A Summary of Appendix

We provide more details of theoretical analysis (Appendix B and D), experiment results and settings (Appendix C) and a brief introduction to our new implementation of LightGBM CUDA version (Appendix E).

B Theoretical Analysis

For the simplicity of notations, we will use s in place of I_s to represent the indices of data instances in leaf s in this section.

B.1 Existence of $\gamma_s > 0$

Theorem B.1.1 With constant hessian value h, if leaf s has a split gain $\Delta\mathcal{L}_{s\to s_1,s_2} > 0$, then with weights $|g_i|$ and labels $\text{sign}(g_i)$, there exists $\hat{\gamma}_s > 0$ such that the split $s \to s_1, s_2$ has a weighted classification error rate $\frac{1}{2} - \hat{\gamma}_s < \frac{1}{2}$ for $\mathcal{D}_s$.

Proof of Theorem B.1.1 Since

$$\Delta\mathcal{L}_{s\to s_1,s_2} = \frac{G_{s_1}^2}{2H_{s_1}} + \frac{G_{s_2}^2}{2H_{s_2}} - \frac{G_s^2}{2H_s} = \frac{G_{s_1}^2}{2hn_{s_1}} + \frac{G_{s_2}^2}{2hn_{s_2}} - \frac{G_s^2}{2hn_s} > 0, \tag{10}$$

we have $|G_{s_1}| > 0$ or $|G_{s_2}| > 0$. W.L.O.G., suppose $|G_{s_1}| > 0$. Define s_1^+ as the set of indices such that $\forall i \in s_1^+, \text{sign}(g_i)$ equals the weighted majority of $\text{sign}(g_i)$ for all $i \in s_1$ (weighted by $|g_i|$), and $s_1^- = s_1 - s_1^+$. Then

$$|G_{s_1}| = \left|\sum_{i\in s_1} g_i\right| = \sum_{i\in s_1^+} |g_i| - \sum_{i\in s_1^-} |g_i| > 0. \tag{11}$$

Similarly we define s_2^+ and s_2^-. Then by definition

$$|G_{s_2}| = \left|\sum_{i\in s_1} g_i\right| = \sum_{i\in s_2^+} |g_i| - \sum_{i\in s_2^-} |g_i| \geq 0. \tag{12}$$

Thus the weighted error rate

$$\begin{aligned}\frac{\sum_{i\in s_1^-}|g_i| + \sum_{i\in s_2^-}|g_i|}{\sum_{i\in s_1}|g_i| + \sum_{i\in s_2}|g_i|} &= \frac{1}{2} - \frac{\sum_{i\in s_1^+}|g_i| + \sum_{i\in s_2^+}|g_i| - \sum_{i\in s_1^-}|g_i| - \sum_{i\in s_2^-}|g_i|}{2\left(\sum_{i\in s_1}|g_i| + \sum_{i\in s_2}|g_i|\right)} \\ &= \frac{1}{2} - \frac{|G_{s_1}| + |G_{s_2}|}{2\left(\sum_{i\in s_1}|g_i| + \sum_{i\in s_2}|g_i|\right)} < \frac{1}{2}.\end{aligned} \tag{13}$$

Setting

$$\hat{\gamma}_s = \frac{|G_{s_1}| + |G_{s_2}|}{2\left(\sum_{i\in s_1}|g_i| + \sum_{i\in s_2}|g_i|\right)} > 0 \tag{14}$$

completes the proof.

B.2 Proof of Theorem 5.3

Definition 5.1 (Weak Learnability of Stumps) Given a binary classification dataset $\mathcal{D} = \{(\mathbf{x}_i, c(\mathbf{x}_i))\}_{i=1}^N$ where $c(\mathbf{x}_i) \in \{-1, 1\}$, weights $\{w_i\}_{i=1}^N$, $w_i \geq 0$ and $\sum_i w_i > 0$, there exists $\gamma > 0$ and a two-leaf decision tree with leaf values in $\{-1, 1\}$ s.t. the weighted classification error rate on $\mathcal{D}$ is $\frac{1}{2} - \gamma$. Then the dataset $\mathcal{D}$ is γ-empirically weakly learnable by stumps w.r.t. c and $\{w_i\}_{i=1}^N$.

Assumption 5.2 Let $\text{sign}(\cdot)$ be the sign function (with $\text{sign}(0) = 1$). For data subset $\mathcal{D}_s \subset \mathcal{D}$ in leaf s, there exists a stump and a $\gamma_s > 0$ s.t. $\mathcal{D}_s$ is γ_s-empirically weakly learnable by stumps, w.r.t. concept $c(\mathbf{x}_i) = \text{sign}(g_i)$ and weights $w_i = |g_i|$, where $i \in I_s$.

Theorem 5.3 For loss functions with constant hessian value $h > 0$, if Assumption 5.2 holds for the subset $\mathcal{D}_s$ in leaf s for some $\gamma_s > 0$, then with stochastic rounding and leaf-value refitting, for any $\epsilon > 0$, and $\delta > 0$, at least one of the following conclusions holds:

1. With any split of leaf s and its descendants, the resultant average of absolute values of prediction values by the tree in current boosting iteration for data in $\mathcal{D}_s$ is no greater than ϵ/h.

2. For any split $s \to s_1, s_2$ of leaf s, with a probability of at least $1-\delta$,

$$\frac{\left|\widetilde{\mathcal{G}}_{s\to s_1,s_2} - \mathcal{G}_{s\to s_1,s_2}\right|}{\mathcal{G}_s^*} \leq \frac{\max\limits_{i\in[N]}|g_i|\sqrt{2\ln\frac{4}{\delta}}}{\gamma_s^2\epsilon\cdot 2^{B-1}}\left(\sqrt{\frac{1}{n_{s_1}}}+\sqrt{\frac{1}{n_{s_2}}}\right) + \frac{\left(\max\limits_{i\in[N]}|g_i|\right)^2\ln\frac{4}{\delta}}{\gamma_s^2\epsilon^2 n_s\cdot 4^{B-2}}. \tag{15}$$

Proof of Theorem 5.3 By leaf-wise weak learnability (Assumption 5.2), there exists a split $s \to s_L, s_R$ and $\gamma_s > 0$ for s s.t. for data in $\mathcal{D}_s$, with binary labels $c(\mathbf{x}_i) = \text{sign}(g_i)$ and weights $w_i = |g_i|$, the split results in a stump with weighted binary-classification error rate is $\frac{1}{2} - \gamma_s$. Suppose that in s_L, s_L^+ is the set of weighted majority samples, and s_L^- is the set of weighted minority samples (thus $\text{sign}(g_i) = +1, \forall i \in s_L^+$ and $\text{sign}(g_i) = -1, \forall i \in s_L^-$, or $\text{sign}(g_i) = -1, \forall i \in s_L^+$ and $\text{sign}(g_i) = +1, \forall i \in s_L^-$) such that $\sum_{i\in s_L^+}|g_i| \geq \sum_{i\in s_L^-}|g_i|$. Similarly, we define s_R^+ and s_R^-. Then we have the weighted error rate

$$\overline{\text{err}} = \frac{\sum_{i\in s_L^-}|g_i| + \sum_{i\in s_R^-}|g_i|}{\sum_{i\in s}|g_i|} = \frac{1}{2} - \gamma_s. \tag{16}$$

Thus

$$\frac{\sum_{i\in s_L^+}|g_i| + \sum_{i\in s_R^+}|g_i|}{\sum_{i\in s}|g_i|} = 1 - \overline{\text{err}} = \frac{1}{2} + \gamma_s. \tag{17}$$

Since $\mathcal{G}_s^*$ is for the optimal split in leaf s, we have

$$\begin{aligned}
\mathcal{G}_s^* \geq \mathcal{G}_{s\to s_L,s_R} &= \frac{\left(\sum_{i\in s_L} g_i\right)^2}{2hn_{s_L}} + \frac{\left(\sum_{i\in s_R} g_i\right)^2}{2hn_{s_R}} \\
&\geq \frac{\left(\left|\sum_{i\in s_L} g_i\right| + \left|\sum_{i\in s_R} g_i\right|\right)^2}{2h\left(n_{s_L}+n_{s_R}\right)} \\
&= \frac{\left(\sum_{i\in s_L^+}|g_i| - \sum_{i\in s_L^-}|g_i| + \sum_{i\in s_R^+}|g_i| - \sum_{i\in s_R^-}|g_i|\right)^2}{2h\left(n_{s_L}+n_{s_R}\right)} \\
&= \frac{\left(\sum_{i\in s_L^+\cup s_R^+}|g_i| - \sum_{i\in s_L^-\cup s_R^-}|g_i|\right)^2}{2h\left(n_{s_L}+n_{s_R}\right)} \\
&= \frac{\left(\left(\frac{1}{2}+\gamma_s\right)\sum_{i\in s}|g_i| - \left(\frac{1}{2}-\gamma_s\right)\sum_{i\in s}|g_i|\right)^2}{2h\left(n_{s_L}+n_{s_R}\right)} \\
&= \frac{2\gamma_s^2\left(\sum_{i\in s}|g_i|\right)^2}{h\left(n_{s_L}+n_{s_R}\right)} = \frac{2\gamma_s^2\left(\sum_{i\in s}|g_i|\right)^2}{hn_s}
\end{aligned} \tag{18}$$

If $\frac{\sum_{i\in s}|g_i|}{n_s} \leq \epsilon$, then suppose $s_1', ..., s_m'$ are all descendant leaves of s, then the average prediction values by current iteration for data in $\mathcal{D}_s$ in current tree is

$$\frac{\sum_{i=1}^m n_{s_i'}\left|w_{s_i'}^*\right|}{\sum_{i=1}^m n_{s_i'}} = \frac{\sum_{i=1}^m\left|\sum_{i\in s_i'} g_i\right|}{\sum_{i=1}^m n_{s_i'}h} \leq \frac{\sum_{i\in s}|g_i|}{n_s h} \leq \frac{\epsilon}{h}, \tag{19}$$

which guarantees that the first conclusion holds.

If $\frac{\sum_{i\in s}|g_i|}{n_s} > \epsilon$, let $\epsilon_i = \delta_g\widetilde{g}_i - g_i$, thus $|\epsilon_i| \leq \delta_g$ and $\mathbb{E}[\epsilon_i] = 0$. Then

$$\begin{aligned}
\left|\widetilde{\mathcal{L}}_{s_1}^* - \mathcal{L}_{s_1}^*\right| &= \left|\frac{\left(\sum_{i\in s_1} g_i + \epsilon_i\right)^2}{2hn_{s_1}} - \frac{\left(\sum_{i\in s_1} g_i\right)^2}{2hn_{s_1}}\right| \\
&= \frac{1}{2hn_{s_1}}\left|\sum_{i\in s_1} 2g_i + \epsilon_i\right|\left|\sum_{i\in s_1}\epsilon_i\right| \leq \frac{1}{2hn_{s_1}}\left(\left|\sum_{i\in s_1} 2g_i\right|\left|\sum_{i\in s_1}\epsilon_i\right| + \left|\sum_{i\in s_1}\epsilon_i\right|^2\right)
\end{aligned} \tag{20}$$

Note that ϵ_i's are independent random variables. Let $t_{s_1} = \delta_g\sqrt{2n_{s_1}\ln\frac{4}{\delta}}$, then by Hoeffding's inequality,

$$P\left(\left|\sum_{i\in n_{s_1}}\epsilon_i\right| \geq t_{s_1}\right) \leq 2\exp\left(-\frac{2t_{s_1}^2}{n_{s_1}\left(2\delta_g\right)^2}\right) = \frac{\delta}{2}. \tag{21}$$

Then with a probability of at least $1-\frac{\delta}{2}$

$$\begin{aligned}\left|\widetilde{\mathcal{L}}_{s_1}^* - \mathcal{L}_{s_1}^*\right| &\leq \frac{1}{hn_{s_1}}\left(\left|\sum_{i\in s_1} g_i\right|\cdot\delta_g\sqrt{2n_{s_1}\ln\frac{4}{\delta}} + \delta_g^2 n_{s_1}\ln\frac{4}{\delta}\right) \\ &= \frac{\left|\sum_{i\in s_1} g_i\right|\cdot\delta_g\sqrt{\frac{2\ln\frac{4}{\delta}}{n_{s_1}}}}{h} + \frac{\delta_g^2\ln\frac{4}{\delta}}{h}.\end{aligned} \tag{22}$$

We have

$$\begin{aligned}\frac{\left|\sum_{i\in s_1} g_i\right|\cdot\delta_g\sqrt{\frac{2\ln\frac{4}{\delta}}{n_{s_1}}}}{h\mathcal{G}_s^*} &\leq \frac{\sum_{i\in s_1}|g_i|\cdot\delta_g\sqrt{\frac{2\ln\frac{4}{\delta}}{n_{s_1}}}}{\frac{2\gamma_s^2\left(\sum_{i\in s}|g_i|\right)^2}{n_s}} \leq \frac{\delta_g\sqrt{\frac{2\ln\frac{4}{\delta}}{n_{s_1}}}}{\frac{2\gamma_s^2\left(\sum_{i\in s}|g_i|\right)}{n_s}} \\ &\leq \frac{\delta_g\sqrt{\frac{2\ln\frac{4}{\delta}}{n_{s_1}}}}{2\gamma_s^2\epsilon} = \frac{\max_{i\in[N]}|g_i|}{2\gamma_s^2\epsilon\left(2^{B-1}-1\right)}\sqrt{\frac{2\ln\frac{4}{\delta}}{n_{s_1}}}.\end{aligned} \tag{23}$$

And since

$$\mathcal{G}_s^* \geq \frac{2\gamma_s^2\left(\sum_{i\in s}|g_i|\right)^2}{hn_s} > \frac{2\gamma_s^2\epsilon^2 n_s}{h}, \tag{24}$$

we have

$$\frac{\delta_g^2\ln\frac{4}{\delta}}{h\mathcal{G}_s^*} \leq \frac{\delta_g^2\ln\frac{4}{\delta}}{2\gamma_s^2\epsilon^2 n_s} = \frac{\left(\max_{i\in[N]}|g_i|\right)^2\ln\frac{4}{\delta}}{2\gamma_s^2\epsilon^2\left(2^{B-1}-1\right)^2 n_s}. \tag{25}$$

Thus with a probability of at least $1-\frac{\delta}{2}$,

$$\begin{aligned}\frac{\left|\widetilde{\mathcal{L}}_{s_1}^* - \mathcal{L}_{s_1}^*\right|}{\mathcal{G}_s^*} &\leq \frac{\max_{i\in[N]}|g_i|}{2\gamma_s^2\epsilon\left(2^{B-1}-1\right)}\sqrt{\frac{2\ln\frac{4}{\delta}}{n_{s_1}}} + \frac{\left(\max_{i\in[N]}|g_i|\right)^2\ln\frac{4}{\delta}}{2\gamma_s^2\epsilon^2\left(2^{B-1}-1\right)^2 n_s} \\ &\leq \frac{\max_{i\in[N]}|g_i|}{\gamma_s^2\epsilon\cdot 2^{B-1}}\sqrt{\frac{2\ln\frac{4}{\delta}}{n_{s_1}}} + \frac{\left(\max_{i\in[N]}|g_i|\right)^2\ln\frac{4}{\delta}}{2\gamma_s^2\epsilon^2 n_s\cdot 4^{B-2}}\end{aligned} \tag{26}$$

Similarly, with a probability of at least $1-\frac{\delta}{2}$

$$\frac{\left|\widetilde{\mathcal{L}}_{s_2}^* - \mathcal{L}_{s_2}^*\right|}{\mathcal{G}_s^*} \leq \frac{\max_{i\in[N]}|g_i|}{\gamma_s^2\epsilon\cdot 2^{B-1}}\sqrt{\frac{2\ln\frac{4}{\delta}}{n_{s_2}}} + \frac{\left(\max_{i\in[N]}|g_i|\right)^2\ln\frac{4}{\delta}}{2\gamma_s^2\epsilon^2 n_s\cdot 4^{B-2}} \tag{27}$$

By union bound, with a probability of at least $1-\delta$

$$\begin{aligned}&\frac{\left|\widetilde{\mathcal{G}}_{s\to s_1,s_2} - \mathcal{G}_{s\to s_1,s_2}\right|}{\mathcal{G}_s^*} \leq \frac{\left|\widetilde{\mathcal{L}}_{s_1}^* - \mathcal{L}_{s_1}^*\right| + \left|\widetilde{\mathcal{L}}_{s_2}^* - \mathcal{L}_{s_2}^*\right|}{\mathcal{G}_s^*} \\ &\leq \frac{\max_{i\in[N]}|g_i|\sqrt{2\ln\frac{4}{\delta}}}{\gamma_s^2\epsilon\cdot 2^{B-1}}\left(\sqrt{\frac{1}{n_{s_1}}} + \sqrt{\frac{1}{n_{s_2}}}\right) + \frac{\left(\max_{i\in[N]}|g_i|\right)^2\ln\frac{4}{\delta}}{\gamma_s^2\epsilon^2 n_s\cdot 4^{B-2}}\end{aligned} \tag{28}$$

B.3 Loss Functions with Non-constant Hessians

Commonly used loss functions for binary-classification, multi-classification, and ranking have non-constant hessian values. Note that all these loss functions have non-negative hessian values. We analyze the error caused by quantization for these functions in this section. Let $\overline{h}_s = \frac{\sum_{i \in s} h_i}{n_s}$ to be the average of hessian values in leaf s. We have the following theorem.

Theorem B.3.1 For loss functions with non-constant hessian values, if Assumption 5.2 holds for the subset $\mathcal{D}_s$ in leaf s for some $\gamma_s > 0$, then with stochastic rounding and leaf-value refitting, for any $\epsilon > 0$ and $\delta > 0$, at least one of the following conclusions holds:

1. With any split of leaf s and its descendants, the resultant weighted average (weighted by h_i) of absolute values of prediction values by the tree in current boosting iteration for data in $\mathcal{D}_s$ is no greater than ϵ.

2. For any split $s \to s_1, s_2$ of leaf s, if $n_{s_1} \geq \frac{8\delta_h^2 \ln 8/\delta}{\overline{h}_{s_1}^2}$ and $n_{s_2} \geq \frac{8\delta_h^2 \ln 8/\delta}{\overline{h}_{s_2}^2}$, i.e. $n_{s_1} \leq \frac{\left(\sum_{i \in s_1} h_i\right)^2}{8\delta_h^2 \ln 8/\delta}$ and $n_{s_2} \leq \frac{\left(\sum_{i \in s_2} h_i\right)^2}{8\delta_h^2 \ln 8/\delta}$, then with a probability of at least $1 - \delta$

$$\begin{aligned}
\frac{\left|\widetilde{\mathcal{G}}_{s \to s_1, s_2} - \mathcal{G}_{s \to s_1, s_2}\right|}{\mathcal{G}_s^*} \leq & \frac{\delta_g \sqrt{2 \ln 8/\delta}}{\gamma_s^2 \epsilon} \left(\frac{1}{\overline{h}_{s_1} \sqrt{n_{s_1}}} + \frac{1}{\overline{h}_{s_2} \sqrt{n_{s_2}}} \right) \\
& + \frac{\overline{h}_s \delta_h \sqrt{2 \ln 8/\delta}}{2\gamma_s^2} \left(\frac{n_s}{\overline{h}_{s_1}^2 n_{s_1} \sqrt{n_{s_1}}} + \frac{n_s}{\overline{h}_{s_2}^2 n_{s_2} \sqrt{n_{s_2}}} \right) \\
& + \frac{\delta_g^2 \ln 8/\delta}{\gamma_s^2 \overline{h}_s \epsilon^2} \left(\frac{1}{\overline{h}_{s_1} n_s} + \frac{1}{\overline{h}_{s_2} n_s} \right)
\end{aligned} \tag{29}$$

Proof of Theorem B.3.1 By leaf-wise weak learnability (Assumption 5.2), there exists a split $s \to s_L, s_R$ and $\gamma_s > 0$ for s s.t. for data in $\mathcal{D}_s$, with binary labels $c(\mathbf{x}_i) = \text{sign}(g_i)$ and weights $w_i = |g_i|$, the split results in a stump with weighted binary-classification error is $\frac{1}{2} - \gamma_s$. Similar to the case of loss functions with constant hessian, we define s_L^+, s_L^-, s_R^+ and s_R^-, and first derive a lower bound for $\mathcal{G}_s^*$,

$$\begin{aligned}
\mathcal{G}_s^* \geq \mathcal{G}_{s \to s_L, s_R} &= \frac{\left(\sum_{i \in s_L} g_i\right)^2}{2 \sum_{i \in s_L} h_i} + \frac{\left(\sum_{i \in s_R} g_i\right)^2}{2 \sum_{i \in s_R} h_i} \\
&\geq \frac{\left(\left|\sum_{i \in s_L} g_i\right| + \left|\sum_{i \in s_R} g_i\right|\right)^2}{2 \sum_{i \in s} h_i} \\
&= \frac{\left(\sum_{i \in s_L^+} |g_i| - \sum_{i \in s_L^-} |g_i| + \sum_{i \in s_R^+} |g_i| - \sum_{i \in s_R^-} |g_i|\right)^2}{2 \sum_{i \in s} h_i} \\
&= \frac{\left(\sum_{i \in s_L^+ \cup s_R^+} |g_i| - \sum_{i \in s_L^- \cup s_R^-} |g_i|\right)^2}{2 \sum_{i \in s} h_i} \\
&= \frac{\left(\left(\frac{1}{2} + \gamma_s\right) \sum_{i \in s} |g_i| - \left(\frac{1}{2} - \gamma_s\right) \sum_{i \in s} |g_i|\right)^2}{2 \sum_{i \in s} h_i} \\
&= \frac{2\gamma_s^2 \left(\sum_{i \in s} |g_i|\right)^2}{\sum_{i \in s} h_i}
\end{aligned} \tag{30}$$

Let $\xi_i = \delta_h \widetilde{h}_i - h_i$ and $\epsilon_i = \delta_g \widetilde{g}_i - g_i$, thus $|\xi_i| \leq \delta_h$, $|\epsilon_i| \leq \delta_g$, $\mathbb{E}[\xi_i] = 0$ and $\mathbb{E}[\epsilon_i] = 0$. We then bound the error of $\sum_{i \in s_1} h_i$,

$$\left| \sum_{i \in s_1} (h_i + \xi_i) - \sum_{i \in s_1} h_i \right| = \left| \sum_{i \in s_1} \xi_i \right|. \tag{31}$$

Note that ξ_i's and ϵ_i's are independent variables. Let $t'_{s_1} = \delta_h\sqrt{2n_{s_1}\ln\frac{8}{\delta}}$, then by Hoeffding's inequality,

$$P\left(\left|\sum_{i\in s_1}\xi_i\right| \geq t'_{s_1}\right) \leq 2\exp\left(-\frac{2{t'_{s_1}}^2}{n_{s_1}(2\delta_h)^2}\right) = \frac{\delta}{4} \tag{32}$$

Similarly, let $t''_{s_1} = \delta_g\sqrt{2n_{s_1}\ln\frac{8}{\delta}}$, then by Hoeffding's inequality we can bound the error of $\sum_{i\in s_1} g_i$,

$$P\left(\left|\sum_{i\in s_1}\epsilon_i\right| \geq t''_{s_1}\right) \leq 2\exp\left(-\frac{2{t''_{s_1}}^2}{n_{s_1}(2\delta_g)^2}\right) = \frac{\delta}{4} \tag{33}$$

If $\frac{\sum_{i\in s}|g_i|}{\sum_{i\in s}h_i} \leq \epsilon$, then suppose $s'_1, ..., s'_m$ are all descendant leaves of s, then the average (weighted by h_i's) prediction value for data in $\mathcal{D}_s$ in current tree is

$$\frac{\sum_{i=1}^m\sum_{i\in s'_i}h_i\left|w^*_{s'_i}\right|}{\sum_{i=1}^m\sum_{i\in s'_i}h_i} = \frac{\sum_{i=1}^m\left|\sum_{i\in s'_i}g_i\right|}{\sum_{i=1}^m\sum_{i\in s'_i}h_i} \leq \frac{\sum_{i\in s}|g_i|}{\sum_{i\in s}h_i} \leq \epsilon \tag{34}$$

which guarantees that the first conclusion holds.

If $\frac{\sum_{i\in s}|g_i|}{\sum_{i\in s}h_i} > \epsilon$, then we denote $\overline{h}_{s_1} = \frac{\sum_{i\in s_1}h_i}{n_{s_1}}$. If $n_{s_1} \geq \frac{8\delta_h^2\ln 8/\delta}{\overline{h}_{s_1}^2}$, i.e. $n_{s_1} \leq \frac{\left(\sum_{i\in s_1}h_i\right)^2}{8\delta_h^2\ln 8/\delta}$, then we have $\left|\sum_{i\in s_1}\xi_i\right| \leq \frac{\sum_i h_i}{2}$ with a probability of at least $1-\frac{\delta}{4}$ by equation (32). Thus with a probability of at least $1-\frac{\delta}{2}$,

$$\begin{aligned}
\left|\widetilde{\mathcal{L}}^*_{s_1} - \mathcal{L}^*_{s_1}\right| &= \frac{1}{2}\left|\frac{\left(\sum_{i\in s_1}g_i+\epsilon_i\right)^2}{\sum_{i\in s_1}h_i+\xi_i} - \frac{\left(\sum_{i\in s_1}g_i\right)^2}{\sum_{i\in s_1}h_i}\right| \\
&= \frac{1}{2}\left|\frac{\left(\sum_{i\in s_1}g_i+\epsilon_i\right)^2\left(\sum_{i\in s_1}h_i\right) - \left(\sum_{i\in s_1}h_i+\xi_i\right)\left(\sum_{i\in s_1}g_i\right)^2}{\left(\sum_{i\in s_1}h_i+\xi_i\right)\left(\sum_{i\in s_1}h_i\right)}\right| \\
&\leq \frac{\left|\left(\sum_{i\in s_1}g_i+\epsilon_i\right)^2\left(\sum_{i\in s_1}h_i\right) - \left(\sum_{i\in s_1}g_i\right)^2\left(\sum_{i\in s_1}h_i\right)\right|}{2\left(\sum_{i\in s_1}h_i+\xi_i\right)\left(\sum_{i\in s_1}h_i\right)} \\
&+ \frac{\left|\left(\sum_{i\in s_1}g_i\right)^2\left(\sum_{i\in s_1}h_i\right) - \left(\sum_{i\in s_1}g_i\right)^2\left(\sum_{i\in s_1}h_i+\xi_i\right)\right|}{2\left(\sum_{i\in s_1}h_i+\xi_i\right)\left(\sum_{i\in s_1}h_i\right)} \\
&\leq \frac{\left|\sum_{i\in s_1}\epsilon_i\right|\left|\sum_{i\in s_1}g_i\right|}{\sum_{i\in s_1}h_i+\xi_i} + \frac{\left|\sum_{i\in s_1}\epsilon_i\right|^2}{2\left(\sum_{i\in s_1}h_i+\xi_i\right)} + \frac{\left(\sum_{i\in s_1}g_i\right)^2\left|\sum_{i\in s_1}\xi_i\right|}{2\left(\sum_{i\in s_1}h_i+\xi_i\right)\left(\sum_{i\in s_1}h_i\right)} \\
&\leq \frac{2\left|\sum_{i\in s_1}\epsilon_i\right|\left|\sum_{i\in s_1}g_i\right|}{\sum_{i\in s_1}h_i} + \frac{\left|\sum_{i\in s_1}\epsilon_i\right|^2}{\sum_{i\in s_1}h_i} + \frac{\left(\sum_{i\in s_1}g_i\right)^2\left|\sum_{i\in s_1}\xi_i\right|}{\left(\sum_{i\in s_1}h_i\right)^2}
\end{aligned} \tag{35}$$

Because

$$\frac{2\left|\sum_{i\in s_1}\epsilon_i\right|\left|\sum_{i\in s_1}g_i\right|}{\mathcal{G}^*_s\sum_{i\in s_1}h_i} \leq \frac{\left(\sum_{i\in s}h_i\right)\left|\sum_{i\in s_1}\epsilon_i\right|\left|\sum_{i\in s_1}g_i\right|}{\gamma_s^2\left(\sum_{i\in s}|g_i|\right)^2\sum_{i\in s_1}h_i} \leq \frac{\left|\sum_{i\in s_1}\epsilon_i\right|}{\gamma_s^2\overline{h}_{s_1}\epsilon n_{s_1}} \leq \frac{\delta_g\sqrt{2\ln 8/\delta}}{\gamma_s^2\overline{h}_{s_1}\epsilon\sqrt{n_{s_1}}}, \tag{36}$$

$$\frac{\left|\sum_{i\in s_1}\epsilon_i\right|^2}{\mathcal{G}^*_s\sum_{i\in s_1}h_i} \leq \frac{\left(\sum_{i\in s}h_i\right)\left|\sum_{i\in s_1}\epsilon_i\right|^2}{2\gamma_s^2\left(\sum_{i\in s}|g_i|\right)^2\sum_{i\in s_1}h_i} \leq \frac{\left|\sum_{i\in s_1}\epsilon_i\right|^2}{2\gamma_s^2\overline{h}_s\overline{h}_{s_1}\epsilon^2 n_s n_{s_1}} \leq \frac{\delta_g^2\ln 8/\delta}{\gamma_s^2\overline{h}_s\overline{h}_{s_1}\epsilon^2 n_s}, \tag{37}$$

$$\frac{\left(\sum_{i\in s_1}g_i\right)^2\left|\sum_{i\in s_1}\xi_i\right|}{\mathcal{G}^*_s\left(\sum_{i\in s_1}h_i\right)^2} \leq \frac{n_s\overline{h}_s\left|\sum_{i\in s_1}\xi_i\right|}{2\gamma_s^2\left(\sum_{i\in s_1}h_i\right)^2} \leq \frac{n_s\overline{h}_s\delta_h\sqrt{2\ln 8/\delta}}{2\gamma_s^2\overline{h}_{s_1}^2 n_{s_1}\sqrt{n_{s_1}}}. \tag{38}$$

Thus with probability at least $1 - \frac{\delta}{2}$ we have

$$\frac{\left|\widetilde{\mathcal{L}}^*_{s_1} - \mathcal{L}^*_{s_1}\right|}{\mathcal{G}^*_s} \leq \frac{\sqrt{2\ln 8/\delta}}{\gamma_s^2}\left(\frac{\delta_g}{\overline{h}_{s_1}\epsilon\sqrt{n_{s_1}}} + \frac{\overline{h}_s\delta_h n_s}{2\overline{h}^2_{s_1} n_{s_1}\sqrt{n_{s_1}}}\right) + \frac{\delta_g^2\ln 8/\delta}{\gamma_s^2\overline{h}_s\overline{h}_{s_1}\epsilon^2 n_s}. \tag{39}$$

Similarly with probability at least $1 - \frac{\delta}{2}$ we have

$$\frac{\left|\widetilde{\mathcal{L}}^*_{s_2} - \mathcal{L}^*_{s_2}\right|}{\mathcal{G}^*_s} \leq \frac{\sqrt{2\ln 8/\delta}}{\gamma_s^2}\left(\frac{\delta_g}{\overline{h}_{s_2}\epsilon\sqrt{n_{s_2}}} + \frac{\overline{h}_s\delta_h n_s}{2\overline{h}^2_{s_2} n_{s_2}\sqrt{n_{s_2}}}\right) + \frac{\delta_g^2\ln 8/\delta}{\gamma_s^2\overline{h}_s\overline{h}_{s_2}\epsilon^2 n_s}. \tag{40}$$

And finally, with probability at least $1 - \delta$,

$$\begin{aligned}\frac{\left|\widetilde{\mathcal{G}}_{s\to s_1,s_2} - \mathcal{G}_{s\to s_1,s_2}\right|}{\mathcal{G}^*_s} \leq\ & \frac{\delta_g\sqrt{2\ln 8/\delta}}{\gamma_s^2\epsilon}\left(\frac{1}{\overline{h}_{s_1}\sqrt{n_{s_1}}} + \frac{1}{\overline{h}_{s_2}\sqrt{n_{s_2}}}\right) \\ &+ \frac{\overline{h}_s\delta_h\sqrt{2\ln 8/\delta}}{2\gamma_s^2}\left(\frac{n_s}{\overline{h}^2_{s_1} n_{s_1}\sqrt{n_{s_1}}} + \frac{n_s}{\overline{h}^2_{s_2} n_{s_2}\sqrt{n_{s_2}}}\right) \\ &+ \frac{\delta_g^2\ln 8/\delta}{\gamma_s^2\overline{h}_s\epsilon^2}\left(\frac{1}{\overline{h}_{s_1} n_s} + \frac{1}{\overline{h}_{s_2} n_s}\right).\end{aligned} \tag{41}$$

C Experiment Details

In this section, we provide more details about the results, experiment environments, and hyperparameter settings.

C.1 Variance of Quantized Training

In Table 2 the metric values are averaged over 5 random seeds. The seeds are used to generate random numbers for stochastic rounding. We omit the standard deviation in Table 2 due to limited space. Here we provide a full table with standard deviation listed in Table 4. Note that we report the metric on the best iteration in the test sets. As we can see the variance caused by stochastic rounding in quantization is small, and quantized training is quite stable with different random seeds.

C.2 Accuracy of Quantized Training on GPU

Table 5 shows the accuracy of quantized training on GPU, averaged over 5 random seeds for stochastic rounding with leaf-value refitting. For the GPU version, we run up to 5 bits for gradient discretization. As we can see, for most datasets a comparable performance is achieved with quantized training. Note that we report the metric on the best iteration in the test sets.

C.3 Detailed Training Time

Table 6 shows the detailed training time records. Note that training time numbers are evaluated with 5 random seeds. Time for data loading and pre-processing is excluded. When recording training time, all metric evaluations are disabled. For quantized training, the number of bits does not influence the training time much. This indicates that more acceleration can be achieved for low-bitwidth gradients like 2-bit or 3-bit, with better hardware support for operations of low-bitwidth integers.

C.4 Experiment Environments

Table 7 and 8 list the experiment environments used in this paper for standalone machines. For CPU clusters in distributed experiments, we use 16 nodes each with one Intel(R) Xeon(R) CPU E5-2673 v4 or Intel(R) Xeon(R) Platinum 8171M CPU. The nodes are connected by a network of bandwidth between $7 \sim 8$Gbps (tested with iperf[9]).

[9] https://iperf.fr/

Table 4: Accuracy with standard variance, w.r.t. different quantized bits (CPU version).

Algorithm	Binary Classification					Regression	Ranking	
	Higgs↑	Epsilon↑	Kitsune↑	Criteo↑	Bosch↑	Year↓	Yahoo LTR↑	LETOR↑
XGBoost	0.845778 ±.000000	0.950210 ±.000000	0.948329 ±.000000	0.802030 ±.000000	**0.706423** ±.000000	8.954460 ±.000000	**0.794919** ±.000000	0.505058 ±.000000
CatBoost	0.845425 ±.000144	0.943211 ±.000073	0.944557 ±.008969	0.803150 ±.000232	0.687795 ±.000282	8.951745 ±.005324	0.794215 ±.000196	0.519952 ±.000568
LightGBM	0.845694 ±.000162	0.950203 ±.000144	0.950561 ±.000638	0.803791 ±.000052	0.703471 ±.001110	8.956278 ±.004803	0.793792 ±.000259	0.524191 ±.000360
2-bit SR_{refit}	0.845587 ±.000042	0.949472 ±.000166	0.952703 ±.000729	0.803293 ±.000091	0.700040 ±.001759	8.953388 ±.012679	0.788579 ±.000357	0.519067 ±.000681
3-bit SR_{refit}	0.845725 ±.000193	0.949884 ±.000095	0.951309 ±.001352	0.803768 ±.000050	0.702025 ±.001519	**8.937374** ±.007487	0.791077 ±.000894	0.522220 ±.000721
4-bit SR_{refit}	0.845507 ±.000127	0.950049 ±.000072	0.950911 ±.001557	0.803783 ±.000073	0.702959 ±.000607	8.942898 ±.008327	0.792664 ±.000467	0.523702 ±.000653
5-bit SR_{refit}	0.845706 ±.000171	0.950298 ±.000108	0.949229 ±.001964	0.803766 ±.000053	0.703242 ±.001041	8.948542 ±.003732	0.793166 ±.000487	**0.524616** ±.000439
2-bit $SR_{no\ refit}$	**0.846713** ±.000184	0.944509 ±.000174	0.952974 ±.001024	0.803750 ±.000068	0.701399 ±.001663	9.112302 ±.014516	0.764862 ±.000858	0.486193 ±.001789
3-bit $SR_{no\ refit}$	0.846040 ±.000178	0.949593 ±.000119	0.951385 ±.001158	**0.803922** ±.000064	0.702460 ±.000768	8.990034 ±.009847	0.780041 ±.000618	0.507689 ±.001126
4-bit $SR_{no\ refit}$	0.845816 ±.000304	0.950127 ±.000172	0.951197 ±.001067	0.803812 ±.000074	0.704053 ±.000277	8.955256 ±.003074	0.787575 ±.001173	0.515767 ±.000448
5-bit $SR_{no\ refit}$	0.845842 ±.000119	0.950275 ±.000234	0.949794 ±.002275	0.803790 ±.000096	0.702717 ±.001075	8.952768 ±.009403	0.791631 ±.000590	0.520900 ±.001087
2-bit RN_{refit}	0.795991 ±.000582	0.889149 ±.000856	0.962201 ±.000820	0.779906 ±.000323	0.685407 ±.001055	9.429014 ±.017197	0.765103 ±.000918	0.454512 ±.005565
3-bit RN_{refit}	0.830506 ±.000495	0.944329 ±.000319	**0.966606** ±.001074	0.782732 ±.000210	0.688372 ±.000351	9.062854 ±.014744	0.772364 ±.000822	0.476874 ±.001500
4-bit RN_{refit}	0.840747 ±.000241	0.949946 ±.000207	0.961938 ±.001970	0.795803 ±.000099	0.691163 ±.000698	8.968694 ±.005092	0.777347 ±.000969	0.487394 ±.003279
5-bit RN_{refit}	0.843820 ±.000073	**0.950457** ±.000071	0.962427 ±.001150	0.802438 ±.000083	0.698529 ±.000425	8.952418 ±.003649	0.784333 ±.000612	0.494828 ±.001543
2-bit $RN_{no\ refit}$	0.836683 ±.000468	0.925220 ±.001545	0.946016 ±.005072	0.768338 ±.000202	0.695089 ±.001510	10.685840 ±.001819	0.632058 ±.005683	0.203732 ±.005507
3-bit $RN_{no\ refit}$	0.843482 ±.000306	0.946850 ±.000399	0.940961 ±.006586	0.791709 ±.000379	0.697933 ±.001424	9.377560 ±.042545	0.732487 ±.001121	0.350127 ±.004163
4-bit $RN_{no\ refit}$	0.845788 ±.000176	0.949676 ±.000126	0.949228 ±.002973	0.802689 ±.000096	0.702767 ±.000408	8.969828 ±.005646	0.765432 ±.000426	0.437317 ±.001514
5-bit $RN_{no\ refit}$	0.845765 ±.000248	0.950307 ±.000150	0.952420 ±.003838	0.803645 ±.000102	0.695559 ±.000582	8.965400 ±.002101	0.782608 ±.000514	0.485752 ±.001105

C.5 Hyperparameter Settings

For all accuracy and training time evaluations in this paper, we use the hyperparameters of LightGBM, XGBoost, and Catboost as listed in Table 9, 10, and 11, except for the Bosch dataset. For the Bosch dataset, we use learing_rate 0.015 for LightGBM and CatBoost, eta 0.015 for XGBoost, num_leaves 45 for LightGBM, max_leaves 45 for XGBoost and CatBoost, and keep other hyperparameters the same as in the tables. The hyperparameters are chosen so that all these algorithms have similar tree sizes for a fair comparison of training time. In addition, since our GPU machine has 24 physical CPU cores, we set the number of threads of all experiments on GPU to be 24. For training time of LightGBM on CPU, we use force_col_wise=true mode For Bosch, Yahoo LTR, Year, and Epsilon (except 2-bit and 3-bit quantized training). For all other cases we use force_row_wise=true mode of LightGBM CPU version.

The git commit used for CatBoost is 2617a690c77c9c8da1c22d1750b5fbe0eb0eb36d (Date: Thu Jan 12 05:29:47 2023 +0300). And for XGBoost it is cfa994d57fb713e203a494319a6d33fcd007adf8 (Date: Wed Jan 11 05:51:14 2023 +0800). For LightGBM, we use the commit 5df7d516f3e64e8431aecfea892a850afeb787bb (Date: Fri Jan 13 17:02:42 2023 +0200) for the FP32 experiments on CPU. For other experiments of LightGBM, we use the version provided in our Github link https://github.com/Quantized-GBDT/Quantized-GBDT.

Table 5: Accuracy with standard variance, w.r.t. different quantized bits (GPU version).

Algorithm	Binary-Class					Regression	Ranking	
	Higgs↑	Epsilon↑	Kitsune↑	Criteo↑	Bosch↑	Year↓	Yahoo LTR↑	LETOR↑
XGBoost	0.846035 ±.000000	**0.950305** ±.000000	0.949952 ±.000000	0.802159 ±.000000	0.705095 ±.000000	**8.949647** ±.000000	0.794026 ±.000000	0.506957 ±.000000
CatBoost	0.845237 ±.000211	0.949214 ±.000115	0.953293 ±.001535	0.803766 ±.000083	**0.713931** ±.000685	8.968944 ±.004703	0.794176 ±.000175	N/A
LightGBM+	0.845729 ±.000081	0.950228 ±.000110	**0.955914** ±.000962	0.803805 ±.000122	0.703479 ±.001100	8.956194 ±.004720	**0.794892** ±.000688	**0.526611** ±.000614
2-bit SR_{refit}	**0.846582** ±.000159	0.945205 ±.000255	0.952898 ±.001554	0.803594 ±.000077	0.700896 ±.001099	9.107948 ±.007273	0.769811 ±.000495	0.492340 ±.001090
3-bit SR_{refit}	0.845877 ±.000255	0.949494 ±.000277	0.951672 ±.002186	**0.803847** ±.000059	0.702262 ±.000903	8.980230 ±.006827	0.784282 ±.000535	0.511955 ±.001200
4-bit SR_{refit}	0.845872 ±.000199	0.950176 ±.000066	0.951918 ±.001138	0.803799 ±.000089	0.703318 ±.001258	8.962148 ±.016629	0.790682 ±.000536	0.519747 ±.000716
5-bit SR_{refit}	0.845849 ±.000238	0.950177 ±.000174	0.950538 ±.000354	0.803827 ±.000095	0.703152 ±.000360	8.953900 ±.004574	0.793978 ±.000490	0.523701 ±.000390

C.6 Data Split and Preprocessing

For most datasets (Higgs, Epsilon, Yahoo, LETOR, Year, Bosch) we use the convention in previous works or the default split [34, 25, 3], without additional preprocessing. For Criteo, we encode the categorical features in the original dataset with target and count encoding. We use the `train.txt` file of the Kaggle version of the Criteo dataset, with the first $41,256,555$ rows as the training set and the last $4,584,061$ rows as the test set. For Kitsune, we select the first 80% packets in each attack method to form the training set, and the final 20% packets to form the test set. The datasets can be freely downloaded from https://pretrain.blob.core.windows.net/quantized-gbdt/dataset.zip.

D Discussion on Loss Functions with Non-constant Hessians

Appendix B.3 provides the theoretical analysis and proof for the error caused by quantization for loss functions with non-constant hessians. The assumption is a little bit stronger than constant hessian loss functions in that we are expecting the average hessian values per leaf won't be too small, so that $n_{s_1} \geq \frac{8\delta_h^2 \ln 8/\delta}{\overline{h}_{s_1}^2}$ and $n_{s_2} \geq \frac{8\delta_h^2 \ln 8/\delta}{\overline{h}_{s_2}^2}$ hold. Figure 7 shows the average hessian values in each iteration with 3-bit gradients. We first calculate the average hessian values for all leaves in each iteration. Then we plot the mean of the average hessian values over the leaves in each iteration in solid blue curves, with the shadow area indicating the range between 10% and 90% percentiles over the leaves in each iteration. For most leaves, the average hessian values are not too small. And it is easy to meet the condition $n_s \geq \frac{8\delta_h^2 \ln 8/\delta}{\overline{h}_s^2}$ with enough training data. For example, suppose $\overline{h}_s = 0.01$, then for binary classification, with 3-bit gradients, $\delta_h = \frac{0.25}{6} = \frac{1}{24}$. Let $\delta = 0.01$, then $\frac{8\delta_h^2 \ln 8/\delta}{\overline{h}_s^2} \approx 928$.

In addition, in the first conclusion of Theorem B.3.1 we consider weighted prediction values by h_i. Since with second-order approximation of the loss function, h_i influences how much a training sample contributes to the approximated loss by second-order Taylor expansion [5]. Thus, considering the weighted prediction values by h_i is meaningful.

Finally, the upper bound in Theorem B.3.1 requires a balanced split to be small. In other words, the data sizes in child nodes n_{s_1}, n_{s_2} shouldn't be significantly smaller than that in parent node n_s, so that the terms $\frac{n_s}{n_{s_1}\sqrt{n_{s_1}}}$ and $\frac{n_s}{n_{s_2}\sqrt{n_{s_2}}}$ can be bounded by a small value.

E New CUDA Framework of LightGBM

We implement a new CUDA version for LightGBM. Previous GPU versions of LightGBM only run histogram construction on GPUs. Our new implementation performs the whole training process

Table 6: Detailed time costs for different algorithms in different datasets (seconds).

	Algorithm	Higgs	Epsilon	Kitsune	Criteo	Bosch	Year	Yahoo LTR	LETOR
GPU total time	XGBoost	33.97 ±0.09	311.12 ±0.48	181.24 ±0.26	326.82 ±0.31	68.44 ±0.19	20.47 ±0.36	28.64 ±0.19	51.29 ±0.25
	CatBoost	61.10 ±3.98	105.00 ±0.00	80.20 ±1.72	187.80 ±1.17	22.12 ±0.30	33.96 ±2.04	59.22 ±0.37	N/A
	LightGBM+	29.05 ±0.22	87.12 ±0.39	77.43 ±0.50	102.33 ±1.61	21.41 ±0.44	24.33 ±0.30	30.79 ±0.20	41.79 ±0.42
	LightGBM+ 2-bit	24.78 ±0.31	**39.04** ±0.31	**38.26** ±0.12	61.04 ±0.27	12.57 ±0.29	**18.19** ±0.07	**23.09** ±0.47	**33.60** ±0.29
	LightGBM+ 3-bit	**24.45** ±0.36	39.25 ±0.31	38.63 ±0.52	59.93 ±0.37	12.60 ±0.39	18.24 ±0.18	24.93 ±0.42	33.87 ±0.38
	LightGBM+ 4-bit	24.53 ±0.26	39.82 ±0.15	40.00 ±0.24	**59.49** ±0.26	12.55 ±0.27	18.34 ±0.20	25.65 ±0.38	34.11 ±0.12
	LightGBM+ 5-bit	24.55 ±0.14	41.30 ±0.10	40.83 ±0.46	60.24 ±0.37	**12.08** ±0.24	18.41 ±0.15	25.50 ±0.40	34.36 ±0.07
CPU total time	XGBoost	109.16 ±2.06	1282.97 ±12.53	281.72 ±4.84	565.52 ±1.54	130.92 ±0.45	28.85 ±0.47	103.87 ±0.46	72.37 ±0.22
	CatBoost	1009.8 ±9.22	1283.4 ±3.32	1495.0 ±31.43	7702.2 ±55.16	998.4 ±30.36	95.8 ±1.17	588.2 ±4.83	865.4 ±9.13
	LightGBM	83.27 ±0.41	519.89 ±1.91	332.12 ±4.17	524.61 ±9.32	59.94 ±0.08	12.67 ±0.11	75.44 ±0.37	103.09 ±1.05
	LightGBM 2-bit	73.36 ±0.36	**426.50** ±5.25	215.91 ±3.05	444.28 ±7.39	46.63 ±0.47	12.94 ±0.16	61.50 ±0.41	**72.08** ±0.55
	LightGBM 3-bit	69.64 ±0.50	459.39 ±2.83	**207.96** ±2.45	440.68 ±38.97	47.35 ±0.30	12.79 ±0.22	**61.07** ±0.41	74.35 ±0.29
	LightGBM 4-bit	**69.30** ±0.40	458.62 ±2.55	208.99 ±1.23	**416.60** ±6.73	**46.45** ±0.28	11.90 ±0.23	61.15 ±0.43	77.66 ±1.94
	LightGBM 5-bit	69.86 ±0.93	457.68 ±2.81	211.53 ±4.20	423.80 ±19.23	47.52 ±1.54	**11.79** ±0.09	61.76 ±0.45	77.92 ±0.50
GPU Hist. time	LightGBM+	11.26 ±0.03	46.96 ±0.18	54.77 ±0.37	70.97 ±1.33	16.57 ±0.41	9.61 ±0.03	11.59 ±0.09	17.75 ±0.24
	LightGBM+ 2-bit	4.84 ±0.03	**12.11** ±0.07	16.41 ±0.13	21.74 ±0.10	8.52 ±0.29	**4.08** ±0.02	**8.23** ±0.25	**10.20** ±0.07
	LightGBM+ 3-bit	4.74 ±0.07	12.25 ±0.08	**16.07** ±0.15	21.14 ±0.07	8.49 ±0.32	4.11 ±0.02	8.54 ±0.27	10.29 ±0.09
	LightGBM+ 4-bit	4.74 ±0.04	12.51 ±0.03	16.45 ±0.12	**21.02** ±0.10	8.44 ±0.29	4.15 ±0.02	8.66 ±0.25	10.52 ±0.04
	LightGBM+ 5-bit	**4.70** ±0.04	13.54 ±0.02	16.78 ±0.09	21.31 ±0.15	**7.93** ±0.25	4.24 ±0.03	8.60 ±0.31	10.78 ±0.05
CPU Hist. time	LightGBM	50.74 ±0.33	458.46 ±1.74	253.07 ±5.00	385.98 ±9.60	53.08 ±0.19	6.68 ±0.09	58.53 ±0.26	66.39 ±1.05
	LightGBM 2-bit	32.82 ±0.26	**375.70** ±5.12	147.10 ±3.57	269.00 ±7.70	**39.80** ±0.54	5.99 ±0.14	43.59 ±0.33	**38.23** ±0.27
	LightGBM 3-bit	30.84 ±0.38	410.29 ±2.56	137.54 ±1.84	273.68 ±38.78	40.36 ±0.34	5.85 ±0.16	**43.47** ±0.36	40.21 ±0.18
	LightGBM 4-bit	**30.72** ±0.30	406.91 ±2.35	133.15 ±1.26	**252.20** ±6.67	39.88 ±0.26	5.23 ±0.15	43.73 ±0.27	43.23 ±0.92
	LightGBM 5-bit	30.97 ±0.48	406.26 ±2.83	**130.13** ±4.23	259.50 ±18.63	40.70 ±1.06	**5.14** ±0.06	44.80 ±0.44	43.68 ±0.35

including boosting (calculation of gradients and hessians) and tree learning on GPUs. We denote this new GPU version of LightGBM by LightGBM+ in our paper.

Table 7: Experiment Environments (CPU)

CPU	Intel(R) Xeon(R) Platinum 8370C
Cores	16 Physical CPU Cores
OS	Ubuntu 18.04
Memory	256 GB

Table 8: Experiment Environments (GPU)

CPU	Intel(R) Xeon(R) CPU E5-2690 v4
Cores	24 Physical CPU Cores
GPU	NVIDIA Tesla V100 PCIe 16GB
OS	Ubuntu 18.04
Memory	448 GB

Table 9: Hyperparameters of LightGBM

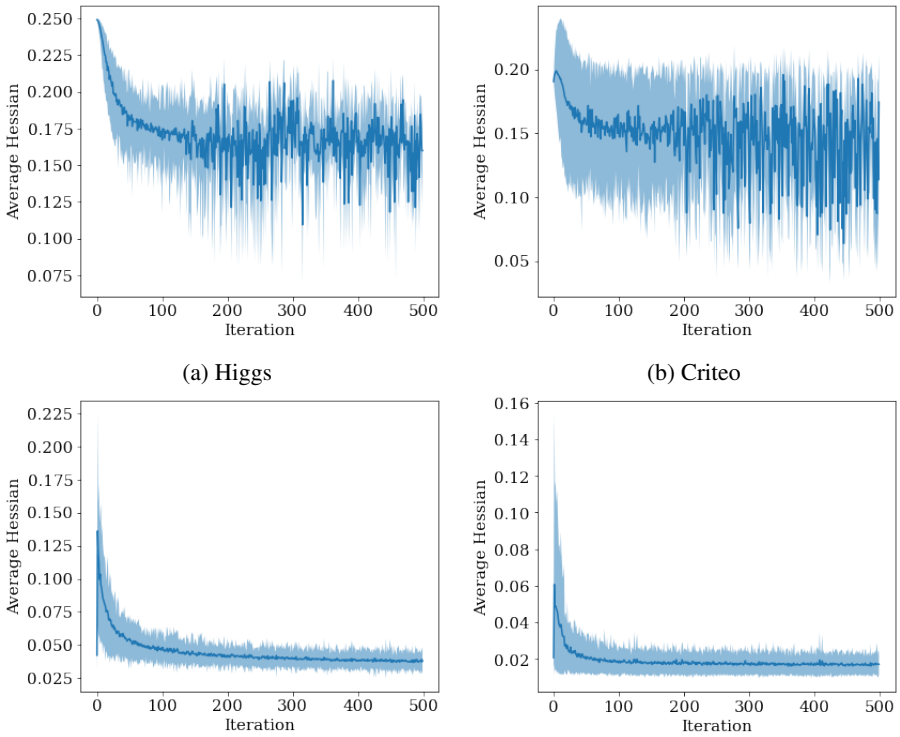
boosting_type	gbdt
learning_rate	0.1
min_child_weight	100
num_leaves	255
max_bin	255
num_iterations	500
num_threads	16
objective (binary)	binary
objective (regression)	regression
objective (ranking)	lambdarank
metric (binary)	auc
metric (regression)	rmse
metric (ranking)	ndcg@10

Table 10: Hyperparameters of XGBoost

tree_method	hist/gpu_hist
eta	0.1
max_depth	0
max_leaves	255
num_round	500
min_child_weight	100
nthread	16
gamma	0
lambda	0
alpha	0
objective (binary)	binary:logistic
objective (regression)	reg:linear
objective (ranking)	rank:pairwise
metric (binary)	auc
metric (regression)	rmse
metric (ranking)	ndcg@10

Table 11: Hyperparameters of CatBoost

thread_count	16
border_count	255
iterations	500
learning_rate	0.1
grow_policy	Lossguide
boosting_type	Plain
max_leaves	255
depth	256
min_data_in_leaf	100
objective (binary)	Logloss
objective (regression)	RMSE
objective (ranking)	YetiRank
metric (binary)	AUC
metric (regression)	RMSE
metric (ranking)	NDCG:top=10;type=Exp
bootstrap_type	No
random_strength	0
rsm	1



(a) Higgs

(b) Criteo

(c) Yahoo LTR

(d) LETOR

Figure 7: Average Hessian Values by Iteration with 3-Bit Gradients