# SGLB: Stochastic Gradient Langevin Boosting
# Supplementary Materials

**Aleksei Ustimenko  Liudmila Prokhorenkova**

## A. Proof of Lemma 1

First, let us prove that $\Phi_{s_\tau} = (H_{s_\tau}^T H_{s_\tau})^\dagger H_{s_\tau}^T$.

We can rewrite Equation (3) from the main text as

$$\theta_*^{s_\tau} = \lim_{\delta \to 0} \arg\min_{\theta^{s_\tau}} \| - \epsilon \widehat{g}_\tau - H_{s_\tau} \theta^{s_\tau} \|_2^2 + \delta^2 \|\theta^{s_\tau}\|_2^2 \,.$$

Taking the derivative of the inner expression, we obtain:

$$\left( H_{s_\tau}^T H_{s_\tau} + \delta^2 I_N \right) \theta^{s_\tau} - \epsilon H_{s_\tau}^T \widehat{g}_\tau = 0 \,.$$

So, $\Phi_{s_\tau}$ can be defined as $\lim_{\delta \to 0} (H_{s_\tau}^T H_{s_\tau} + \delta^2 I_N)^{-1} H_{s_\tau}^T$. Such limit is well defined and is known as the pseudo-inverse of the matrix (Gulliksson et al., 2000).

Let us now prove Lemma 1 from the main text.

The matrix $P_{s_\tau}$ is symmetric since $P_{s_\tau} = \lim_{\delta \to 0} H_{s_\tau} (H_{s_\tau}^T H_{s_\tau} + \delta^2 I_N)^{-1} H_{s_\tau}^T$.

Observe that if $H_{s_\tau} \theta^{s_\tau} = v$, then $P_{s_\tau} v = v$, since the problem in Equation (3) of the main text has an exact solution for the $\arg\min$ subproblem. As a result, $\mathrm{im} P_{s_\tau} = \mathrm{im} H_{s_\tau}$. Also, for an arbitrary $v \in \mathbb{R}^N$, we have $P_{s_\tau}(P_{s_\tau} v) = P_{s_\tau} v$ since $P_{s_\tau} v \in \mathrm{im} H_{s_\tau}$.

## B. CatBoost Implementation

We implemented SGLB as a part of the CatBoost gradient boosting library, which was shown to provide state-of-the-art results on many datasets (Prokhorenkova et al., 2018). Now we specify the particular tuple $\mathcal{B} = (\mathcal{H}, p(s|g))$ such that all the required assumption are satisfied. Therefore, the implementation must converge globally for a wide range of functions, not only for convex ones.

Let us describe the weak learners set $\mathcal{H}$ used by CatBoost. For each numerical feature, CatBoost chooses between a finite number of splits $\mathbb{1}_{\{x_i \le c_{ij}\}}$, where $\{c_{ij}\}_{j=1}^{d_i}$ are some constants typically picked as quantiles of $x_i$ estimated on $\mathcal{D}_N$ and $d_i$ is bounded by a hyperparameter *border-count*. So, the set of weak learners $\mathcal{H}$ consists of all non-trivial binary oblivious trees with splits $\mathbb{1}_{\{x_i \le c_{ij}\}}$ and with depth bounded by a hyperparameter *depth*. This set is finite, $|S| < \infty$. We take $\theta^s \in \mathbb{R}^{m_s}$ as a vector of leaf values of the obtained tree.

Now we are going to describe $p(s|g)$. Assume that we are given a vector $g \in \mathbb{R}^N$ and already built a tree up to a depth $j$ with remaining (not used) binary candidate splits $b_1, \dots b_p$. Each split, being added to the currently built tree, divides the vector $g$ into components $g_1 \in \mathbb{R}^{N_1}, \dots, g_k \in \mathbb{R}^{N_k}$, where $k = 2^{j+1}$. To decide which split $b_l$ to apply, CatBoost calculates the following statistics:

$$s_l := \sqrt{\sum_{i=1}^{k} \mathrm{Var}(g_i)},$$

where $\mathrm{Var}(\cdot)$ is the variance of components from the component-wise mean. Denote also $\sigma := \sqrt{\mathrm{Var}(g)}$. Then, CatBoost evaluates:

$$s_l' := \mathcal{N}\left( s_l, \left( \frac{\rho\sigma}{1 + N^{\epsilon\tau}} \right)^2 \right),$$

where $\rho \ge 0$ is a hyperparameter defined by the *random-strength* parameter. After obtaining $s_l'$, CatBoost selects the split with a highest $s_l'$ value and adds it to the tree. Then, it proceeds recursively until a stopping criteria is met.

Since $\epsilon\tau \to \infty$, we can assume that the variance of $s_l'$ equals zero in the limit. Thus, the stationarity of sampling is preserved. So, $p(s|g)$ is fully specified, and one can show that it satisfies all the requirements. Henceforth, such CatBoost implementation $\mathcal{B}$ must converge globally for a large class of losses as $\epsilon \to 0_+, \epsilon\tau \to \infty$.

## C. Experimental Setup

### C.1. Dataset Description

The datasets are listed in Table 1.

### C.2. Parameter Tuning

For all algorithms, we use the default value 64 for the parameter *border-count* and the default value 0 for *random-strength* ($\rho \ge 0$).

For SGB, we tune *learning-rate* ($\epsilon > 0$), *depth* (the maximal tree depth), and the regularization parameter *l2-leaf-reg*. Moreover, we set *bootstrap-type=Bernoulli*.

*Table 1.* Datasets description

| Dataset | # Examples | # Features |
|---|---|---|
| Appetency (KDD, 2009) | 50000 | 231 |
| Churn (KDD, 2009) | 50000 | 231 |
| Upselling (KDD, 2009) | 50000 | 231 |
| Adult (Kohavi and Becker, 1996) | 48842 | 15 |
| Amazon (Kaggle, 2017) | 32769 | 9 |
| Click (KDD, 2012) | 399482 | 12 |
| Epsilon (PASCAL Challenge, 2008) | 500K | 2000 |
| Higgs (Whiteson, 2014) | 11M | 28 |
| Internet (KDD, 1998) | 10108 | 69 |
| Kick (Kaggle, 1998) | 72983 | 36 |

For SGLB, we tune *learning-rate*, *depth*, *model-shrink-rate* ($\gamma \geq 0$), and *diffusion-temperature* ($\beta > 0$).

For all methods, we set *leaf-estimation-method=Gradient* as our main purpose is to compare first order optimization, and use the option *use-best-model=True*.

For tuning, we use the random search (200 samples) with the following distributions:

- For *learning-rate* log-uniform distribution over $[10^{-5}, 1]$.
- For *l2-leaf-reg* log-uniform distribution over $[10^{-1}, 10^{1}]$ for SGB and *l2-leaf-reg=0* for SGLB.
- For *depth* uniform distribution over $\{6, 7, 8, 9, 10\}$.
- For *subsample* uniform distribution over $[0, 1]$.
- For *model-shrink-rate* log-uniform distribution over $[10^{-5}, 10^{-2}]$ for SGLB.
- For *diffusion-temperature* log-uniform distribution over $[10^{2}, 10^{5}]$ for SGLB.

## References


M. E. Gulliksson, P.-Å. Wedin, and Yimin Wei. 2000. Perturbation Identities for Regularized Tikhonov Inverses and Weighted Pseudoinverses. *BIT Numerical Mathematics* 40, 3 (2000), 513–523.

Kaggle. 1998. Don't Get Kicked! https://www.kaggle.com/c/DontGetKicked. (1998).

Kaggle. 2017. Amazon dataset. https://www.kaggle.com/bittlingmayer/amazonreviews. (2017).

KDD. 1998. KDD Internet Usage Data. https://www.kdd.org/kdd-cup/view/kdd-cup-2012-track-2. (1998).

KDD. 2009. KDD Cup 2009: Customer relationship prediction. https://www.kdd.org/kdd-cup/view/kdd-cup-2009/Data. (2009).

KDD. 2012. KDD Cup 2012 (Track 2): Predict the click-through rate of ads given the query and user information. https://www.kdd.org/kdd-cup/view/kdd-cup-2012-track-2. (2012).

Ronny Kohavi and Barry Becker. 1996. Adult dataset. https://archive.ics.uci.edu/ml/datasets/Adult. (1996).

PASCAL Challenge. 2008. Epsilon dataset. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#epsilon. (2008).

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*. 6638–6648.

Daniel Whiteson. 2014. Higgs dataset. https://archive.ics.uci.edu/ml/datasets/HIGGS. (2014).

*Table 2.* Notation used throughout the paper

| Variable | Description |
|---|---|
| $x \in \mathcal{X}$ | Features, typically from $\mathbb{R}^k$ |
| $y \in \mathcal{Y}$ | Target, typically from $\mathbb{R}$ or $\{0,1\}$ |
| $z \in \mathcal{Z}$ | Prediction, typically from $\mathbb{R}$ |
| $\mathcal{D}$ | Data distribution over $\mathcal{X} \times \mathcal{Y}$ |
| $\mathcal{D}_N = \{(x_i, y_i)\}_{i=1}^N$ | I.i.d. samples from $\mathcal{D}$ |
| $L(z,y) : \mathcal{Z} \times \mathcal{Y} \to \mathbb{R}$ | Loss function |
| $\mathcal{L}(f\|\mathcal{D})$ | Expected loss w.r.t. $\mathcal{D}$ |
| $\mathcal{L}_N(f)$ | Empirical loss |
| $\mathcal{L}_N(F, \gamma)$ | Regularized or implicitly regularized loss |
| $\mathcal{H}$ | Set of weak learners |
| $h^s(x, \theta^s) \in \mathcal{H}$ | Weak learner parameterized by $\theta^s$ |
| $H_s : \mathbb{R}^{m_s} \to \mathbb{R}^N$ | Linear operator converting $\theta^s$ to $(h^s(x_i, \theta^s))_{i=1}^N$ |
| $\Theta \in \mathbb{R}^m$ | Ensemble parameters |
| $f_\Theta(x) : \mathcal{X} \to \mathcal{Z}$ | Model parametrized by $\Theta \in \mathbb{R}^m$ |
| $\tau \in \mathbb{Z}_+$ | Discrete time |
| $t \in [0, \infty)$ | Continuous time |
| $\hat{F}_\tau$ | Predictions' Markov Chain $\left(f_{\widehat{\Theta}_\tau}(x_i)\right)_{i=1}^N$ |
| $F(t)$ | Markov process $\left(f_{\Theta(t)}(x_i)\right)_{i=1}^N$ |
| $V_\mathcal{B} \subset \mathbb{R}^N$ | Subspace of predictions of all possible ensembles |
| $p(s\|g)$ | Probability distribution over weak learners' indices |
| $\Phi_s : \mathbb{R}^N \to \mathbb{R}^{m_s}$ | Weak learner parameters estimator |
| $P_s := H_s \Phi_s$ | Orthoprojector |
| $P_\infty = N\mathbb{E}_{s \sim p(s\|\mathbb{0}_N)} P_s$ | Implicit limiting preconditioner matrix of the boosting |
| $P = P_\infty$ | Symmetric preconditioner matrix |
| $\Gamma = \sqrt{P^{-1}}$ | Regularization matrix |
| $\delta_\Gamma(\gamma)$ | Error from the regularization |
| $p_\beta(\Theta)$ | Limiting distribution of $\widehat{\Theta}_\tau$ |
| $\lambda_*$ | Uniform spectral gap parameter |
| $\epsilon > 0$ | Learning rate |
| $\beta > 0$ | Inverse diffusion temperature |
| $\gamma > 0$ | Regularization parameter |
| $I_m \in \mathbb{R}^{m \times m}$ | Identity matrix |
| $\mathbb{0}_m \in \mathbb{R}^m$ | Zero vector |
| $W(t)$ | Standard Wiener process |
| $\phi(x) : \mathcal{X} \to \mathbb{R}^m$ | Feature map, s.t. $f_\Theta(x) = \langle \phi(x), \Theta \rangle_2$ |
| $\Psi := \left[\phi(x_1), \dots, \phi(x_N)\right]^T \in \mathbb{R}^{N \times m}$ | Design matrix |