Minimal Variance Sampling in Stochastic Gradient Boosting

Bulat Ibragimov, Gleb Gusev

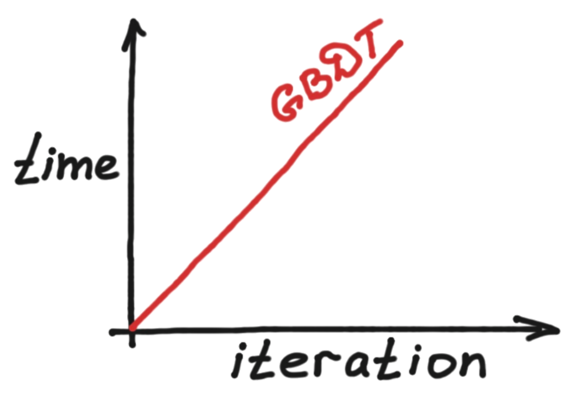
Gradient Boosted Decision Trees (GBDT)

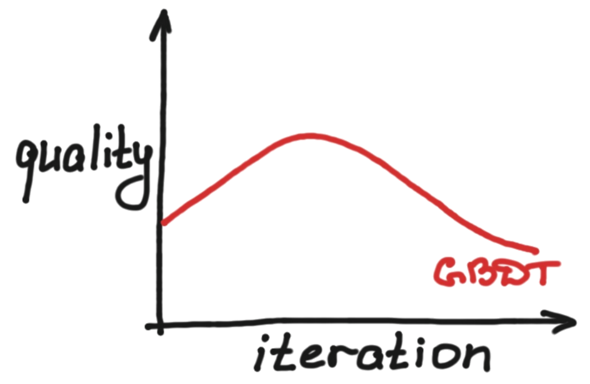
Iterative process of predicting negative gradient step

$$g_i^k = \frac{\partial L(y_i, \hat y_i)}{\partial \hat y_i}\bigg|_{\hat y_i = F_{k-1}(x_i)} \qquad h_i^k = \frac{\partial^2 L(y_i, \hat y_i)}{\partial \hat y_i^2}\bigg|_{\hat y_i = F_{k-1}(x_i)}$$

$$F_N(x_i) = \sum_{k=1}^{N} \alpha f_k(x_i) \approx \sum_{k=1}^{N} -\alpha \frac{g_i^k}{h_i^k}$$

f_k - decision tree function



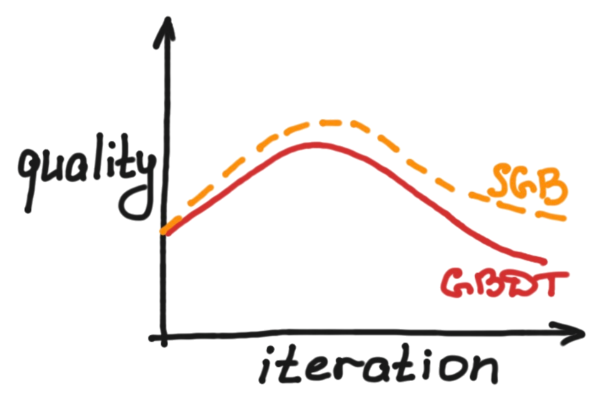


1

Stochastic Gradient Boosting (SGB)

Speed-up & Regularization

Idea: use subsample at each iteration



Classic approach: uniform sampling (SGB)

Advanced approach: non-uniform sampling (e.g. GOSS)

: it works

: understudied, choice is heuristic

2

Problem formulation

Unknown split score:

$$S = \sum_{l \in L} \frac{\sum_{i\in l} g_i^2}{\sum_{i \in l} h_i}, \textit{where } L - \textit{set of leaves}$$

Perform random sampling:

$$\xi_1, \dots, \xi_n \ - \textit{independent s.t. } \xi_i \sim Bern(p_i)$$

$$\xi_i = 1 \textit{ iff } i_{th} \textit{ object is sampled}$$

Construct an estimator: $\hat S = \sum_{l \in L} \frac{\sum_{i \in l} \frac{1}{p_i} \xi_i g_i^2}{\sum_{i\in l} \frac{1}{p_i}\xi_i h_i}$

Such that: $E\Delta^2 = E\left(\hat S - S\right)^2 \to \min_{p_1,\dots,p_n}$

3

Main result

$$E\Delta^2 \approx \sum_{l \in L} c_l^2 \left(4Var(x_l) - 4c_l Cov(x_l, y_l) + c_l^2 Var(y_l)\right)$$

$$E\Delta^2 \lesssim 2 \sum_{l \in L} c_l^2 \left(4Var(x_l) + c_l^2 Var(y_l)\right),$$

$$\textit{where } x_l = \sum_{i \in l} \frac{1}{p_i}\xi_i g_i,\ y_l = \sum_{i\in l}\frac{1}{p_i}\xi_i h_i \textit{ and } c_l = \frac{Ex_l}{Ey_l}$$

Upper bound minimization problem:

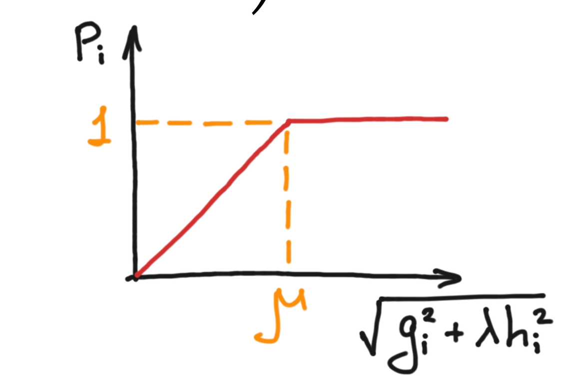
$$\sum_{i=1}^{n} \frac{1}{p_i} g_i^2 + \lambda \sum_{i=1}^{n} \frac{1}{p_i} h_i^2 \to \min \ w.r.t. \sum p_i = ns,$$

$$\textit{where } s - \textit{sampling ratio}, \lambda - \textit{hyperparameter}$$

4

Minimal Variance Sampling (MVS)

$p_i = min\left(\frac{1}{\mu}\sqrt{g_i^2 + \lambda h_i^2}, 1\right)$, μ *is set such that* $\sum p_i = ns$



Algorithm:

1. *Calculate* $\{\hat g_i\}_{i=1}^n = \left\{\sqrt{g_i^2 + \lambda h_i^2}\right\}_{i=1}^n$
2. *Calculate threshold* μ
3. *Sample object* x_i *with probability* p_i
4. *Assign weight* $w_i = \frac{1}{p_i}$

5

Experiments

6