Gradient Boosting Performs Low-Rank Gaussian Process Inference

Aleksei Ustimenko, Artem Beliakov, Liudmila Prokhorenkova
Yandex Research, Moscow, Russia

Abstract

This paper shows that gradient boosting based on symmetric decision trees can be equivalently reformulated as a kernel method that converges to the solution of a certain Kernel Ridgeless Regression problem. Thus, for low-rank kernels, we obtain the convergence to a Gaussian Process’ posterior mean, which, in turn, allows us to easily transform gradient boosting into a sampler from the posterior to provide better knowledge uncertainty estimates through Monte-Carlo estimation of the posterior variance. We show that the proposed sampler allows for better knowledge uncertainty estimates leading to improved out-of-domain detection.

1 Introduction

Gradient boosting [3] is a classic machine learning algorithm successfully used for web search, recommendation systems, weather forecasting, and other problems [19, 2, 18, 21, 1, 22]. In a nutshell, gradient boosting methods iteratively combine simple models (usually decision trees), minimizing a given loss function. Despite recent success of neural approaches in various areas, gradient-boosted decision trees (GBDT) are still state-of-the-art algorithms for tabular datasets containing heterogeneous features [6, 9].

This paper aims at better theoretical understanding of GBDT methods for regression problems assuming the widely used RMSE loss function. Importantly, we consider a classic gradient boosting procedure (Algorithm 2) without any simplifications. Namely, we only need decision trees to be symmetric (oblivious) and properly randomized (Algorithm 1). These assumptions are non-restrictive and are satisfied in some popular gradient boosting implementations, e.g., CatBoost [16].

We derive a non-linear integro-differential equation in reproducing kernel Hilbert space (RKHS) which describes the evolution of gradient boosting iterations similarly to the gradient flow description of (stochastic) gradient descent. It turns out that under mild conditions, it is possible to asymptotically approximate the equation’s solution with gradient flow equation solution, which is a linear differential equation and can be solved analytically. As a result, the dynamics of gradient boosting learning can be studied by using continuous methods despite the discrete nature of the boosting.

Based on our analysis, we reformulate GBDT as a Kernel Method that converges to the solution of a certain Kernel Ridgeless Regression problem. For low-rank kernels, we obtain the convergence to a Gaussian process’ posterior mean, which, in turn, allows us to easily transform gradient boosting into a sampler from the posterior to provide better knowledge uncertainty estimates through Monte-Carlo estimation of the posterior variance. Our experiments confirm that the proposed sampler outperforms the previous approaches [14] and gives better knowledge uncertainty estimates and improved out-of-domain detection.

Preprint. Under review.
2 Background

Assume that we are given a distribution \( \rho \) over \( X \times Y \), where \( X \subset \mathbb{R}^d \) is called a feature space and \( Y \subset \mathbb{R} \) — a target space. Further assume that we are given a dataset \( \mathbf{z} = \{(x_i, y_i)\}_{i=1}^N \subset X \times Y \) of size \( N \geq 1 \) sampled i.i.d. from \( \rho \). Let us denote by \( \rho(dx) = \int_Y d\rho(dx, dy) \). W.l.o.g., we also assume that \( X = \text{supp} \rho = \text{clos} \{ x \in \mathbb{R}^d : \forall \epsilon > 0, \rho(\{ x' \in \mathbb{R}^d : \| x - x' \|_{\mathbb{R}^d} < \epsilon \}) > 0 \} \) which is a closed subset of \( \mathbb{R}^d \). Moreover, for technical reasons we assume that \( \frac{1}{N} \sum_{i=1}^N y_i^2 \leq R^2 \) for some constant \( R > 0 \) almost surely, which can always be enforced by clipping. Throughout the paper, we also denote by \( \mathbf{x}_N \) and \( \mathbf{y}_N \) the matrix of all feature vectors and the vector of targets.

2.1 Gradient boosted decision trees

Given a loss function \( L : \mathbb{R}^2 \to \mathbb{R} \), a classic gradient boosting algorithm \[ \text{GBDT} \] iteratively combines weak learners (usually decision trees) to reduce the average loss over the train set \( \mathbf{z} : \mathbb{L}(f) = \mathbb{E}_\mathbf{z}[L(f(x), y)] \). At each iteration \( \tau \), the model is updated as: \( f_\tau(x) = f_{\tau-1}(x) + \epsilon w_\tau(x) \), where \( w_\tau(\cdot) \in \mathcal{W} \) is a weak learner chosen from some family of functions \( \mathcal{W} \), and \( \epsilon \) is a learning rate. The weak learner \( w_\tau \) is usually chosen to approximate the negative gradient of the loss function \( -g_\tau(x, y) := \frac{\partial L(s, y)}{\partial s} |_{s = f_{\tau-1}(x)} \).

\[
 w_\tau = \arg\min_{w \in \mathcal{W}} \mathbb{E}_\mathbf{z} \left[ \left( -g_\tau(x, y) - w(x) \right)^2 \right].
\] (1)

The family \( \mathcal{W} \) usually consists of decision trees. In this case, the algorithm is called GBDT (Gradient Boosted Decision Trees). A decision tree is a model that recursively partitions the feature space into disjoint regions called leaves. Each leaf \( R_j \) of the tree is assigned to a value, which is the estimated response \( y \) in the corresponding region. Thus, we can write \( w(x) = \sum_{j=1}^d \theta_j 1_{\{x \in R_j\}} \), so the decision tree is a linear function of the leaf values \( \theta_j \).

A recent paper \[ \text{SGB} \] proposes a modification of classic stochastic gradient boosting (SGB) called Stochastic Gradient Langevin Boosting (SGLB). SGLB combines gradient boosting with stochastic gradient Langevin dynamics to achieve global convergence even for non-convex loss functions. The algorithm has two modifications compared to the classic boosting. First, Gaussian noise (parametrized by a temperature parameter) is injected into the gradients, and the temperature regulates the level of exploration. Second, shrinkage is added to the boosting process: at each iteration, the currently built model is multiplied by a constant smaller than one. As a result, the obtained algorithm provably converges to some stationary distribution (invariant measure) concentrated near the global optimum of the loss function.

2.2 Estimating uncertainty

In addition to the predictive quality, it is often important to detect when the system is uncertain and can be mistaken. For this, different measures of uncertainty can be used. There are two main sources of uncertainty: data uncertainty (a.k.a. aleatoric uncertainty) and knowledge uncertainty (a.k.a. epistemic uncertainty). Data uncertainty arises due to the inherent complexity of the data, such as additive noise or overlapping classes. For instance, if the target is distributed as \( y|x \sim \mathcal{N}(f(x), \sigma^2) \), then \( \sigma \) reflects the level of data uncertainty. This uncertainty can be assessed if the model is probabilistic.

Knowledge uncertainty arises when the model gets input from a region either sparsely covered by or far from the training data. Since the model does not have enough data in this region, it is likely to make a mistake. A standard approach to estimating knowledge uncertainty is based on ensembles \[ \text{ensembles} \] \[ \text{ensembles} \] \[ \text{ensembles} \]. Assume that we have trained an ensemble of several independent models. If all the models understand an input (low knowledge uncertainty), they will give similar predictions. However, for out-of-domain examples (high knowledge uncertainty), the models are likely to provide diverse predictions. For regression tasks, one can obtain knowledge uncertainty by measuring the variance of the predictions provided by multiple models \[ \text{ensembles} \].

Such ensemble-based approaches are standard for neural networks \[ \text{ensembles} \]. Recently, ensembles were also tested for GBDT models \[ \text{ensembles} \]. The authors consider two ways of generating ensembles: ensembles of independent SGB models and ensembles of independent SGLB models. While empirically...
To estimate the posterior mean of the target function, we use the following iterative scheme:

1. sample \( f^{\text{init}} \sim \mathcal{GP}(0_{L_2(\rho)}^{\sigma^2\mathcal{K}}) \);
2. set new labels \( y_N^{\text{new}} = y_N - f^{\text{init}}(x_N) \);
3. fit GD \( F_n(\cdot) \) on \( y_N^{\text{new}} \) assuming \( F_0(\cdot) = 0_{L_2(\rho)}^{\sigma^2\mathcal{K}} \);
4. output \( f^{\text{init}}(\cdot) + F_n(\cdot) \) as final model.

This method is known as Sample-then-Optimize [15] and is widely adopted for Bayesian inference. As \( n \to \infty \), we get \( f^{\text{init}} + F_\infty \) distributed as a Gaussian Process posterior with the desired mean and variance. Formally, the following result holds:

\[ \mathcal{K}(x, x') \] is a kernel function if \( \mathcal{K}(x_i, x_j) \) is available for any \( N \geq 1 \) and any \( x_i, x_j \in X \) almost surely.

\[ A^{\dagger} \] denotes the pseudo-inverse of a matrix \( A \).
Lemma 2.1. \( f^{\text{init}} + F_\infty \) follows the Gaussian Process posterior \( \mathcal{GP}(f_*, \mathcal{K}) \) with:
\[
    f_*(x) = \mathcal{K}(x, x_N)\mathcal{K}(x_N, x_N)^\dagger y_N,
\]
\[
    \mathcal{K}(x, x) = \sigma^2(\mathcal{K}(x, x) - \mathcal{K}(x, x_N)\mathcal{K}(x_N, x_N)^\dagger \mathcal{K}(x_N, x)).
\]

3 Evolution of GBDT in RKHS

3.1 Preliminaries

In our analysis, we assume that we are given a finite set \( \mathcal{V} \) of weak learners used for the gradient boosting\(^3\). Each \( \nu \) corresponds to a decision tree that defines a partition of the feature space into disjoint regions (leaves). For each \( \nu \in \mathcal{V} \), we denote the number of leaves in the tree by \( L_\nu \geq 1 \). Also, let \( \phi_\nu : X \to \{0, 1\}^{L_\nu} \) be a mapping that maps \( x \) to the vector indicating its leaf index in the tree \( \nu \). This mapping defines a decomposition of \( X \) into the disjoint union: \( X = \bigcup_{\nu=1}^{L_\nu} \{ x \in X | \phi_\nu(j)(x) = 1 \} \).

Having \( \phi_\nu \), we define a weak learner associated with it as \( x \mapsto \langle \theta, \phi_\nu(x) \rangle_{\mathbb{R}^{L_\nu}} \) for any choice of \( \theta \in \mathbb{R}^{L_\nu} \) which we refer to as ‘leaf values’. In other words, \( \theta \) corresponds to predictions assigned to each region of the space defined by \( \nu \).

Let us define a linear space \( \mathcal{F} \subset L_2(\rho) \) of all possible ensembles of trees from \( \mathcal{V} \):
\[
    \mathcal{F} = \text{span} \{ \phi_\nu^{(j)}(\cdot) : X \to \{0, 1\} | \nu \in \mathcal{V}, j \in \{1, \ldots, L_\nu\} \}.
\]

We note that the space \( \mathcal{F} \) can be data-dependent since \( \mathcal{V} \) may depend on \( z \), but we omit this dependence in the notation for simplicity. Note that we do not take the closure w.r.t. the topology of \( L_2(\rho) \) since we assume that \( \mathcal{V} \) is finite and therefore \( \mathcal{F} \) is finite-dimensional and thus topologically closed.

Assumption 3.1. We assume the low-rankness of \( \mathcal{F} \), i.e., that \( \text{dim}(\mathcal{F}) \ll N \).

This condition naturally holds if we fix constant depth \( m \) and the number of bins \( n \) of the trees (see Section 3.2). In future work, this assumption can be alleviated using shrinkage as proposed by \(^2\). An immediate consequence of Assumption 3.1 is that any ensemble \( f \in \mathcal{F} \) is determined by the values \( f(x_1), \ldots, f(x_N) \) with high probability. Thus, we further assume this to hold.

Now, let us define the empirical error of a model \( f \) relative to the best possible \( f' \in \mathcal{F} \):
\[
    V(f) = \frac{1}{2N} \| y_N - f(x_N) \|^2_{\mathbb{R}^N} - \inf_{f' \in \mathcal{F}} \left( \frac{1}{2N} \| y_N - f'(x_N) \|^2_{\mathbb{R}^N} \right).
\]

Let us also define the following unique function:
\[
    f_* = \lim_{\lambda \to 0^+} \arg \min_{f \in \mathcal{F}} \left( V(f) + \lambda \| f \|^2_{L_2(\rho)} \right) \in \arg \min_{f \in \mathcal{F}} V(f).
\]

It is known that such \( f_* \) exists and is unique since the set of all solutions is convex, and therefore there is a unique minimizer of the norm\(^4\).

Finally, the following lemmas will be helpful for our analysis: in particular, to analyze the distribution of trees in Section 3.3 and to derive the formula of the GBDT iteration in terms of kernels in lemma 3.12. See Appendix C for the proofs.

Lemma 3.2. \( \langle y_N - f_*(x_N), f(x_N) \rangle_{\mathbb{R}^N} = 0 \) for any \( f \in \mathcal{F} \).

Lemma 3.3. \( V(f) = \frac{1}{2N} \| f_*(x_N) - f(x_N) \|^2_{\mathbb{R}^N} \).

3.2 Weak learner selection algorithm

Our theoretical analysis holds for classic GBDT algorithms discussed in Section 2.1. The only requirement we need is that a procedure of choosing each new tree has to be properly randomized. Let us discuss a tree selection algorithm that we assume in our analysis.

\(^3\)The finiteness of \( \mathcal{V} \) is important for our analysis, and it usually holds in practice, see Section 3.2.

\(^4\)Note that in a general setting, we cannot define \( f_* \) without equipping \( \mathcal{F} \) with some norm since \( \mathcal{F} \) may be incomplete in \( L_2 \) topology. But in our setting, we rely on finite-dimensionality of \( \mathcal{F} \), which implies that \( L_2 \) and any other norm is equivalent on \( \mathcal{F} \) and complete.
Each new tree approximates the gradients of the loss function with respect to the current predictions of the model. Since we consider the RMSE loss function, the gradients are proportional to the residuals \( r_j = y_j - f(x_j) \), where \( f \) is the currently built model. The tree structure is defined by the features and the corresponding thresholds used to split the space.

### Algorithm 1: SampleTree\((r; m, n, \beta)\)

**input:** residuals \( r = (r_i)_{i=1}^N \),

**output:** oblivious tree structure \( \nu \in \mathcal{V} \)

**hyper-parameters:** number of feature splits \( n \),

maximum tree depth \( m \),

random strength \( \beta \in [0, \infty) \)

**definitions:**

\[ \mathcal{S} = \{ (j, k) | j \in \{1, \ldots, d\}, k \in \{1, \ldots, n\} \} \]

---

**instructions:**

initialize \( i = 0, \nu_0 = \emptyset, \mathcal{S}^{(0)} = \mathcal{S} \)

repeat

sample \((u_i(s))_{s \in \mathcal{S}^{(i)}} \sim U([0, 1]^{d-i})\)

choose next split as \( \nu_{i+1} = \arg\max_{s \in \mathcal{S}^{(i)}} D((\nu_i, s), r) - \beta \log(-\log u_i(s)) \)

update tree: \( \nu_{i+1} = (\nu_i, s_{i+1}) \)

update candidate splits: \( \mathcal{S}^{(i+1)} = \mathcal{S}^{(i)}\backslash \{s_{i+1}\} \)

\( i = i + 1 \)

until \( i \geq m \) or \( \mathcal{S}^{(i)} = \emptyset \)

return: \( \nu_i \)

### Algorithm 2: TrainGBDT\((z; \epsilon, T, m, n, \beta)\)

**input:** dataset \( z = (x_N, y_N) \)

**hyper-parameters:** learning rate \( \epsilon > 0 \),

iterations of boosting \( T \),

parameters of SampleTree \( m, n, \beta \).

**instructions:**

\( \tau = 0 \)

\( f_0(\cdot) = \Phi_L(\rho) \)

repeat

\( r_\tau = y_N - f_\tau(x_N) \) — compute residuals

\( \nu_\tau = \text{SampleTree}(r_\tau; m, n, \beta) \) — construct a tree

\( \theta_\tau = \left( \sum_{i=1}^N \phi_{\nu_\tau}^I(x_i) r_i \right)_{j=1} \) — set values in leaves

\( f_{\tau+1}(\cdot) = f_\tau(\cdot) + \epsilon \left( \phi_{\nu_\tau}(\cdot), \theta_\tau \right)_{\Phi_L} \) — update model

\( \tau = \tau + 1 \)

until \( \tau \geq T \)

return: \( f_T(\cdot) \)

The analysis in this paper is based on the SampleTree procedure described in Algorithm 1 which is a classic approach equipped with proper randomization. SampleTree builds an oblivious decision tree [14], i.e., all nodes at a given level share the same splitting criterion (feature and threshold). To limit the number of candidate splits, each feature is quantized into \( n + 1 \) bins. In other words, for each feature, we have \( n \) thresholds that can be chosen arbitrarily. The maximum depth of the tree is limited by \( m \). Recall that we denote the set of all possible tree structures by \( \mathcal{V} \).

We build the tree in a top-down greedy manner. At each step, we choose one split among all remaining candidates based on the following score defined for \( \nu \in \mathcal{V} \) and residuals \( r \):

\[
D(\nu, r) := \frac{1}{N} \sum_{j=1}^L \left( \frac{\sum_{i=1}^N \phi_\nu^I(x_i) r_i}{\sum_{i=1}^N \phi_\nu^I(x_i)} \right)^2 .
\]

(3)

In classic gradient boosting, one builds a tree recursively by choosing such split \( s \) that maximizes the score \( \hat{D}((\nu_i, s), r) \) \( \square \). Random noise is often added to the scores to improve generalization. In SampleTree, we choose a split that maximizes

\[
D((\nu_i, s), r) + \varepsilon, \text{ where } \varepsilon \sim \text{Gumbel}(0, \beta) .
\]

(4)

Here \( \beta \) is random strength: \( \beta = 0 \) gives the standard greedy approach, while \( \beta \to \infty \) gives the uniform distribution among all possible split candidates.

To sum up, SampleTree is a classic oblivious tree construction but with added random noise. We do this to make the distribution of trees regular in a certain sense: roughly speaking, the distributions should stabilize with iterations by converging to some fixed distribution. Given the algorithm SampleTree, our analysis assumes the classic gradient boosting procedure; see Algorithm 2.

\( \square \)A standard approach is to quantize the feature such that all \( n + 1 \) buckets have approximately the same number of training samples.

\( \square \)Maximizing \( \square \) is equivalent to minimizing the squared error between the residuals and the mean values in the leaves.
3.3 Distribution of trees

The SampleTree algorithm induces a local family of distributions \( p(\cdot|f, \beta) \) for each \( f \in \mathcal{F} \):

\[
p(d\nu|f, \beta) = \mathbb{P} \left( \text{SampleTree}(y_N - f(x_N); m, n, \beta) \in d\nu \right).
\]

Remark 3.4. Lemmas 3.2 and 3.3 ensure that such distribution coincides with the one where we use \( f_*(x_N) \) instead of \( y_N \).

The following lemma describes the distribution \( p(d\nu|f, \beta) \), see Appendix 3 for the proof. Note that for oblivious decision trees, changing the order of splits does not affect the obtained partition. Hence, we assume that each tree is defined by an unordered set of splits.

**Lemma 3.5. (Probability of a tree)**

\[
p(\nu|f, \beta) = \sum_{\varsigma \in \mathcal{P}_m} \prod_{i=1}^{m} \frac{\nu(s_{\varsigma(i-1)})}{\nu(s_{\varsigma(i)})},
\]

where the sum is over all permutations \( \varsigma \in \mathcal{P}_m \), \( \nu_{\varsigma, i-1} = (s_{\varsigma(1)}, \ldots, s_{\varsigma(i-1)}) \), and \( \nu = (s_1, \ldots, s_m) \).

Let us define the stationary distribution of trees as \( \pi(\cdot) = \lim_{\beta \to \infty} p(\cdot|f, \beta) \). It follows from Remark 3.4 that we also have \( \pi(\cdot) = p(\cdot|f_*, \beta) \).

**Corollary 3.6. (Stationary distribution is uniform distribution over tree structures)**

We have \( \pi(d\nu) = |d\nu|/(\text{nd}_m), \) where \( \text{nd}_m = \frac{\text{nd}_m}{(\text{nd}_m)!} \).

3.4 RKHS Structure

In this section, we describe the evolution of GBDT in a certain Reproducing Kernel Hilbert Space (RKHS). Let us start with defining necessary kernels.

**Definition 3.7.** Weak learner’s kernel \( k_{\nu}(\cdot, \cdot) \) is a kernel function associated with a tree structure \( \nu \in \mathcal{V} \) which can be defined as:

\[
k_{\nu}(x, x') = \sum_{j=1}^{L_{\nu}} w^{(j)}_{\nu}(x) \phi^{(j)}_{\nu}(x'), \text{ where } w^{(j)}_{\nu} = \frac{N_{j}}{\max\{N_{ij}, 1\}}, \quad N_{j} = \sum_{i=1}^{N} \phi^{(j)}_{\nu}(x_i).
\]

**Definition 3.8.** We also define a greedy kernel of the gradient boosting algorithm as follows:

\[
\mathcal{K}_f(x, x') = \int_{\mathcal{V}} k_{\nu}(x, x') p(d\nu|f, \beta).
\]

**Definition 3.9.** Finally, there is a stationary kernel \( \mathcal{K}(x, x') \) that is independent from \( f \):

\[
\mathcal{K}(x, x') = \int_{\mathcal{V}} k_{\nu}(x, x') \pi(d\nu),
\]

which we call a prior kernel of the gradient boosting.

Having the space of functions \( \mathcal{F} \), we define a reproducing kernel Hilbert space (RKHS) structure \( \mathcal{H} = (\mathcal{F}, \langle \cdot, \cdot \rangle_{\mathcal{H}}) \) on it using a scalar product defined as

\[
\langle f, f \rangle_{\mathcal{H}} = \min_{g \in L_2(\rho)} \int_{\mathcal{X}} g^2(x)\rho(dx) \text{ s.t. } \mathcal{K}^{\frac{1}{2}} g = f,
\]

\[
\langle f, g \rangle_{\mathcal{H}} = \frac{1}{4} \left( \langle f + g, f + g \rangle_{\mathcal{H}} - \langle f - g, f - g \rangle_{\mathcal{H}} \right),
\]

where \( \mathcal{K}^{\frac{1}{2}} \) corresponds to the unique symmetric positive definite (SPD) square root operator such that \( \left( \mathcal{K}^{\frac{1}{2}} \right)^2 = \mathcal{K} \) assuming that it acts as \( \mathcal{K}[g] = \int_{\mathcal{X}} \mathcal{K}(\cdot, x) g(x)\rho(dx) \in L_2(\rho) \) for any \( g \in L_2(\rho) \).

For a weak learner \( \nu \), we define a covariance operator:

\[
\Sigma_{\nu}[f] = \frac{1}{N} k_{\nu}(\cdot, x_N) f(x_N), \quad \Sigma_{\nu} : \mathcal{H} \to \mathcal{H}.
\]
Using this operator, we define $\Sigma_f = \int_Y \Sigma_\nu p(\nu|f, \beta)$ and $\Sigma = \int_Y \Sigma_\nu \pi(\nu)$. These operators are typically referred to as covariance operators.

Let us formulate several statements that will be useful for proving the main results of our paper.

**Lemma 3.10.** The following equality holds for any $\pi$:

$$f_* (\cdot) \equiv K(\cdot, \mathbf{x}_N) \left\{ K(\mathbf{x}_N, \mathbf{x}_N) \right\}^{\dagger} \mathbf{y}_N.$$

**Proof.** Since the operator $K$ is defined as a bounded symmetric integral operator in $L_2(\rho)$ with image $\mathcal{F}$ and $f_*$ is defined as $L_2(\rho)$-min norm solution, then it holds that $L_2(\rho) = \mathcal{F} \oplus \mathcal{F}^\perp$, where $\mathcal{F}^\perp$ is $L_2(\rho)$-orthogonal complement such that $\mathcal{F}^\perp = \ker (K) \forall \pi$, which implies that it is not only $L_2(\rho)$-min norm solution but also min-norm in RKHS $\mathcal{H}$. Then the lemma follows from Lemma C.4.

**Lemma 3.11.** For any $\nu \in \mathcal{V}$, we have $k_\nu (\cdot, \mathbf{x}_N) [\mathbf{y}_N - f_*(\mathbf{x}_N)] = 0$.

**Proof.** Follows from Lemma 3.2.

**Lemma 3.12.** Iterations $f_\tau$ of Gradient Boosting (Algorithm 2) can be written in the form:

$$f_{\tau+1} = f_\tau + \epsilon \frac{1}{N} k_{\nu_\tau} (\cdot, \mathbf{x}_N) [\mathbf{y}_N - f_\tau (\mathbf{x}_N)] = f_\tau + \epsilon \frac{1}{N} k_{\nu_\tau} (\cdot, \mathbf{x}_N) [f_* (\mathbf{x}_N) - f_\tau (\mathbf{x}_N)],$$

$$\nu_\tau \sim p(\nu|f_\tau, \beta).$$

**Proof.** According to Algorithm 2,

$$f_{\tau+1} (\cdot) = f_\tau (\cdot) + \epsilon \langle \phi_{\nu_\tau} (\cdot), \theta_\tau \rangle_{\mathcal{F}^{\perp} \mathcal{V}_{\nu_\tau}}.$$ for $\theta_\tau = \left( \frac{\sum_{i=1}^N \phi_{\nu_\tau}^j (x_i)^T \nu_\tau}{\sum_{i=1}^N \phi_{\nu_\tau}^j (x_i)} \right)_{j=1}^{L_{\nu_\tau}}$.

Thus,

$$f_{\tau+1} = f_\tau + \epsilon \frac{1}{N} \sum_{j=1}^{L_{\nu_\tau}} \omega_j \phi_j \sum_{i: \phi_j (x_i) = 1} \nu_\tau.$$

Now note that $k_{\nu_\tau} (\cdot, x_i) = \omega_j \phi_j (\cdot)$, where $j$ is such that $\phi_j (x_i) = 1$. From this the lemma follows.

### 4 Kernel Gradient Boosting

#### 4.1 Kernel Ridgeless Regression

Consider the sequence $\{ f_\tau \}_{\tau \in \mathbb{N}}$ generated by the gradient boosting algorithm. According to Lemma 3.12 we can write the boosting iterations as:

$$f_{\tau+1} = f_\tau + \epsilon \frac{1}{N} k_{\nu_\tau} (\cdot, \mathbf{x}_N) [f_* (\mathbf{x}_N) - f_\tau (\mathbf{x}_N)].$$

This can be rewritten as

$$f_{\tau+1} = f_\tau + \epsilon \Sigma_{\nu_\tau} [f_* - f_\tau], \quad \nu_\tau \sim p(\nu|f_\tau, \beta).$$

Using the covariance $\Sigma_{f_*}$, we derive that:

$$\mathbb{E} \left[ f_{\tau+1} - f_\tau \left| f_\tau \right. \right] = \epsilon \Sigma_{f_*} [f_* - f_\tau].$$

The following theorem estimates the expected (w.r.t. the randomness of tree selection) empirical error of $f_T$ relative to the best possible ensemble. The proof can be found in Appendix F.

**Theorem 4.1.** Let

$$\lambda_* := \inf_{f \in \mathcal{F}_0 \cap V (f) \leq V (f_0)} \lambda_{\min} (\Sigma_f) > 0$$

and consider an arbitrary $0 < \epsilon \leq \lambda_*$. The following inequality holds:

$$\mathbb{E} V (f_T) \leq \frac{1}{2} R^2 e^{-\lambda_* \epsilon T}.$$

7
We first note that the process $h_T(\cdot)$ is centered with covariance operator $K$:
\[
\mathbb{E} h_T(x) = 0 \quad \forall x \in X, \\
\mathbb{E} h_T(x) h_T(y) = K(x, y) \quad \forall x, y \in X. 
\]

Then, we show that $h_T(\cdot)$ converges to the Gaussian Process in the limit.

**Lemma 4.3.** The following convergence holds almost surely in $x \in X$:
\[
h_T(\cdot) \xrightarrow{T \to \infty} GP(0_{L_2(\rho)}, K). 
\]

**Proof.** From (5), we have that the covariance of $h_T$ is $K$ independently from $T$. Thus, it remains to show that the limit is Gaussian almost surely which essentially holds due to the central limit theorem almost surely in $x \in X$:
\[
h_T(x) = \frac{1}{\sqrt{T}} \sum_{t=1}^{T} h_{T,i}(x) \to N(\mathbb{E}h_1(x), K(x, x)),
\]
where each individual tree $h_{T,i}(\cdot)$ is centered i.i.d. (with the same distribution as $h_1$).

\[
\mathbb{E} \left\| f_T - f_\star \right\|_{L_2(\rho)}^2 \leq \frac{e^{-\lambda_\star T R^2}}{\lambda_\star},
\]
which suggests that if $\lambda_\star > O(1) > 0$ and $\epsilon = \lambda_\star$, then for $T \geq \frac{1}{\lambda_\star^2} \log \left( \frac{R^2}{\lambda_\star^2} \right) = O\left( \log \left( \frac{1}{\lambda_\star^2} \right) \right)$ we obtain an error $\delta$, i.e., linear convergence.

### 4.2 Prior Sampling

We introduce Algorithm 3 for sampling from the prior distribution. SamplePrior generates an ensemble of random trees (with random splits and random values in leaves). Note that while being random, the tree structure depends on the dataset features $x_N$ since candidate splits are based on $x_N$.

We first note that the process $h_T(\cdot)$ is centered with covariance operator $K$:
\[
\mathbb{E} h_T(x) = 0 \quad \forall x \in X, \\
\mathbb{E} h_T(x) h_T(y) = K(x, y) \quad \forall x, y \in X. 
\]

Then, we show that $h_T(\cdot)$ converges to the Gaussian Process in the limit.

**Algorithm 3 SamplePrior($T, m, n$)**

**hyper-parameters:** number of iterations $T$, parameters of SampleTree $m, n$

**instructions:**
initialize $\tau = 0$, $h_0(x) = 0$
repeat
\[
\nu = \text{SampleTree}(0_{\mathcal{R}^N}; m, n, 1) \quad \text{— sample random tree}
\]
\[
\theta \sim \mathcal{N}(0_{\mathcal{R}^{L_{\nu}}}, \text{diag}(\frac{N}{\max(N^{2}, 1)} : j \in \{1, \ldots, L_{\nu} \})) \quad \text{— generate random values in leaves}
\]
\[
h_{\tau+1}(\cdot) = h_{\tau}(\cdot) + \sqrt{T} \langle \phi_{\nu}(\cdot), \theta \rangle_{R^{L_{\nu}}} \quad \text{— update model}
\]
until $\tau \geq T$
return: $h_T(\cdot)$

### 4.3 Posterior Sampling

Now we are ready to introduce Algorithm 4 for sampling from the posterior. The procedure is simple: we first perform $T_0$ iterations of SamplePrior to obtain a function $h_{T_0}(\cdot)$ and then we train a standard GBDT model $f_{T_1}(\cdot)$ approximating $y_N - \sigma h_{T_0}(x_N)$. Our final model is $\sigma h_{T_0}(\cdot) + f_{T_1}(\cdot)$. We further refer to this procedure as SamplePosterior or KGB (Kernel Gradient Boosting) for brevity. Denote
\[
h_\infty = \lim_{T_0 \to \infty} h_{T_0} , \\
f_\infty = \lim_{T_1 \to \infty} f_{T_1} ,
\]
where the first limit is with respect to the point-wise convergence of stochastic processes and the second one with respect to $L_2(\rho)$ convergence.
The following theorem shows that KGB indeed samples from the desired posterior. The proof directly follows from Lemmas 4.3 and 2.1.

**Theorem 4.4.** In the limit, the output of Algorithm 4 follows the Gaussian posterior:

\[
\sigma h_\infty(x) + f_\infty(x) \sim GP(\tilde{f}, \tilde{K})
\]

with mean \( \tilde{f}(x) = \mathcal{K}(x, x_N)\mathcal{K}(x_N, x_N)^\dagger y_N \) and covariance \( \tilde{K}(x, x) = \sigma^2(\mathcal{K}(x, x) - \mathcal{K}(x, x_N)\mathcal{K}(x_N, x_N)(\mathcal{K}(x_N, x_N)^\dagger\mathcal{K}(x_N, x)) . \)

### 5 Experiments

This section empirically evaluates the proposed KGB algorithm and shows that it indeed allows for better knowledge uncertainty estimates. Uncertainty estimates for GBDTs have been previously analyzed in [14]. Our experiments closely follow their setup, and we compare the proposed KGB with SGB, SGLB, and their ensembles. For the experiments, we use several standard regression datasets [5]. The implementation details and data description can be found in Appendix G.

It is interesting to compare KGB with SGLB since they both sample from similar posterior distributions. Thus, this comparison allows us to find out which of the algorithms does a better sampling from the posterior and thus provides us with more reliable estimates of knowledge uncertainty. Moreover, we consider the SGB approach as the most “straightforward” way to generate an ensemble of models. Our goal is to show that our approach is superior to both of them not only in achieving smaller values of RMSE but also in knowledge uncertainty estimation.

First, in Table 1, we compare the predictive performance of the methods. Interestingly, we obtain improvements on almost all the datasets. Here we perform cross-validation to estimate statistical significance with paired \( t \)-test and highlight the approaches that are insignificantly different from the best one (p-value > 0.05).

Then, we check whether uncertainty measured as the variance of the model’s predictions can be used to detect errors and out-of-domain inputs. Detecting errors can be evaluated via the Prediction-Rejection Ratio (PRR) [12, 13]. PRR measures how well uncertainty estimates correlate with errors and rank-order them. Out-of-domain (OOD) detection is usually assessed via the area under the ROC curve (AUC-ROC) [7]. For OOD detection, one needs an OOD test set. We use the same test sets as [14]. The results of this experiment are given in Table 2. We can see that the proposed method significantly outperforms the baselines for out-of-domain detection. These improvements can be explained by the theoretical soundness of KGB: convergence properties are theoretically grounded and non-asymptotic. In contrast, for SGB, there are no general results applicable in our setting, while for SGLB the guarantees are asymptotic.

### 6 Conclusion

This paper theoretically analyses the classic gradient boosting algorithm. In particular, we show that GBDT converges to the solution of a certain Kernel Ridgeless Regression problem. We also introduce a simple modification of the classic algorithm allowing to sample from the Gaussian posterior. The proposed method gives much better knowledge uncertainty estimates than the existing approaches.
We highlight the following important directions for future research. First, analyze how one can control the kernel and use it for better knowledge uncertainty estimates. Also, we do not analyze generalization in the current work, which is another important research topic.

References

[1] C. J. Burges. From RankNet to LambdaRank to LambdaMART: An overview. *Learning*, 11(23-581):81, 2010.

[2] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.

[3] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[4] Y. Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[5] Y. Gal and Z. Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proc. 33rd International Conference on Machine Learning (ICML-16)*, 2016.

[6] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021.

[7] D. Hendrycks and K. Gimpel. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. http://arxiv.org/abs/1610.02136, 2016. arXiv:1610.02136.

[8] B. Ibragimov and G. Gusev. Minimal variance sampling in stochastic gradient boosting. *Advances in Neural Information Processing Systems*, 32, 2019.

[9] L. Katzir, G. Elidan, and R. El-Yaniv. Net-dnf: Effective deep modeling of tabular data. In *International Conference on Learning Representations (ICLR)*, 2021.

[10] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Proc. Conference on Neural Information Processing Systems (NIPS)*, 2017.

[11] D. G. Luenberger. *Optimization by Vector Space Methods*. 1969.

[12] A. Malinin. *Uncertainty Estimation in Deep Learning with application to Spoken Language Assessment*. PhD thesis, University of Cambridge, 2019.

[13] A. Malinin, B. Mlodozeniec, and M. J. Gales. Ensemble distribution distillation. In *International Conference on Learning Representations*, 2020.

[14] A. Malinin, L. Prokhorenkova, and A. Ustimenko. Uncertainty in gradient boosting via ensembles. In *International Conference on Learning Representations*, 2021.

[15] Matthews, Hron, Turner, and Ghahramani. Sample-then-optimize posterior sampling for bayesian linear models. In *NIPS 2017*, 2017.

[16] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. Catboost: unbiased boosting with categorical features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS)*, pages 6638–6648, 2018.

[17] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. the MIT press, 2006.

[18] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*, pages 521–530. ACM, 2007.
[19] B. P. Roe, H.-J. Yang, J. Zhu, Y. Liu, I. Stancu, and G. McGregor. Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 543(2):577–584, 2005.

[20] A. Ustimenko and L. Prokhorenkova. Sglb: Stochastic gradient langevin boosting. In *International Conference on Machine Learning*, pages 10487–10496. PMLR, 2021.

[21] Q. Wu, C. J. Burges, K. M. Svore, and J. Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270, 2010.

[22] Y. Zhang and A. Haghani. A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58:308–324, 2015.

A Notation used in the paper

For convenience, let us list some frequently used notation:

- $X \subset \mathbb{R}^d$ — feature space;
- $d$ — dimension of feature vectors;
- $Y \subset \mathbb{R}$ — target space;
- $\rho$ — distribution of features;
- $N$ — number of samples;
- $z = (x_N, y_N)$ — dataset;
- $\mathcal{V}$ — set of all possible tree structures;
- $L_{\nu} : \mathcal{V} \rightarrow \mathbb{N}$ — number of leaves for $\nu \in \mathcal{V}$;
- $D(\nu, r)$ — score used to choose a split [3];
- $\mathcal{S}$ — indices of all possible splits;
- $n$ — number of borders in our implementation of SampleTree;
- $m$ — depth of the tree in our implementation of SampleTree;
- $\beta$ — random strength;
- $\epsilon$ — learning rate;
- $\mathcal{F}$ — space of all possible ensembles of trees from $\mathcal{V}$;
- $\phi_{\nu} : X \rightarrow \{0, 1\}^{L_{\nu}}$ — tree structure;
- $\phi_{\nu}^{(j)}$ — indicator of $j$-th leaf;
- $V(f)$ — empirical error of a model $f$ relative to the best possible $f^* \in \mathcal{F}$;
- $k_{\nu}([\cdot, \cdot])$ — single tree kernel;
- $K([\cdot, \cdot])$ — stationary kernel of the gradient boosting;
- $p([|f, \beta])$ — distribution of trees, $f \in \mathcal{F}$;
- $\pi(\cdot) = \lim_{\beta \rightarrow \infty} p([|f, \beta]) = p([|f^*, \beta])$ — stationary distribution of trees;
- $\sigma$ — kernel scale.

B Convex optimization in functional spaces

In this section, we formulate basic definitions of differentiability in functional spaces and the theorem on the convergence of gradient descent in functional spaces. For the proof of the theorem and further details on convex optimization in functional space, the reader can consult [11].

We consider $\mathcal{H}$ to be a Hilbert space with some scalar product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. 

11
**Definition B.1.** We say that \( F: \mathcal{H} \to \mathbb{R} \) is Fréchet differentiable if for any \( f \in \mathcal{H} \) there exists a bounded linear functional \( L_f: \mathcal{H} \to \mathbb{R} \) such that \( \forall h \in \mathcal{H} \)
\[
F(f + h) = F(f) + L_f[h] + o(\|h\|).
\]
The value of \( L_f: \mathcal{H} \to \mathbb{R} \) is denoted by \( D_f F \) and is called a Fréchet differential of \( F \) at point \( f \). So, Fréchet differential is a functional \( D_f F: \mathcal{H} \to \mathcal{B}(\mathcal{H}, \mathbb{R}) \), where \( \mathcal{B}(X, Y) \) denotes a normed space of linear bounded functionals from \( X \) to \( Y \).

**Definition B.2.** Let \( F: \mathcal{H} \to \mathbb{R} \) be Fréchet differentiable with a Fréchet differential \( D_f F \) that is a bounded linear functional. Then, by the Riesz theorem there exists a unique \( h_f \) such that \( (D_f F[f])[h] = (h_f, h)_{\mathcal{H}} \forall h \in \mathcal{H} \). We call such element a gradient of \( F \) in \( \mathcal{H} \) at \( f \in \mathcal{H} \) and denote it by \( \nabla_F F(f) = h_f \in \mathcal{H} \).

**Definition B.3.** \( F: \mathcal{H} \to \mathbb{R} \) is said to be twice Fréchet differentiable if \( D_f F \) is Fréchet differentiable, where the definition of Fréchet differential is analogous to Definition B.1 with the only difference that \( D_f F \) takes values in \( \mathcal{B}(\mathcal{H}, \mathcal{B}(\mathcal{H}, \mathbb{R})) \). The second Fréchet differential is denoted by \( D^2 F: \mathcal{H} \to \mathcal{B}(\mathcal{H}, \mathcal{B}(\mathcal{H}, \mathbb{R})) \). As there is an isomorphism between \( \mathcal{B}(\mathcal{H}, \mathcal{B}(\mathcal{H}, \mathbb{R})) \) and \( \mathcal{B}(\mathcal{H} \times \mathcal{H}, \mathbb{R}) \), we can consider the second Fréchet differential to take values in \( \mathcal{B}(\mathcal{H} \times \mathcal{H}, \mathbb{R}) \). Henceforth, we will not differentiate between \( \mathcal{B}(\mathcal{H}, \mathcal{B}(\mathcal{H}, \mathbb{R})) \) and \( \mathcal{B}(\mathcal{H} \times \mathcal{H}, \mathbb{R}) \).

**Definition B.4.** A linear operator \( P: \mathcal{H} \times \mathcal{H} \to \mathbb{R} \) is said to be semi-positive definite (denoted by \( P \geq 0 \)) if \( \forall f, g \in \mathcal{H} \) we have \( P(f, f) \geq 0 \). \( P \) is said to be positive definite (denoted by \( P > 0 \)) if \( \forall f \in \mathcal{H} \setminus \{0\} : P(f, f) > 0 \).

**Definition B.5.** Given two linear operators \( P, G: \mathcal{H} \times \mathcal{H} \to \mathbb{R} \) we write \( P \succeq G \) if \( P - G \geq 0 \) and \( P \succ G \) if \( P - G > 0 \).

Let \( I \in \mathcal{B}(\mathcal{H} \times \mathcal{H}, \mathbb{R}) \) be a linear operator defined as \( I(g, h) = (g, h)_{\mathcal{H}} \).

**Theorem B.6.** Let \( F \) be bounded below and twice Fréchet differentiable functional on a Hilbert space \( \mathcal{H} \). Assume that \( D^2 F(f) \) satisfies \( 0 \prec mI \preceq D^2 F(f) \preceq \mu I \). Then the gradient descent scheme:
\[
f_{k+1} = f_k - \epsilon \nabla_F F(f_k)
\]
converges to \( f^* \) that minimizes \( F \).

**Proof.** For the proof see [11].

---

**C Kernel Ridge Regression and RKHS**

**Definition C.1.** \( \mathcal{K}: X \times X \to \mathbb{R} \) is called a kernel function if it is positive semi-definite, i.e., \( \forall N \in \mathbb{N}^+ \forall x_N \in \mathbb{R}^N : \mathcal{K}(x_N, x_N) \geq 0 \).

**Definition C.2.** For any kernel function we can define a Reproducing Kernel Hilbert Space (RKHS)
\[
\mathcal{H}(\mathcal{K}) = \text{span} \{ \mathcal{K}(\cdot, x) | x \in X \} \subset L_2(\rho)
\]
with a scalar product such that
\[
\langle f, \mathcal{K}(\cdot, x) \rangle_{\mathcal{H}(\mathcal{K})} = f(x).
\]
Alternatively, we define the scalar product as
\[
\langle f, f \rangle_{\mathcal{H}} = \min_{g \in L_2(\rho)} \int_X g^2(x)\rho(dx) \text{ s.t. } \mathcal{K}^{\frac{1}{2}} g = f, \quad (f, g)_{\mathcal{H}} = \frac{1}{4} (\langle f + g, f + g \rangle_{\mathcal{H}} - \langle f - g, f - g \rangle_{\mathcal{H}}),
\]
where \( \mathcal{K}^{\frac{1}{2}} \) corresponds to the unique SPD square root operator such that \( \left( \mathcal{K}^{\frac{1}{2}} \right)^2 = \mathcal{K} \) assuming that it acts as \( \mathcal{K}[g] = \int_X \mathcal{K}(\cdot, x) g(x)\rho(dx) \in L_2(\rho) \) for any \( g \in L_2(\rho) \). Both define the same scalar product.
Then, the existence and uniqueness of \( f \) hold.

Following two lemmas hold.

\[
L(f, \lambda) = \frac{1}{2N} \| y_N - f(x_N) \|^2 + \lambda \| f \|^2_{H(K)} \rightarrow \min_{f \in H(K)}
\]

and the following Kernel Ridgeless Regression problem:

\[
L(f) = \frac{1}{2N} \| y_N - f(x_N) \|^2 \rightarrow \min_{f \in H(K)}.
\]

**Lemma C.3.** \( \min_{H(K)} L(f, \lambda) \) has the only solution

\[
f^*_\lambda = K(\cdot, x_N)(K(x_N, x_N) + \lambda NI)^{-1} y_N.
\]

**Proof.** First, let us show that \( f^*_\lambda \in \text{span} \{ K(\cdot, x_i) \} \). Let \( H(K) = \text{span} \{ K(\cdot, x_i) \} \oplus \text{span} \{ K(\cdot, x_i) \}^\perp \) and consider the projector \( P : H(K) \rightarrow H(K) \) onto the space span \( \{ K(\cdot, x_i) \} \). It is easy to show that \( P(f)(x_N) = f(x_N) \) for any \( f \in H(K) \). Indeed, \( (f - P(f))[x_N] = \langle f - P(f), K(\cdot, x_N) \rangle = 0 \). If \( f^*_\lambda \) does not lie in \( \text{span} \{ K(\cdot, x_i) \} \), then \( \| f^*_\lambda \|_{H(K)} > \| P(f^*_\lambda) \|_{H(K)} \) and \( L(P(f^*_\lambda), \lambda) < L(f^*_\lambda, \lambda) \). We get a contradiction with the minimality of \( f^*_\lambda \).

Now, let us prove the existence of \( f^*_\lambda \). Consider \( f = K(\cdot, x_N)c \), where \( c \in \mathbb{R}^N \). Then we find the optimal \( c \) by taking a derivative of \( L(f, \lambda) \) with respect to \( c \) and equating it to zero:

\[
\frac{1}{N} K(x_N, x_N)(K(x_N, x_N)c - y_N) + \lambda K(x_N, x_N)c = 0.
\]

Then, \( c = (K(x_N, x_N) + \lambda NI)^{-1}(y_N + v) \), where \( v \in \ker K(x_N, x_N) \). Note that all \( K(\cdot, x_N)c \), are equal. Then, we have the only solution of the KRR problem:

\[
f^*_\lambda = K(\cdot, x_N)(K(x_N, x_N) + \lambda NI)^{-1} y_N.
\]

**Lemma C.4.** \( \min_{H(K)} L(f) \) has the only solution in \( \text{span} \{ K(\cdot, x_i) \} \) and it is the solution of minimum RKHS norm:

\[
f_* = K(\cdot, x_N)K(x_N, x_N)^\dagger y_N.
\]

**Proof.** Consider \( f = K(\cdot, x_N)c \), where \( c \in \mathbb{R}^N \). Now consider \( L(f) \) and differentiate it with respect to \( c \). If we equate the derivative to zero, we get:

\[
\frac{1}{N} K(x_N, x_N)(K(x_N, x_N)c - y_N) = 0.
\]

Then, \( K(x_N, x_N)c - (y_N + v) = 0 \) for some \( v \in \ker K(x_N, x_N) \). Note that \( y_N + \ker K(x_N, x_N) \cap \text{Im} K(x_N, x_N) \neq \emptyset \). Then, for any \( v \) such that \( y_N + v \in \text{Im} v(x_N, x_N) \) there exist a solution \( c_v = K(x_N, x_N)^\dagger(y_N + v) + \ker K(x_N, x_N) \). This follows from the fact that \( K(x_N, x_N)K(x_N, x_N)^\dagger \) is an orthoprojector onto \( \text{Im} K(x_N, x_N) \). Note that

\[
f_* = K(\cdot, x_N)(K(x_N, x_N)^\dagger(y_N + v) + \ker K(x_N, x_N)) = K(\cdot, x_N)K(x_N, x_N)^\dagger y_N.
\]

Then, the existence and uniquenss of \( f_* \) follow.

Now, consider a linear space \( F \subset L_2(\rho) \) of all possible ensembles of trees from \( \mathcal{V} \):

\[
F = \text{span} \{ \phi_{j}(\cdot) : X \rightarrow \{0, 1\} \ | \ j \in \{1, \ldots, L_\nu\} \}.
\]

Consider such functional on space \( F \):

\[
V(f) = \frac{1}{2N} \| y_N - f(x_N) \|^2 - \inf_{f \in F} \left( \frac{1}{2N} \| y_N - f(x_N) \|^2 + \| f \|^2_{L_2(\rho)} \right).
\]

and define unique function:

\[
f_* = \lim_{\lambda \rightarrow 0} \arg \min_{f \in F} (V(f) + \lambda\| f \|^2_{L_2(\rho)}) \in \arg \min_{f \in F} V(f).
\]

Then following two lemmas hold.
We will show that it satisfies the conditions needed for Theorem B.6. We will also deduce the
for small enough $\langle y_N - f_*(x_N), f(x_N) \rangle_{\mathbb{R}^N} > 0$. We have:

$$
\| y_N - (f_* + \alpha f)(x_N) \|^2_{\mathbb{R}^N} = \| y_N - f_*(x_N) \|^2_{\mathbb{R}^N} - 2\alpha\langle y_N - f_*(x_N), f(x_N) \rangle_{\mathbb{R}^N} + \alpha^2 \| f(x_N) \|^2_{\mathbb{R}^N} < \| y_N - f_*(x_N) \|^2_{\mathbb{R}^N}
$$

for small enough $\alpha > 0$, which contradicts with the definition of $f_*$. \hfill \square

**Lemma C.6** (Lemma 3.3 in the main text). $V(f) = \frac{1}{2N} \| f_*(x_N) - f(x_N) \|^2_{\mathbb{R}^N}$.

**Proof.** By definition,

$$
V(f) = \frac{1}{2N}\| y_N - f(x_N) \|^2_{\mathbb{R}^N} - \frac{1}{2N}\| y_N - f_*(x_N) \|^2_{\mathbb{R}^N}.
$$

Now, let us prove that

$$
\| y_N - f(x_N) \|^2_{\mathbb{R}^N} - \| y_N - f_*(x_N) \|^2_{\mathbb{R}^N} - \| f_*(x_N) - f(x_N) \|^2_{\mathbb{R}^N} = 0.
$$

Indeed,

$$
\| y_N - f(x_N) \|^2_{\mathbb{R}^N} - \| y_N - f_*(x_N) \|^2_{\mathbb{R}^N} - \| f_*(x_N) - f(x_N) \|^2_{\mathbb{R}^N} = -2\langle f_*(x_N), f(x_N) \rangle_{\mathbb{R}^N} - 2\langle y_N, f(x_N) - f_*(x_N) \rangle_{\mathbb{R}^N} + 2\langle f_*(x_N), f(x_N) \rangle_{\mathbb{R}^N} = -2\langle y_N - f_*(x_N), f(x_N) - f_*(x_N) \rangle_{\mathbb{R}^N} = 0,
$$

where the last equality follows from the previous lemma. \hfill \square

### D Gaussian Process inference

In this section, we prove Lemma 3.1 from 2.13 of the main text.

Firstly, consider the Kernel Ridgeless Regression problem loss functional:

$$
L(f) = \frac{1}{2N} \sum_{i=1}^{N} (f(x_i) - y_i)^2 \to \min_{f \in \mathcal{H}(\mathcal{K})}.
$$

We will show that it satisfies the conditions needed for Theorem B.6. We will also deduce the
formula of the gradient of $L$ in order to show that gradient descent takes the form (2).

**Lemma D.1.** $L(f)$ is Fréchet differentiable with the differential given by:

$$
\mathcal{D}_f L(f) = \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - y_i)e_{v_{x_i}},
$$

where $e_{v_{x_i}} : \mathcal{H}(\mathcal{K}) \to \mathbb{R}$ is a bounded linear functional such that $e_{v_{x_i}}(f) = f(x_i) = (f, \mathcal{K}(x_i, \cdot))_{\mathcal{H}(\mathcal{K})}$\footnote{We further use the notation $\mathcal{K}_{x_i} := \mathcal{K}(\cdot, x_i)$.}

**Proof.** As Fréchet differential is linear, we only need to find Fréchet differential for $(f(x_i) - y_i)^2$.

Note that $(f(x_i) - y_i)^2$ is a composition of two functions:

$$
F : \mathcal{H}(\mathcal{K}) \rightarrow \mathbb{R}, \quad F = e_{v_{x_i}} - y_i, \\
G : \mathbb{R} \rightarrow \mathbb{R}, \quad G(x) = x^2.
$$
The differential of the composition can be found as:
\[ D_f G(F(f)) = \frac{\partial}{\partial x} G(F(f)) D_f F(f), \]
\[ D_f G(F(f)) = 2(f(x_i) - y_i)ev_{x_i}, \]
where \( D_f F(f) = ev_{x_i} \), because
\[ ev_{x_i}(f + h) - y_i = ev_{x_i}(f) - y_i + ev_{x_i}(h). \]

**Lemma D.2.** The gradient of \( L(f) \), Riesz representative of the functional above, is given by:
\[ \nabla_f L(f) = \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - y_i)K_{x_i}. \]

**Proof.** Trivially follows from the previous lemma.

**Lemma D.3.** \( L \) is twice Fréchet differentiable with the differential given by:
\[ D^2_f L : \mathcal{H}(\mathcal{K}) \to B(\mathcal{H}(\mathcal{K}), B(\mathcal{H}(\mathcal{K}), \mathbb{R})), \]
\[ D^2_f L(f)[h] = \frac{1}{N} \sum_{i=1}^{N} h(x_i)ev_{x_i}. \]

**Proof.** Due to the linearity of Fréchet differential and lemma D.1 we need to find only Fréchet differential for \( (f(x_i) - y_i)ev_{x_i} \).

Consider \( S(f) = (f(x_i) - y_i)ev_{x_i} \). Then we need to find \( L_f \in B(\mathcal{H}(\mathcal{K}), B(\mathcal{H}(\mathcal{K}), \mathbb{R})) \) such that \( S(f + h) = S(f) + L_f[h] + o(\|h\|) \).

It is easy to show that \( h \mapsto h(x_i)ev_{x_i} \in B(\mathcal{H}(\mathcal{K}), B(\mathcal{H}(\mathbb{R}), \mathbb{R})) \) and \( S(f + h) = S(f) + h(x_i)ev_{x_i} \). Thus, we get that \( D_f S(f)[h] = h(x_i)ev_{x_i} \). From this the statement of the lemma follows.

We further assume that \( \mathcal{H} = \text{span} \{ K(\cdot,x_i) \mid i = 1, \ldots, N \} \). Then we deduce following lemma:

**Lemma D.4.** We have \( 0 \leq m I \leq D^2_f L(f) \leq \mu 1 \) for some positive \( m, \mu \).

**Proof.** To get \( \mu \), we only need to find it for the functional \( h \mapsto h(x_i)ev_{x_i} \). As we discussed in Appendix B we do not differentiate between \( B(\mathcal{H}(\mathcal{K}), B(\mathcal{H}(\mathcal{K}), \mathbb{R})) \) and \( B(\mathcal{H}(\mathcal{K}) \times \mathcal{H}(\mathcal{K}), \mathbb{R}) \).

So, let us find \( \mu \) for the functions \( T_i(f,g) = f(x_i)g(x_i) \):
\[ 0 \leq T_i(f,f) = f(x_i)^2 \leq \|K_{x_i}\|^2_{H(\mathcal{K})} \|f\|^2_{H(\mathcal{K})}. \]

Thus, we get \( \mu = \frac{1}{N} \sum_{i=1}^{N} \|K_{x_i}\|^2_{H(\mathcal{K})} \).

This way we showed that \( L(f) \) is quadratic convex. Assuming \( \mathcal{H} = \text{span} \{ K(\cdot,x_i) \mid i = 1, \ldots, N \} \) ensures that we uniquely reconstruct the function \( f \in \mathcal{H} \) by its values \( f(x_1), \ldots, f(x_N) \) which due to connection between system of linear equations and quadratic \( L(f) \) is equivalent to saying that the matrix defining the quadric is non-singular therefore the quadric \( L(f) \) is strongly convex.

Given all the above lemmas, as a corollary of Theorem B.6 we have the following.

**Corollary D.5.** Gradient descent, defined by the following iterative scheme:
\[ F_{n+1} = F_n - \frac{1}{N} \sum_{i=1}^{N} (F_n(x_i) - y_i)K_{x_i}, \]
\[ F_0 = 0_{L_2(\rho)} \]
converges to the optimum \( L^* \). Thus,
\[ F_\infty = \lim_{n \to \infty} F_n = K(\cdot,x_N)K(x_N,x_N)^\dagger y_N. \]
\[ (6) \]
Proof. As mentioned in Section 2.3, the solution of the Kernel Ridgeless Regression problem coincides with a posterior of a Gaussian. Since gradient descent converges to this solution and thus to the posterior of a Gaussian, equation (6) follows.

Lemma D.6. Consider the gradient descent:

\[ F_{n+1} = F_n - \epsilon \frac{1}{N} \sum_{i=1}^{N} (F_n(x_i) - y_i)K_{x_i}, \]
\[ F_0 = 0_{L_2(\rho)}, \]
\[ F_{\infty} = \lim_{n \to \infty} F_n \]

and the following randomization scheme:
1. sample \( f^{\text{init}} \sim GP(0_{L_2(\rho)}, \sigma^2 K); \)
2. set new labels \( y_N^{\text{new}} = y_N - f^{\text{init}}(x_N); \)
3. fit GD \( F_{\infty}(\cdot) \) on \( y_N^{\text{new}} \) assuming \( F_0(\cdot) = 0_{L_2(\rho)}; \)
4. output \( \hat{f}(\cdot) = f^{\text{init}}(\cdot) + F_{\infty}(\cdot) \) as final model.

Then, \( \hat{f} \) from the scheme above follows the Gaussian Process posterior with the following mean and covariance:
\[
\hat{f}_x(x) = K(x, x_N)K(x_N, x_N)^{\dagger}y_N, \\
\tilde{K}(x, x) = \sigma^2 (K(x, x) - K(x, x_N)K(x_N, x_N)^{\dagger}K(x_N, x)).
\]

Proof.
\[
F_{\infty} = K(\cdot, x_N)K(x_N, x_N)^{\dagger}y_N^{\text{new}} = K(\cdot, x_N)K(x_N, x_N)^{\dagger}(y_N - f^{\text{init}}(x_N)).
\]

Let us find the distribution of \( \hat{f} \) at \( x \in \mathbb{R}^n \). It can be easily seen that:
\[
\mathbb{E}\hat{f}(x) = K(x, x_N)K(x_N, x_N)^{\dagger}y_N.
\]

Let us now calculate covariance:
\[
\text{cov}\hat{f}(x) = \mathbb{E}(\hat{f}(x) - \mathbb{E}\hat{f}(x))(\hat{f}(x) - \mathbb{E}\hat{f}(x))^T
\]
\[
= \mathbb{E}(f^{\text{init}}(x) - K(x, x_N)K(x_N, x_N)^{\dagger}f^{\text{init}}(x_N))(f^{\text{init}}(x) - K(x, x_N)K(x_N, x_N)^{\dagger}f^{\text{init}}(x_N))^T
\]
\[
= \mathbb{E}f^{\text{init}}(x)f^{\text{init}}(x)^T - \mathbb{E}f^{\text{init}}(x)f^{\text{init}}(x)^TK(x_N, x_N)^{\dagger}K(x_N, x)
\]
\[
- K(x, x_N)K(x_N, x_N)^{\dagger}\mathbb{E}f^{\text{init}}(x)f^{\text{init}}(x)^T
\]
\[
+ K(x, x_N)K(x_N, x_N)^{\dagger}\mathbb{E}f^{\text{init}}(x_N)f^{\text{init}}(x_N)^TK(x_N, x_N)^{\dagger}K(x_N, x)
\]
\[
= \sigma^2 (K(x, x) - 2K(x, x_N)K(x_N, x_N)^{\dagger}K(x_N, x)
\]
\[
+ K(x, x_N)K(x_N, x_N)^{\dagger}K(x_N, x_N)^{\dagger}K(x_N, x))
\]
\[
= \sigma^2 (K(x, x) - K(x, x_N)K(x_N, x_N)^{\dagger}K(x_N, x_N)^{\dagger}K(x_N, x))
\]

which is exactly what we need. \( \square \)

E  Distribution of trees

Lemma E.1 (Lemma 3.5 in the main text).

\[
p(\nu|f, \beta) = \sum_{\varsigma \in P_m} \prod_{i=1}^{m} \frac{e^{\beta D(\nu_{\varsigma, i-1})}}{\sum_{\varsigma \in S(\nu_{\varsigma, i-1}, i-1)} e^{\beta D(\nu_{\varsigma, i-1}, i-1)}} = \frac{e^{\beta D(\nu)}}{\sum_{\nu \in P_m} e^{\beta D(\nu)}},
\]

where the sum is over all permutations \( \varsigma \in P_m, \nu_{\varsigma, i-1} = (s_{\varsigma(1)}, \ldots, s_{\varsigma(i-1)}), \) and \( \nu = (s_1, \ldots, s_m). \)
Proof. Let us fix some permutation \( \varsigma \in \mathcal{P}_m \). W.l.o.g., let \( \varsigma = \text{id}_{\mathcal{P}_m} \), i.e. \( \varsigma(i) = i \forall i \). It remains to derive the formula for the fixed permutation. The probability of adding the next split given the previously build tree is:

\[
P(\nu_{i-1} \cup s_i | \nu_{i-1}) = \frac{e^{\frac{1}{\beta} D(\nu, r)}}{\sum_{\nu' \in \mathcal{S} \setminus \nu_{i-1}} e^{\frac{1}{\beta} D((\nu_{i-1}, s), r)}},
\]

which comes from (4) and the Gumbel-SoftMax trick. Then, we decompose the probability \( P(\nu) \) of a tree as:

\[
P(\nu) = \prod_{i=1}^{m} P(\nu_{i-1} \cup s_i | \nu_{i-1}),
\]

and so for the fixed permutation we have

\[
P(\nu) = \prod_{i=1}^{m} \frac{e^{\frac{1}{\beta} D(\nu, r)}}{\sum_{\nu' \in \mathcal{S} \setminus \nu_{i-1}} e^{\frac{1}{\beta} D((\nu_{i-1}, s), r)}}.
\]

Then we sum over all permutations and the lemma follows.

\[\]

Now, let us define the following value indicating how different are the distribution of trees for \( f \) and \( f_* \):

\[
\Gamma_\beta(f) = \max \left\{ \left\| \frac{dp(z | f_*, \beta)}{dp(z | f, \beta)} \right\|_{L_\infty(\pi)}, \left\| \frac{dp(z | f, \beta)}{dp(z | f_*, \beta)} \right\|_{L_\infty(\pi)} \right\}.
\]

Lemma E.2. The following bound relates the distributions uniformly. For any \( R \in \mathbb{R} \):

\[
\sup_{f \in \mathcal{F} : V(f) \leq \frac{1}{\beta} R^2} \Gamma_\beta(f) \leq e^{-\frac{m R^2}{\beta}}.
\]

Proof. Consider \( \pi = p(z | f_*, \beta) \) and the following expression \( P(\nu, \varsigma) \):

\[
P(\nu, \varsigma) := \prod_{i=1}^{m} \frac{e^{\frac{1}{\beta} D(\nu, r)}}{\sum_{\nu' \in \mathcal{S} \setminus \nu_{i-1}} e^{\frac{1}{\beta} D((\nu_{i-1}, s), r)}},
\]

Then,

\[
\sum_{\varsigma \in \mathcal{P}_m} P(\nu, \varsigma) \leq e^{\frac{1}{\beta} D(\nu, r)} \prod_{i=1}^{m} \frac{\sum_{\nu' \in \mathcal{S} \setminus \nu_{i-1}} e^{\frac{1}{\beta} D((\nu_{i-1}, s), r)}}{e^{\frac{1}{\beta} D(\nu_{i-1}, s), r}} \leq e^{\frac{2m V(f)}{\beta}} \pi(\nu) \leq e^{-\frac{m R^2}{\beta}} \pi(\nu).
\]

By noting that the probabilities remain the same if we shift \( D(\cdot, r) \leftarrow D(\cdot, r) - 2V(f) \) which becomes everywhere non-positive and allows us to do the above trick once more but in reverse manner: if we formally replace the \( D \) with such modified function and repeat the steps with reversing the inequalities which is needed since the new function is everywhere negative then the lemma follows.

\[
\sum_{\varsigma \in \mathcal{P}_m} P(\nu, \varsigma) \geq e^{-\frac{1}{\beta} D(\nu, r) - 2V(f)} \prod_{i=1}^{m} \frac{\sum_{\nu' \in \mathcal{S} \setminus \nu_{i-1}} e^{\frac{1}{\beta} D((\nu_{i-1}, s), r)}}{e^{\frac{1}{\beta} D((\nu_{i-1}, s), r)}} \geq e^{-\frac{2m V(f)}{\beta}} \pi(\nu) \geq e^{-\frac{m R^2}{\beta}} \pi(\nu).
\]

\[\]

F Proof of Theorem 4.1

F.1 RKHS structure

Here we formulate and prove several lemmas about the RKHS structure and operators \( \Sigma, \Sigma_f, \Sigma_v \) which we introduced in Section 3.3.

Remark F.1. Since the rank of \( \Sigma_f \) does not decay with \( N \), in the limit it coincides with the rank of \( \mathcal{K} \) due to the central limit theorem as it should converge to the latter in the limit, and since we assumed low-rankness, then with probability almost one the rank of \( \Sigma_f \) coincides with the one of \( \mathcal{K} \). Thus, in the paper we are going to ignore cases when it is lesser.
Lemma F.2. The following formula holds \( \forall h, g \in \mathcal{H} \):

\[
\langle \Sigma[h], g \rangle_{\mathcal{H}} = \frac{1}{N} \sum_{i=1}^{N} h(x_i)g(x_i).
\]

Lemma F.3. (Covariation majorization) The following operator inequality holds \( \forall \nu \in \mathcal{V} \) and \( \forall f \in \mathcal{H} \):

\[
\lambda_{\text{max}}(\Sigma) \leq 1,
\]

\[
\lambda_{\text{min}}(\Sigma) \geq \frac{1}{N},
\]

\[
\lim_{N \to \infty} \lambda_{\text{min}}(\Sigma) \geq e^{-\frac{m^2}{N}} \lambda_{\infty}, \lambda_{\infty} > 0.
\]

Proof. We will show that \( \lambda_{\text{max}}(\Sigma) \leq 1 \) where \( \pi \) in the definition of \( \Sigma \) can be chosen arbitrary. This way the statement would follow for \( \Sigma, \Sigma_f, \Sigma_\nu \) by taking \( \pi \) to be equal \( p(\cdot \mid f^*, \beta), p(\cdot \mid f, \beta) \), or \( \delta(\cdot) \) respectively. The operator \( \Sigma \) is dual to \( K_N = \frac{1}{N} K(x_N, x_N) \) since we can define the operator

\[
S_N[f] = \left[ \langle f, K(\cdot, x_1) \rangle_{\mathcal{H}}, \ldots, \langle f, K(\cdot, x_N) \rangle_{\mathcal{H}} \right] \in \mathbb{R}^N,
\]

so that \( \Sigma = \frac{1}{N} S_N^* S_N \) and \( K_N = \frac{1}{N} S_N S_N^* \) and, thus, has the same spectra.

Therefore, instead of \( \Sigma \), we can study the maximal eigenvalue of \( K_N \) but the latter admits another decomposition

\[
K_N = \int_{\mathcal{V}} \Phi_{\nu}^T \{ \Phi_{\nu} \Phi_{\nu}^T \} \Phi_{\nu} \pi(d\nu),
\]

where \( \Phi_{\nu} = \phi_{\nu}(x_N) \in \mathbb{R}^{L \times N} \). The key observation is that the sub-expectation expression

\[
\Phi_{\nu}^T \{ \Phi_{\nu} \Phi_{\nu}^T \} \Phi_{\nu}
\]

defines a matrix with maximal eigenvalue bounded by 1 since the former is an orthoprojector which is also shown in [20]. Therefore, the expectation also inherits this bound on the maximal eigenvalue which with symmetry of \( K_N \) gives us the desired inequality.

Now we need to show that \( \lambda_{\text{min}}(\Sigma_f) \geq \frac{1}{N} \). Similarly to the previous proof, we need to show that \( \lambda_{\text{min}}(\Sigma_{f_i}) \geq \frac{1}{N} \) and then to replace \( \pi \leftarrow p(\cdot \mid f, \beta) \). Consider the following formula:

\[
\Sigma = \frac{1}{N} \sum_{i=1}^{N} K(\cdot, x_i) \otimes \mathcal{K}(\cdot, x_i),
\]

\[
\Sigma = \frac{1}{N} \sum_{i=1}^{N} K(x_i, x_i), \quad \frac{K(x_i, x_i)}{\sqrt{K(x_i, x_i)} \otimes \sqrt{K(x_i, x_i)}},
\]

where \( (a \otimes \mathcal{K} b)[c] = \langle b, c \rangle_{\mathcal{H}} a \). If \( a = b \) and \( \|a\|_{\mathcal{H}} = 1 \), then 1 and 0 are the only eigenvalues of \( a \otimes \mathcal{K} a \). Finally, by noting that \( \inf_{x \in \mathcal{X}} K(x, x) \geq 1 \), we obtain the bound.

Finally, Lemma [\ref{lemma:bounded_eigenvalues}](\ref{lemma:bounded_eigenvalues}) implies that \( \lambda_{\text{min}}(\Sigma_f) \geq \frac{1}{N} \lambda_{\text{min}}(\Sigma_{f_i}) \) which gives us the bound between eigenvalues and then \( \lambda_{\text{min}}(\Sigma) \to \lambda_{\infty} > 0 \) holds due to finite dimension of \( \mathcal{F} \) and convergence of \( \Sigma \) to \( \mathcal{K} \) by Law of Large Numbers.

Corollary F.4. (Expected squared norm majorization by RKHS norm) The following bound holds \( \forall f \in \mathcal{H} \):

\[
\|f\|_{L_2(\rho)} \leq \|f\|_{\mathcal{H}}.
\]

Proof. From \( \lambda_{\text{max}}(\Sigma) \leq 1 \) it follows that \( \lambda_{\text{max}}(\mathcal{K}) = \lim_{N \to \infty} \lambda_{\text{max}}(\Sigma) \leq 1 \). Let \( \mathcal{K}^{-1} \) be taken as the unique inverse \( \mathcal{K}^{-1} \) on \( \mathcal{F} \) and for arbitrary \( f \in L_2(\rho) \) we consider \( L_2(\rho) = \mathcal{F} \oplus \mathcal{F}^\perp \) which implies \( f = f_1 + f_2, f_1 \in \mathcal{F}, f_2 \in \mathcal{F}^\perp \forall f \in L_2(\rho) \) and, thus, allows us to define \( \mathcal{K}^{-1} f = \mathcal{K}^{-1} f_1 \in \mathcal{F} \). Such definition of the inverse implies that \( \lambda_{\text{min}}(\mathcal{K}^{-1}) = \frac{1}{\lambda_{\text{max}}(\mathcal{K})} \geq 1 \) which when substituted back into the product

\[
\langle f, f \rangle_{\mathcal{H}} = \langle \mathcal{K}^{-\frac{1}{2}} f, \mathcal{K}^{-\frac{1}{2}} f \rangle_{L_2(\rho)} = \langle \mathcal{K}^{-1} f, f \rangle_{L_2(\rho)}
\]

implies the bound.
Lemma F.5. \( \|k_\nu(x_N, x_N)\| \leq N \) for any \( \nu \in \mathcal{V} \).

**Proof.** Consider \( S_1, ..., S_{L_\nu} \) — unions such that \( S_i = \{ x_j | \phi_\nu^{(i)}(x_j) = 1 \} \). Then, it is easy to see that
\[
k_\nu = \sum_{i=1}^{L_\nu} w_\nu^{(i)} \mathbf{1}_{N_\nu \times N_\nu}^T,
\]
where \( \mathbf{1}_{n \times n} \) is a matrix of size \( n \times n \) consisting of ones. Then, we note that \( \|\mathbf{1}_{n \times n}\| = n \) and now the statement of the lemma follows.

**F.2 Main Theorem**

**Theorem F.6** (Theorem F.1 in the main text). Let
\[
\lambda_* := \inf_{f \in \mathcal{F}_0} \lambda_{\min}(\Sigma_f) > 0
\]
and consider an arbitrary \( \epsilon, 0 < \epsilon \leq \lambda_* \). The following inequality holds:
\[
\mathbb{E}V(f_T) \leq \frac{1}{2} R^2 e^{-\lambda_* \epsilon T}.
\]

**Proof.** \( \lambda_* > 0 \) follows from lemma F.3. Note that \( V(f) \) can be written in the following form by lemmas F.3 and F.2:
\[
V(f) = \frac{1}{2N} \|f_\nu(x_N) - f(x_N)\|_R^2 = \frac{1}{2} \langle \Sigma [f_\nu - f], f_\nu - f \rangle_H.
\]
Now, we can write:
\[
\mathbb{E}_\nu V(f - \epsilon \Sigma_\nu[f - f_*]) = V(f) - \epsilon \langle \Sigma [f - f_*], \Sigma_\nu[f - f_*] \rangle_H + \frac{\epsilon^2}{2} \mathbb{E}_\nu \langle \Sigma \Sigma_\nu[f - f_*], \Sigma_\nu[f - f_*] \rangle_H.
\]
Then, we observe that
\[
\langle \Sigma [f - f_*], \Sigma_\nu[f - f_*] \rangle_H \geq \lambda_{\min}(\Sigma_f) \langle \Sigma [f - f_*], f - f_* \rangle_H \geq 2 \lambda_{\min}(\Sigma_f) V(f)
\]
and, moreover, for any \( \nu \in \mathcal{V} \):
\[
\langle \Sigma \Sigma_\nu[f - f_*], \Sigma_\nu[f - f_*] \rangle_H = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{1}{N} k_\nu(x_i, x_N)(f(x_N) - f_\nu(x_N)) \right)^2
\]
\[
= \frac{1}{N^2} \|k_\nu(x_N, x_N)(f(x_N) - f_\nu(x_N))\|^2_R \leq \frac{1}{N^2} \|k_\nu(x_N, x_N)\|^2 \|f(x_N) - f_\nu(x_N)\|^2_R
\]
\[
\leq \frac{1}{N} \|f(x_N) - f_\nu(x_N)\|^2_R = \langle \Sigma [f - f_*], f - f_* \rangle_H.
\]
where the last inequality follows by lemma F.3. Then,
\[
\mathbb{E}_\nu \langle \Sigma \Sigma_\nu[f - f_*], \Sigma_\nu[f - f_*] \rangle_H \leq \| \Sigma [f - f_*], f - f_* \rangle_H = 2V(f).
\]
Combining the above inequalities yields:
\[
\mathbb{E}_\nu V(f - \epsilon \Sigma_\nu[f - f_*]) \leq (1 - (2\lambda_* - \epsilon) \epsilon) V(f).
\]
Finally, \( V(f_0) \leq \frac{1}{2} R^2 \) holds by the assumptions. Therefore, by applying this to \( f_T \), we obtain the result.

**Corollary F.7.** (Convergence to the solution of the KRR / Convergence to the Gaussian Process posterior mean function). The following bound holds:
\[
\mathbb{E} \|f_T - f_*\|_{L_2(\nu)}^2 \leq \frac{e^{-\lambda_* T R^2}}{\lambda_*}.
\]
which suggests that if \( \lambda_* > O(1) > 0 \) and \( \epsilon = \lambda_* \), then for \( T \geq \frac{1}{\lambda_*^2} \log \left( \frac{R^2}{\lambda_* \epsilon} \right) = O\left( \log \left( \frac{1}{\epsilon} \right) \right) \) we obtain an error \( \delta \), i.e., linear convergence.
Proof. By Theorem F.6

\[ EV(f_T) \leq \frac{1}{2} R^2 e^{-\lambda_* c_T} \]

Also consider writing:

\[
V(f_T) = \frac{1}{2N} \| f_*(x_N) - f(x_N) \|_{\mathbb{R}^N}^2 = \frac{1}{2} \langle \Sigma [f_* - f], f_* - f \rangle_H \geq \frac{1}{2} \lambda_* \| f_* - f \|_H^2 \geq \frac{1}{2} \lambda_* \| f_* - f \|_{L_2(\rho)}^2
\]

where the last inequality follows by Corollary F.4. Then the result follows.

\[ \square \]

G Implementation details

In the experiments, we fix \( \sigma = 0.1 \) (scale of the kernel), which theoretically can be taken arbitrarily. As a hyperparameter (that is estimated on the validation set), we consider \( \beta \in \{10^{-2}, 10^{-1}, 1\} \). We use the standard CatBoost library and add the Gumbel noise term in selecting the trees for the “L2” scoring function, which is implemented in CatBoost out of the box but is not used by SGB and SGLB since it is not the default one for the library. Moreover, we do not consider subsampling of the data (as SGLB does also), and differently from SGB and SGLB, we disable the “boost-from-average” option. Finally, we set \( l_2\text{-leaf}\_\text{reg} \) value to 0, as SGLB does.