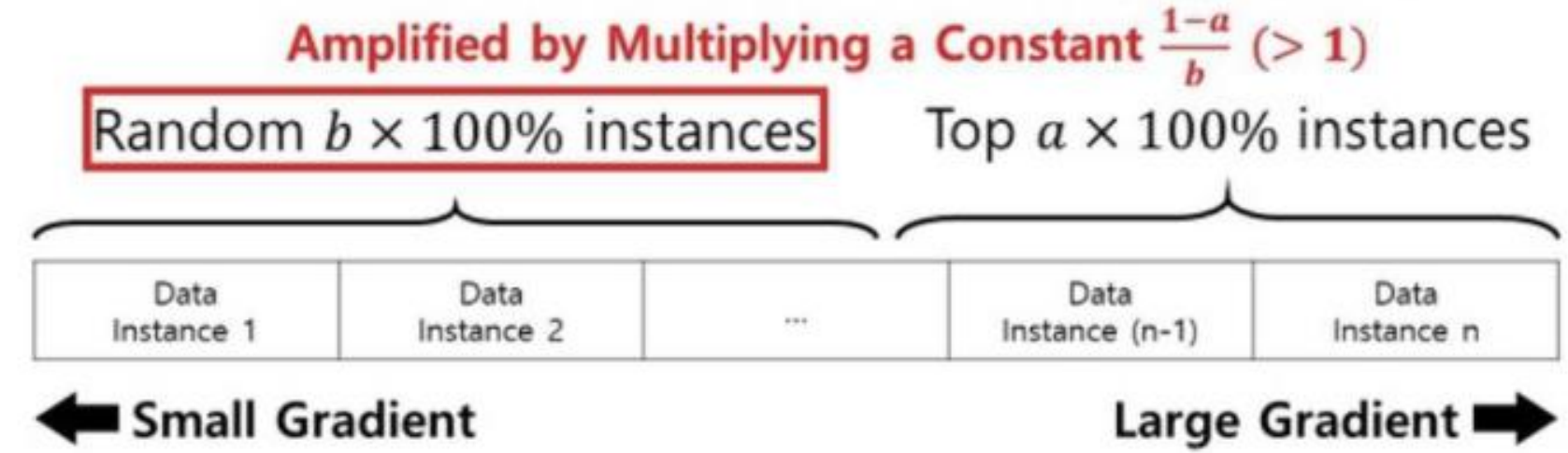
GOSS

- Gradient-based One-Sided Sampling
- Smaller gradients, smaller train loss -> trained well
- Larger gradients, larger train loss -> NOT trained well

- Data instances having larger gradients -> keep it!
- Data instances having smaller gradients -> random sampling

GOSS

- Gradient-based One-Sided Sampling



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

GOSS

- Does it really work …?! Mathematically?

Definition 3.1 *Let O be the training dataset on a fixed node of the decision tree. The variance gain of splitting feature j at point d for this node is defined as*

$$V_{j|O}(d) = \frac{1}{n_O}\left(\frac{\left(\sum_{\{x_i \in O: x_{ij} \le d\}} g_i\right)^2}{n^j_{l|O}(d)} + \frac{\left(\sum_{\{x_i \in O: x_{ij} > d\}} g_i\right)^2}{n^j_{r|O}(d)}\right),$$

where $n_O = \sum I[x_i \in O]$, $n^j_{l|O}(d) = \sum I[x_i \in O : x_{ij} \le d]$ *and* $n^j_{r|O}(d) = \sum I[x_i \in O : x_{ij} > d]$.

For feature j, the decision tree algorithm selects $d^*_j = argmax_d V_j(d)$ and calculates the largest gain $V_j(d^*_j)$. [5] Then, the data are split according feature j^* at point d_{j^*} into the left and right child nodes.

GOSS

of their gradients in the descending order; second, we keep the top-$a \times 100\%$ instances with the larger gradients and get an instance subset A; then, for the remaining set A^c consisting $(1-a) \times 100\%$ instances with smaller gradients, we further randomly sample a subset B with size $b \times |A^c|$; finally, we split the instances according to the estimated variance gain $\tilde{V}_j(d)$ over the subset $A \cup B$, i.e.,

$$\tilde{V}_j(d) = \frac{1}{n}\left(\frac{\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b}\sum_{x_i \in B_l} g_i\right)^2}{n_l^j(d)} + \frac{\left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b}\sum_{x_i \in B_r} g_i\right)^2}{n_r^j(d)}\right), \tag{1}$$

where $A_l = \{x_i \in A : x_{ij} \leq d\}$,$A_r = \{x_i \in A : x_{ij} > d\}$,$B_l = \{x_i \in B : x_{ij} \leq d\}$,$B_r = \{x_i \in B : x_{ij} > d\}$, and the coefficient $\frac{1-a}{b}$ is used to normalize the sum of the gradients over B back to the size of A^c.

GOSS

Theorem 3.2 *We denote the approximation error in GOSS as* $\mathcal{E}(d) = |\tilde{V}_j(d) - V_j(d)|$ *and* $\bar{g}_l^j(d) = \frac{\sum_{x_i \in (A \cup A^c)_l} |g_i|}{n_l^j(d)}, \bar{g}_r^j(d) = \frac{\sum_{x_i \in (A \cup A^c)_r} |g_i|}{n_r^j(d)}$. *With probability at least* $1 - \delta$, *we have*

$$\mathcal{E}(d) \leq C_{a,b}^2 \ln 1/\delta \cdot \max \left\{ \frac{1}{n_l^j(d)}, \frac{1}{n_r^j(d)} \right\} + 2DC_{a,b} \sqrt{\frac{\ln 1/\delta}{n}}, \tag{2}$$

where $C_{a,b} = \frac{1-a}{\sqrt{b}} \max_{x_i \in A^c} |g_i|$, *and* $D = \max(\bar{g}_l^j(d), \bar{g}_r^j(d))$.

GOSS

Theorem 3.2 *Let* $\bar{g}_l^j(d) = \frac{\sum_{x_i \in (A \cup A^c)_l} |g_i|}{n_l^j(d)}$ *and* $\bar{g}_r^j(d) = \frac{\sum_{x_i \in (A \cup A^c)_r} |g_i|}{n_r^j(d)}$. *With probability at least* $1 - \delta$, *we have*

$$\mathcal{E}_A(d) \leq C_{a,b}^2 \ln 1/\delta \cdot \max \left\{ \frac{1}{n_l^j(d)}, \frac{1}{n_r^j(d)} \right\} + 2DC_{a,b} \sqrt{\frac{\ln 1/\delta}{n}}, \tag{1}$$

where $C_{a,b} = \frac{1-a}{\sqrt{b}} \max_{x_i \in A^c} |g_i|$, *and* $D = \max(\bar{g}_l^j(d), \bar{g}_r^j(d))$.

Proof: For a fixed d, we have

$$\begin{aligned}
&\tilde{V}_j(d) - V_j(d) \\
&= \left(\frac{(\sum_{x_i \in A_l} g_i + \frac{1-a}{b} \sum_{x_i \in B_l} g_i)^2}{n_l^j(d)} + \frac{(\sum_{x_i \in A_r} g_i + \frac{1-a}{b} \sum_{x_i \in B_r} g_i)^2}{n_r^j(d)} \right) \\
&\quad - \left(\frac{(\sum_{x_i \in A_l} g_i + \sum_{x_i \in A_l^c} g_i)^2}{n_l^j(d)} + \frac{(\sum_{x_i \in A_r} g_i + \sum_{x_i \in A_r^c} g_i)^2}{n_r^j(d)} \right)
\end{aligned}$$

GOSS

$$=C_l\left(\frac{1-a}{b}\sum_{x_i\in B_l} g_i-\sum_{x_i\in A_l^c} g_i\right)+C_r\left(\frac{1-a}{b}\sum_{x_i\in B_r} g_i-\sum_{x_i\in A_r^c} g_i\right)$$

where $C_l=\frac{\left(\frac{1-a}{b}\sum_{x_i\in B_l} g_i+\sum_{x_i\in A_l^c} g_i+2(\sum_{x_i\in A_l} g_i)\right)}{n_l^j(d)}$,

and $C_r=\frac{\left(\frac{1-a}{b}\sum_{x_i\in B_r} g_i+\sum_{x_i\in A_r^c} g_i+2(\sum_{x_i\in A_r} g_i)\right)}{n_r^j(d)}$.

Thus, we have

$$|\tilde{V}_j(d)-V_j(d)|\leq \max\{C_l,C_r\}\left|\frac{1-a}{b}\sum_{x_i\in B} g_i-\sum_{x_i\in A_c} g_i\right| \tag{2}$$

GOSS

Firstly, we bound C_l and C_r. Let $D_{A^c} = \max_{x_i \in A^c} |g_i|$, we have

$$C_l = \frac{\left(\frac{1-a}{b}\sum_{x_i \in B_l} g_i + \sum_{x_i \in A_l} g_i\right)}{n_l^j(d)} + \frac{\left(\sum_{x_i \in A_l^c} g_i + \sum_{x_i \in A_l} g_i\right)}{n_l^j(d)} \tag{3}$$

$$\leq \frac{D_{A^c}\left|\frac{1-a}{b}\sum I_{[x_i \in B_l]} - \sum I_{[x_i \in A_l^c]}\right|}{n_l^j(d)} + 2D \tag{4}$$

$$= \frac{D_{A^c}(1-a)n}{n_l^j(d)}\left|\frac{\sum I_{[x_i \in B_l]}}{bn} - \frac{\sum I_{[x_i \in A_l^c]}}{(1-a)n}\right| + 2D \tag{5}$$

I is not explicitly defined in the paper ;)
I stands for data Instance, I guess

By Hoeffding's inequality, we have with probability at least $1-\delta$,

$$C_l \leq \frac{D_{A^c}(1-a)n}{n_l^j(d)}\sqrt{\frac{\ln 2/\delta}{2bn}} + 2D. \tag{6}$$

Hoeffding's inequality provides an upper bound on the probability that the sum of bounded independent random variables deviates from its expected value by more than a certain amount

GOSS

Let $X_1, \ldots, X_n$ be independent random variables such that $a_i \leq X_i \leq b_i$ almost surely. Consider the sum of these random variables,

$$S_n = X_1 + \cdots + X_n.$$

Then Hoeffding's theorem states that, for all $t > 0$,[3]

$$\mathrm{P}(S_n - \mathrm{E}[S_n] \geq t) \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right)$$

$$\mathrm{P}(|S_n - \mathrm{E}[S_n]| \geq t) \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right)$$

Here $\mathrm{E}[S_n]$ is the expected value of S_n.

GOSS

Similarly, we have $C_r \leq \frac{D_{A^c}(1-a)n}{n_r^j(d)}\sqrt{\frac{\ln 2/\delta}{2bn}} + 2D.$

For the term $\left(\frac{1-a}{b}\sum_{x_i \in B} g_i - \sum_{x_i \in A^c} g_i\right)$, we have with probability at least $1-\delta$,

$$\frac{1}{n}\left|\frac{1-a}{b}\sum_{x_i \in B} g_i - \sum_{x_i \in A^c} g_i\right| \leq D_{A^c}(1-a)\sqrt{\frac{\ln 2/\delta}{2bn}}. \tag{7}$$

Thus, we have with probability at least $1-\delta$

$$\begin{aligned}
\mathcal{E}(d) &= \left|\frac{\tilde{V}_j(d)}{n} - \frac{V_j(d)}{n}\right| \qquad \textbf{(1 - a)} \\
&\leq \left(D_{A^c}(1-a)\max\left\{\frac{1}{n_l^j(d)}, \frac{1}{n_r^j(d)}\right\}\sqrt{\frac{n\ln 1/\delta}{2b}} + 2D\right) a)\sqrt{\frac{\ln 1/\delta}{2bn}} \\
&\leq \frac{D_{A^c}^2(1-a)^2\ln 1/\delta}{2b}\cdot\max\left\{\frac{1}{n_l^j(d)}, \frac{1}{n_r^j(d)}\right\} + \frac{2D\cdot D_{A^c}\cdot(1-a)}{\sqrt{2b}}\sqrt{\frac{\ln 1/\delta}{n}}.
\end{aligned}$$

Putting $C_{a,b}$ in the above inequality, we can get the result in the theorem. □

Hoeffding's Inequality (as a math ppl)

The use of Rademacher averages in classification was first promoted by Koltchinskii [75] and Bartlett, Boucheron, and Lugosi [76], see also Koltchinskii and Panchenko [77, 78], Bartlett and Mendelson [79], Bartlett, Bousquet, and Mendelson [80], Bousquet, Koltchinskii, and Panchenko [81], Kégl, Linder, and Lugosi [82].

A Probability Tools

This section recalls some basic facts from probability theory that are used throughout this tutorial (sometimes without explicitly mentioning it).

We denote by A and B some events (i.e. elements of a σ-algebra), and by X some real-valued random variable.

A.1 Basic Facts

- Union:
$$\mathbb{P}[A \text{ or } B] \leq \mathbb{P}[A] + \mathbb{P}[B] .$$
- Inclusion: If $A \Rightarrow B$, then $\mathbb{P}[A] \leq \mathbb{P}[B]$.
- Inversion: If $\mathbb{P}[X > t] \leq F(t)$ then with probability at least $1 - \delta$,
$$X \leq F^{-1}(\delta) .$$
- Expectation: If $X \geq 0$,
$$\mathbb{E}[X] = \int_0^\infty \mathbb{P}[X \geq t]\, dt .$$

A.2 Basic Inequalities

All the inequalities below are valid as soon as the right-hand side exists.

- Jensen: for f convex,
$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)] .$$
- Markov: If $X \geq 0$ then for all $t > 0$,
$$\mathbb{P}[X \geq t] \leq \frac{\mathbb{E}[X]}{t} .$$
- Chebyshev: for $t > 0$,
$$\mathbb{P}[|X - \mathbb{E}[X]| \geq t] \leq \frac{\mathrm{Var}X}{t^2} .$$
- Chernoff: for all $t \in \mathbb{R}$,
$$\mathbb{P}[X \geq t] \leq \inf_{\lambda \geq 0} \mathbb{E}\left[e^{\lambda(X-t)}\right] .$$

Applying this to $f(Z) = \mathbb{1}_{g(X) \neq Y}$ we get that for any g, and any $\delta > 0$, with probability at least $1 - \delta$

$$R(g) \leq R_n(g) + \sqrt{\frac{\log \frac{2}{\delta}}{2n}} . \tag{3}$$

Notice that one has to consider a fixed function g and the probability is with respect to the sampling of the data. If the function depends on the data this does not apply!

3.3 Limitations

Although the above result seems very nice (since it applies to any class of bounded functions), it is actually severely limited. Indeed, what it essentially says is that for each (fixed) function $f \in \mathcal{F}$, there is a set S of samples for which $Pf - P_n f \leq \sqrt{\frac{\log \frac{2}{\delta}}{2n}}$ (and this set of samples has measure $\mathbb{P}[S] \geq 1 - \delta$). However, these sets S may be different for different functions. In other words, for the observed sample, only some of the functions in $\mathcal{F}$ will satisfy this inequality.

Another way to explain the limitation of Hoeffding's inequality is the following. If we take for $\mathcal{G}$ the class of all $\{-1, 1\}$-valued (measurable) functions, then for any fixed sample, there exists a function $f \in \mathcal{F}$ such that

$$Pf - P_n f = 1 .$$

To see this, take the function which is $f(X_i) = Y_i$ on the data and $f(X) = -Y$ everywhere else. This does not contradict Hoeffding's inequality but shows that it does not yield what we need.

Figure 2 illustrates the above argumentation. The horizontal axis corresponds

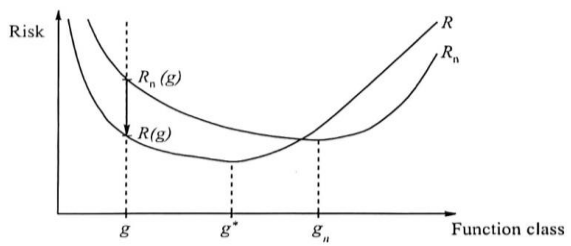


Fig. 2. Convergence of the empirical risk to the true risk over the class of functions.

- *Error bound*: $R(g_n) \leq R_n(g_n) + B(n, \mathcal{G})$. This corresponds to the estimation of the risk from an empirical quantity.
- *Error bound relative to the best in the class*: $R(g_n) \leq R(g^*) + B(n, \mathcal{G})$. This tells how "optimal" is the algorithm given the model it uses.
- *Error bound relative to the Bayes risk*: $R(g_n) \leq R^* + B(n, \mathcal{G})$. This gives theoretical guarantees on the convergence to the Bayes risk.

3 Basic Bounds

In this section we show how to obtain simple error bounds (also called generalization bounds). The elementary material from probability theory that is needed here and in the later sections is summarized in Appendix A.

3.1 Relationship to Empirical Processes

Recall that we want to estimate the risk $R(g_n) = \mathbb{E}\left[\mathbb{1}_{g_n(X) \neq Y}\right]$ of the function g_n returned by the algorithm after seeing the data $(X_1, Y_1), \ldots, (X_n, Y_n)$. This quantity cannot be observed (P is unknown) and is a random variable (since it depends on the data). Hence one way to make a statement about this quantity is to say how it relates to an estimate such as the empirical risk $R_n(g_n)$. This relationship can take the form of upper and lower bounds for

$$\mathbb{P}\left[R(g_n) - R_n(g_n) > \varepsilon\right].$$

For convenience, let $Z_i = (X_i, Y_i)$ and $Z = (X, Y)$. Given $\mathcal{G}$ define the *loss class*

$$\mathcal{F} = \{f : (x, y) \mapsto \mathbb{1}_{g(x) \neq y} : g \in \mathcal{G}\}. \tag{1}$$

Notice that $\mathcal{G}$ contains functions with range in $\{-1, 1\}$ while $\mathcal{F}$ contains non-negative functions with range in $\{0, 1\}$. In the remainder of the tutorial, we will go back and forth between $\mathcal{F}$ and $\mathcal{G}$ (as there is a bijection between them), sometimes stating the results in terms of functions in $\mathcal{F}$ and sometimes in terms of functions in $\mathcal{G}$. It will be clear from the context which classes $\mathcal{G}$ and $\mathcal{F}$ we refer to, and $\mathcal{F}$ will always be derived from the last mentioned class $\mathcal{G}$ in the way of (1).

We use the shorthand notation $Pf = \mathbb{E}\left[f(X, Y)\right]$ and $P_n f = \frac{1}{n}\sum_{i=1}^n f(X_i, Y_i)$. P_n is usually called the *empirical measure* associated to the training sample. With this notation, the quantity of interest (difference between true and empirical risks) can be written as

$$Pf_n - P_n f_n. \tag{2}$$

An empirical process is a collection of random variables indexed by a class of functions, and such that each random variable is distributed as a sum of i.i.d. random variables (values taken by the function at the data):

$$\{Pf - P_n f\}_{f \in \mathcal{F}}.$$

One of the most studied quantity associated to empirical processes is their supremum:

$$\sup_{f \in \mathcal{F}} Pf - P_n f.$$

It is clear that if we know an upper bound on this quantity, it will be an upper bound on (2). This shows that the theory of empirical processes is a great source of tools and techniques for Statistical Learning Theory.

3.2 Hoeffding's Inequality

Let us rewrite again the quantity we are interested in as follows

$$R(g) - R_n(g) = \mathbb{E}\left[f(Z)\right] - \frac{1}{n}\sum_{i=1}^n f(Z_i).$$

It is easy to recognize here the difference between the expectation and the empirical average of the random variable $f(Z)$. By the law of large numbers, we immediately obtain that

$$\mathbb{P}\left[\lim_{n \to \infty} \frac{1}{n}\sum_{i=1}^n f(Z_i) - \mathbb{E}\left[f(Z)\right] = 0\right] = 1.$$

This indicates that with enough samples, the empirical risk of a function is a good approximation to its true risk.
It turns out that there exists a quantitative version of the law of large numbers when the variables are bounded.

Theorem 1 (Hoeffding). *Let $Z_1, \ldots, Z_n$ be n i.i.d. random variables with $f(Z) \in [a, b]$. Then for all $\varepsilon > 0$, we have*

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{i=1}^n f(Z_i) - \mathbb{E}\left[f(Z)\right]\right| > \varepsilon\right] \leq 2\exp\left(-\frac{2n\varepsilon^2}{(b-a)^2}\right).$$

Let us rewrite the above formula to better understand its consequences. Denote the right hand side by δ. Then

$$\mathbb{P}\left[|P_n f - Pf| > (b-a)\sqrt{\frac{\log\frac{2}{\delta}}{2n}}\right] \leq \delta,$$

or (by inversion, see Appendix A) with probability at least $1 - \delta$,

$$|P_n f - Pf| \leq (b-a)\sqrt{\frac{\log\frac{2}{\delta}}{2n}}.$$

① | Introduction

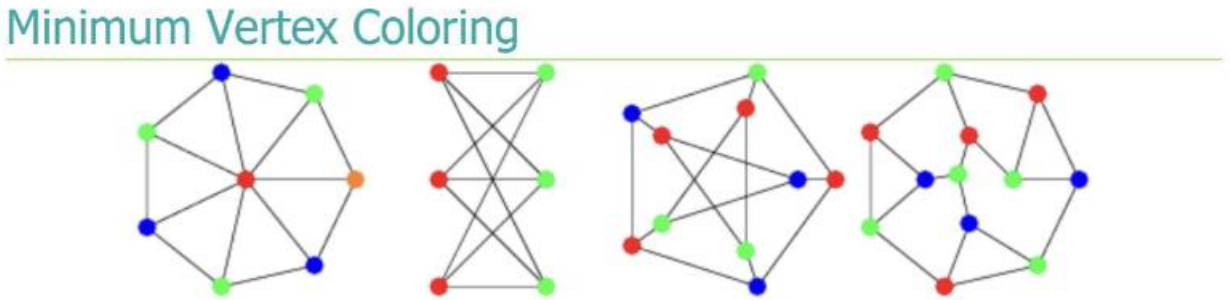
② | GBDT

③ | GOSS

④ | EFB

⑤ | Details

EFB



- Exclusive Feature Bundling
- Graph (V, E)
- V: feature
- E: total conflicts between features

- Exclusive Feature Bundling -> set of vertexes with same colors
- Minimum Vertex Coloring (Graph Coloring, NP-Hard)

-> approximation: Greedy Bundling

EFB

- Sparse feature space
- Many features are (almost) exclusive
- i.e. they rarely take nonzero values simultaneously

- If two nonzero or zero values appear simultaneously, we (are going to) deem them a “CONFLICT”

EFB

	x1	x2	x3	x4	x5
I1	1	1	0	0	1
I2	0	0	1	1	1
I3	1	2	0	0	2
I4	0	0	2	3	1
I5	2	1	0	0	3
I6	3	3	0	0	1
I7	0	0	3	0	2
I8	1	2	3	4	3
I9	1	0	1	0	0
I10	2	3	0	0	2

	x1	x2	x3	x4	x5
x1	-	6	2	1	6
x2	6	-	1	1	6
x3	2	1	-	3	4
x4	1	1	3	-	3
x5	6	6	4	3	-
	15	14	10	8	19

	x1	x2	x3	x4	x5
d	19	15	14	10	8

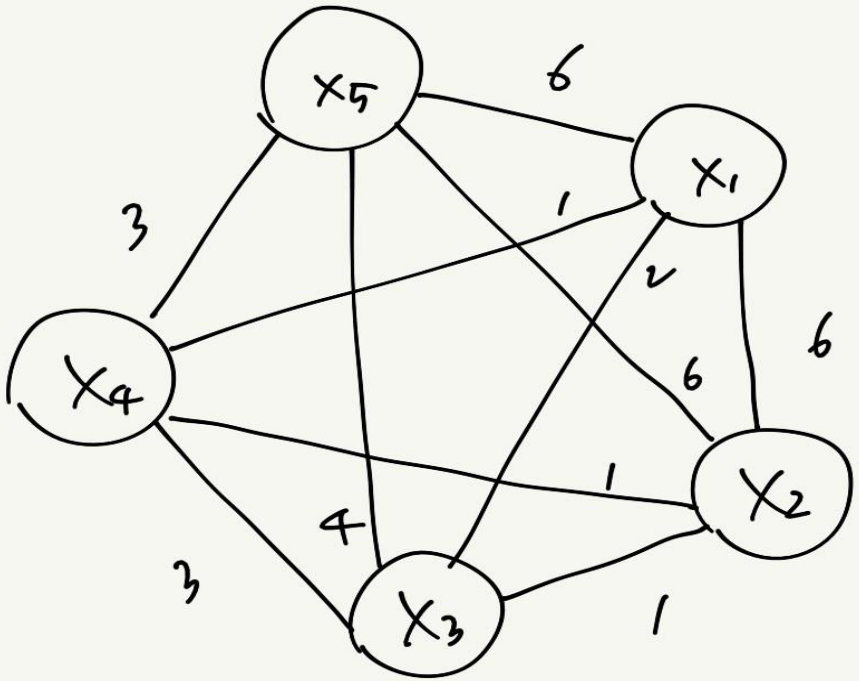
degree: # of edges that are connected
start with the node having the largest degree

EFB

hyperparameter

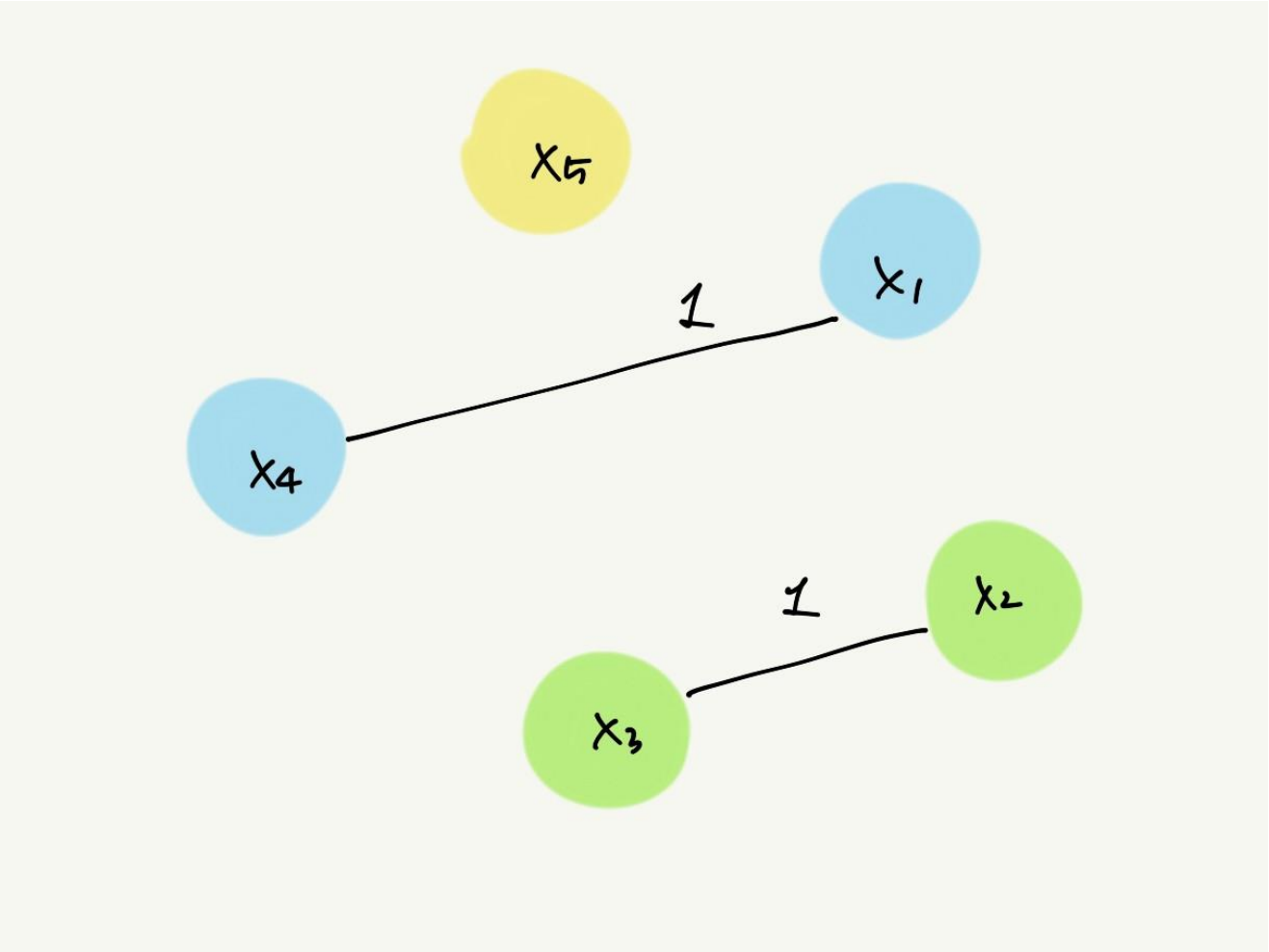
cut-off=0.2

N=10 (#I)



N=10, 10 x 0.2 = 2 -> cut if 이상 than 2

EFB



Exclusive Feature Bundling!

EFB

- Add offsets to the original values of the features

(nonzero, zero)

zero

	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 max 3 / min 0 max 4

	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

Conflict: Use the value of x_1

Conflict: Use the value of x_2

adding referencing variables to the referenced variables
referencing variables: 1 / 2 (smaller one)

① | Introduction

② | GBDT

③ | GOSS

④ | EFB

⑤ | Details

Details

Table 2: Overall training time cost comparison. LightGBM is lgb_baseline with GOSS and EFB. EFB_only is lgb_baseline with EFB. The values in the table are the average time cost (seconds) for training one iteration.

	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**

Table 3: Overall accuracy comparison on test datasets. Use AUC for classification task and NDCG@10 for ranking task. SGB is lgb_baseline with Stochastic Gradient Boosting, and its sampling ratio is the same as LightGBM.

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

Details

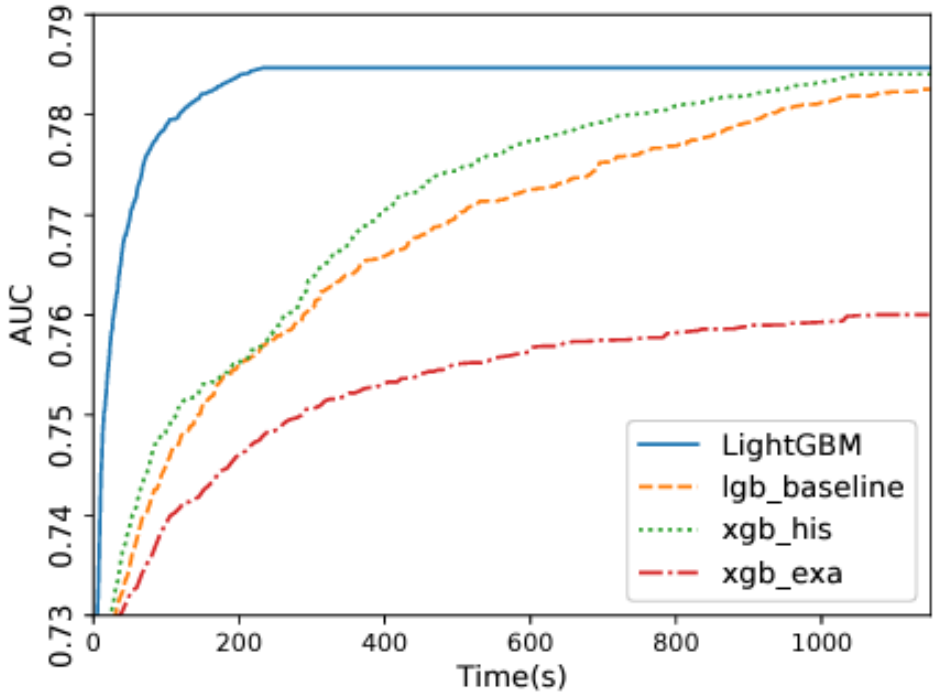


Figure 1: Time-AUC curve on Flight Delay.

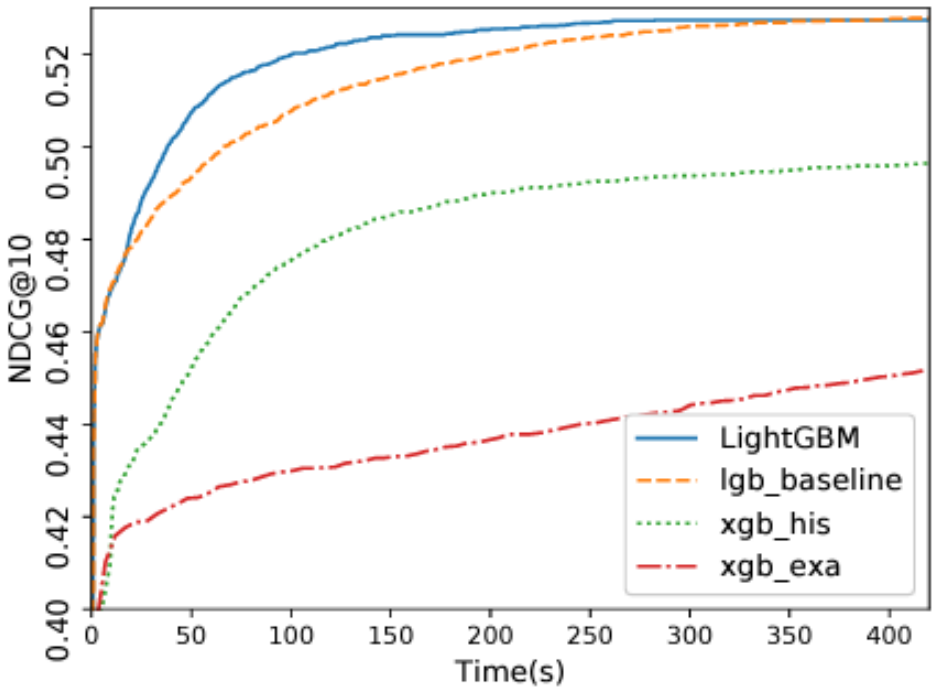


Figure 2: Time-NDCG curve on LETOR.

Details

- LightGBM can handle MISSING VALUES
- Missing values

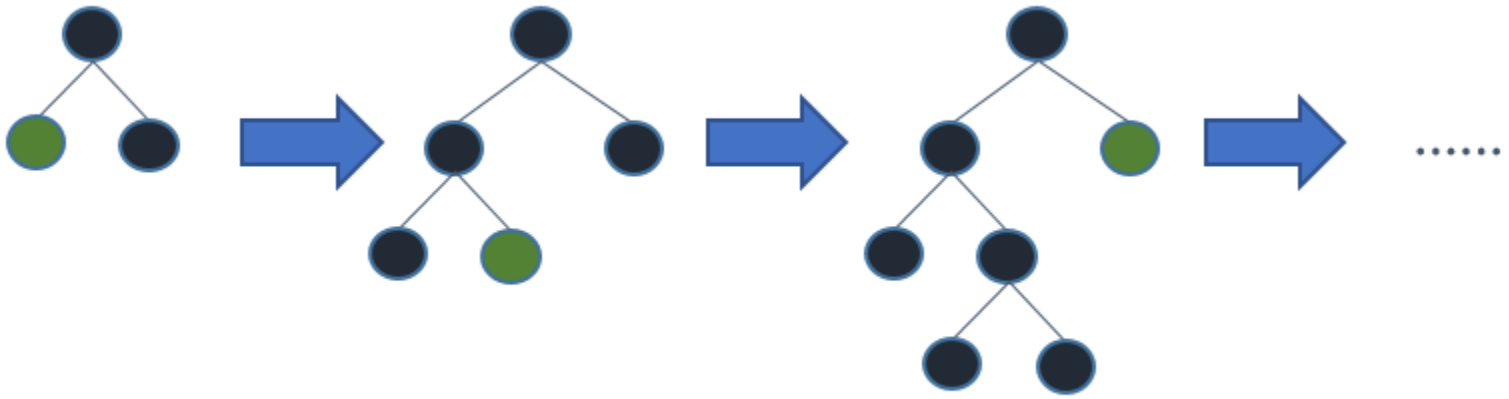
(1) Does not exist

(2) Not applicable (unknown)

- Handling missing value as it is

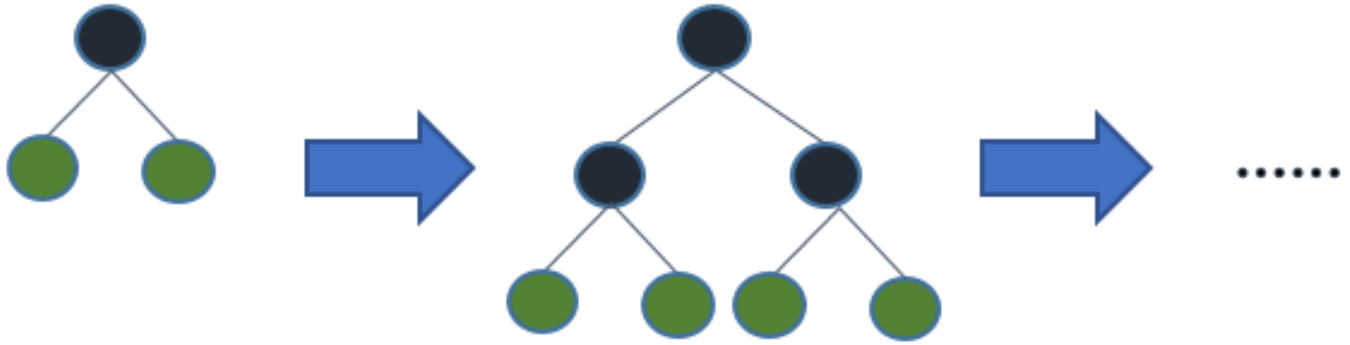
- Code implementation
- https://drive.google.com/drive/u/5/folders/1OPBBz-FA4IRkEEodf9kmfwJBz7fzazPu

Details



Leaf-wise tree growth

- Concentrates on parts that have larger loss than the other side



Level-wise tree growth

- balancing

Reference

- https://bluemumin.github.io/review/2021/04/11/Review-onepicturelgbm/
- LightGBM: A Highly Efficient Gradient Boosting Decision Tree (& supplementary materials attached)
- https://github.com/pilsung-kang/Business-Analytics-IME654-/

What’s left?

- Tabnet
- Multi-modal analysis
- Meta Learning
- Deep Tabular Learning - Meta Learning (STUNT)
- (SAC)