Boosting in Online Non-parametric Regression

Paul Liautaud[1], Pierre Gaillard[2] & Olivier Wintenberger[1,3]

[1] *Sorbonne Université, Laboratoire de Probabilités, Statistique et Modélisation (LPSM), F-75005 Paris, France ;*
{paul.liautaud,olivier.wintenberger}@sorbonne-universite.fr
[2] *Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France ;*
pierre.gaillard@inria.fr
[3] *Wolfgang Pauli Institut, c/o Fakultät für Mathematik, Universität Wien, 1090 Vienna, Austria*

Résumé. Dans de nombreuses applications, les données ne sont pas disponibles dès le départ pour apprendre un modèle, mais elles sont observées séquentiellement sous forme de flux de données. De plus, l'environnement est parfois si complexe qu'il est difficile sinon impossible de déterminer un modèle convenable et d'utiliser les techniques d'apprentissage statistique classiques. En particulier les hypothèses d'indépendance ou i.i.d. sur nos données peuvent ne plus être pertinentes. Il est ainsi nécessaire d'adopter une approche robuste en utilisant une méthode qui apprend au fur et à mesure, en tirant des enseignements des données au cours du temps. Tels sont les objectifs de la théorie de l'apprentissage en ligne.

D'autre part, le boosting est une puissante technique d'apprentissage par ensemble introduite par Freund et al. [4] ayant gagné en popularité dans les environnements d'apprentissage batch automatique. Son adaptation au paradigme d'apprentissage séquentiel présente alors des défis et des opportunités uniques. Des efforts récents ont notamment cherché à étendre l'efficacité des algorithmes de boosting à de tels contextes par exemple dans Beygelzimer et al. [1] ou dans Brukhim and Hazan [2]. En entraînant et en optimisant dynamiquement des weak learners (par exemple, de simples arbres de prédiction) sur des données collectées de manière séquentielle, le boosting en ligne permet d'améliorer les performances prédictives et l'adaptabilité aux distributions de données évolutives.

Nous considérerons ici le cadre de la régression non paramétrique séquentielle avec des paires de données (x_t, y_t) arbitraires (comme dans Rakhlin and Sridharan [7] ou Cesa-Bianchi and Lugosi [3]). En particulier, nous analyserons un algorithme de Boosting, en discutant de ses garanties de convergence et en les comparant aux taux optimaux (minimax) déjà établis sous certaines hypothèses par exemple dans Rakhlin and Sridharan [7].

Abstract. The rapid proliferation of data streams, coupled with the escalating complexity of data, has precipitated a shift towards sequential methods capable of processing information in real-time. As a consequence, traditional statistical assumptions like stationarity or independently and identically distributed data are no longer tenable. In this context, designing algorithms that can adapt to evolving data streams with minimal assumptions is imperative.

On the other hand, Boosting, a powerful ensemble learning technique presented in Freund et al. [4], has gained significant traction in offline machine learning settings. However, its adaptation to the sequential learning paradigm presents unique challenges and opportunities. Recent efforts have sought to extend the efficacy of boosting algorithms to online learning paradigms for instance in Beygelzimer et al. [1] or Brukhim and Hazan [2]. By dynamically training and optimizing weak learners (e.g. simple decision trees) based on sequentially collected data, online boosting holds promise for enhancing predictive performance and adaptability to evolving data distributions.

Here, we will consider the framework of sequential non-parametric regression with arbitrary data pairs (x_t, y_t) (as in Rakhlin and Sridharan [7] or Cesa-Bianchi and Lugosi [3]). In particular, we will analyze a Boosting algorithm, discussing its convergence guarantees compared to the optimal rates (minimax) already established, for example, in Rakhlin and Sridharan [7].

Key words. Statistical Learning, Online Learning, Boosting, Ensemble Learning, Nonparametric Regression.

1 Online Nonparametric Regression

1.1 Setting

Online nonparametric regression refers to a learning framework where a model is trained sequentially on streaming data to predict an output variable without assuming any specific functional form for the underlying relationship between inputs and outputs. Unlike traditional regression methods, which estimate parameters of a predefined model using all available data at once, online nonparametric regression updates the model continuously as new data points arrive. This adaptive learning process allows the model to capture complex and evolving patterns in the data without requiring strong assumptions about its structure. Online nonparametric regression is particularly well-suited for applications where data is generated sequentially or where the underlying relationship between variables may change over time.

We consider that pairs of data $(x_1, y_1), \ldots, (x_t, y_t), \cdots \in \mathcal{X} \times \mathcal{Y}$ arrive in a stream and we are tasked with sequentially predicting each next response y_t given the current x_t and the past data $\{(x_s, y_s)\}_{s=1}^{t-1}$. Let $\hat{y}_t$ be our prediction and let the quality of this forecast be evaluated via ℓ_t (e.g. the square loss $\ell_t(\hat{y}_t, y_t) = (\hat{y}_t - y_t)^2$). More formally, the scenario can be formalized as follows:

Learning Scheme

For each round $t = 1, \ldots, T$, the learner or algorithm

- observes an input $x_t \in \mathcal{X}$
- makes a prediction $\hat{y}_t \in \mathcal{Y}$
- observes the true output $y_t \in \mathcal{Y}$
- measures loss by $\ell_t(\hat{y}_t, y_t)$
- updates his prediction rule

1.2 How to make predictions with weak learners?

For each $t = 1, \ldots, T$, we build our predictions by combining $K \geq 1$ *sequential* predictors $(f_{k,t})_t, k \in \{1, \ldots, K\}$ from a class of *weak learners*

$$\mathcal{W} = \{x \mapsto f(x; \theta, I) : \theta \text{ parameter of } f \text{ with support } I\} \subset \mathcal{Y}^{\mathcal{X}},$$

e.g. the set of regression trees with (low) depth 1, $\mathcal{W}_1 = \{f(\cdot; \theta, I) : \theta \in \mathbb{R}^2 \text{ and } I = (I^{(1)}, I^{(2)}), I^{(1)} \sqcup I^{(2)} = \mathcal{X}\}$.

Let $K \geq 1$ and $(f_1, \ldots, f_K) \in \mathcal{W}^K$. A prediction of order K (i.e. using $f_1, \ldots, f_K$) will be, at any time $t \geq 1$,

$$\hat{y}_t = F_{K,t}(x_t) = \sum_{k=1}^{K} f_{k,t}(x_t),$$

using the *strong estimator* $F_K = \sum_{k=1}^{K} f_k$ belonging to the class

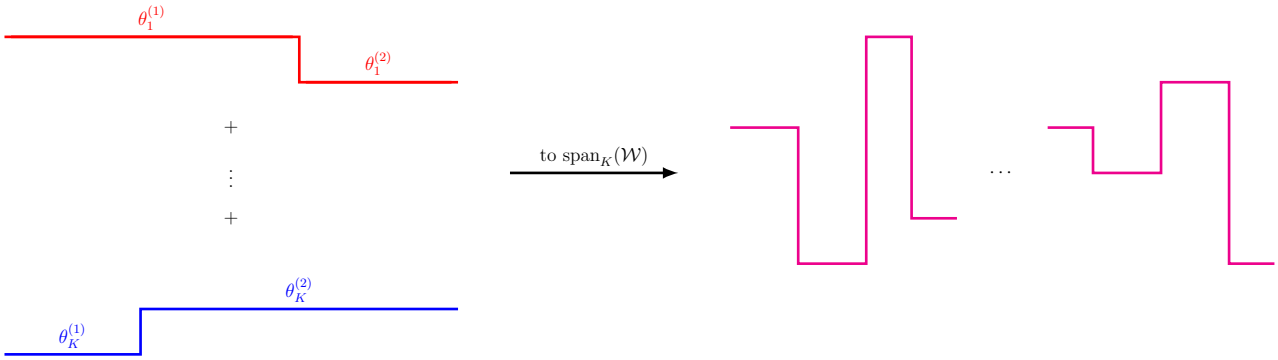
$$\text{span}_K(\mathcal{W}) := \left\{ F_K = \sum_{k=1}^{K} f_k : f_k \in \mathcal{W} \right\}.$$

Why several f_k? What is their goal? A weak learner, as the name suggests, is too weak to make accurate predictions on its own. Therefore, it needs to rely on its peers. Considering a mixture of K weak learners, each estimator f_k favors learning from the errors (alias residuals) of $K - 1$ others and ensures to correct what it can. The collective learning of these weak learners leads to the formation of a *strong learner* capable of making high-quality predictions.

Example: Combining K predictors in $\mathcal{W}_1 = \{\text{uniform regression trees of depth 1}\}$ which is the set of 2-piecewise constant functions with random cut will lead to a strong predictor in

$$\text{span}_K(\mathcal{W}) \subset \left\{ F_{K'} : F_{K'}(x) = \sum_{k=1}^{K'} \theta_k \mathbf{1}_{x \in J_k}, \theta_k \in \mathbb{R}, \bigsqcup_{k=1}^{K'} J_k = \mathcal{X} \right\} =: \mathcal{F}_{K'},$$

where $\mathcal{F}_{K'}$ is the space of functions that are constant on $K' = 2K + 1$ intervals. Here are some illustrations:



1.3 How to measure the performance?

In traditional statistical (and batch) learning theory, the goal is to build a predictor or estimator $\hat{f}_{1:T}$ using a full batch of T data and which minimizes the empirical risk

$$\frac{1}{T}\sum_{t=1}^{T}\ell_t(\hat{f}_{1:T}(x_t), y_t).$$

First problem is that we do not have access to the whole dataset to build such a $\hat{f}_{1:T}$. Second, if the environment chooses large losses ℓ_t for all decisions and time t, it is then impossible for the learner to ensure a small cumulative loss. Therefore, one needs a relative criterion: the **regret** of the learner defined as the difference between the cumulative loss he incurred and that of the best fixed decision in hindsight.
For a given time horizon $T \geq 1$, and a sequence of losses (ℓ_t) the problem of regression is then formulated as that of minimizing the regret

$$\mathrm{Reg}_T(\mathcal{F}) := \sum_{t=1}^{T}\ell_t(\hat{y}_t, y_t) - \inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\ell_t(f(x_t), y_t) \tag{1}$$

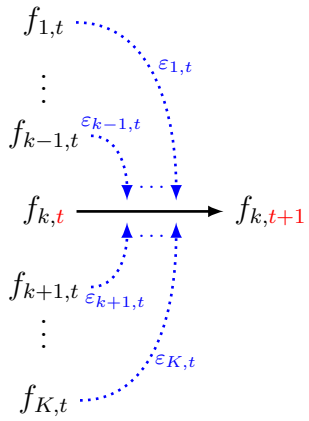
with respect to some benchmark class of non-parametric functions $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ (e.g. the space of Lipschitz functions).
We assume that the losses (ℓ_t) are differentiable and convex in their first argument (in $\hat{y}_t$). The problem is now to optimize in each predictor f_k, so we can rewrite $\ell_t : \mathrm{span}_K(\mathcal{W}) \times \mathbb{R} \to \mathbb{R}$ as a function $\ell_t : \mathcal{W}^K \to \mathbb{R}$ such that

$$\ell_t(f_{1,t}, \ldots, f_{K,t}) = \ell_t\left(\sum_{k=1}^{K} f_k(x_t), y_t\right).$$

2 Analysis of an Online Boosting Algorithm

2.1 Architecture of the Algorithm

Let's begin by introducing the algorithm designed to address this problem:



Algorithm 1: Online Gradient Boosting

for $t = 1$ **to** T **do**
- Receive data x_t;
- Predict $\hat{y}_t = \hat{F}_{K,t}(x_t) = \sum_{k=1}^{K} \hat{f}_{k,t}(x_t)$;
- Reveal residuals $\varepsilon_{k,t}$, gradients $g_{k,t} = \nabla_{\hat{f}_{k,t}} \ell_t(\hat{f}_{1,t}, \ldots, \hat{f}_{K,t})$ for all $k = 1, \ldots, K$ and incur $\ell_t(\hat{y}_t, y_t)$;
- **for** $k = 1$ **to** K **do**

$$\hat{f}_{k,t+1} \leftarrow \texttt{gradient\_step}(\varepsilon_{k,t}, g_{k,t}) \tag{2}$$

Return: $\hat{F}_{K,T} = \sum_{k=1}^{K} f_{K,t}$

$\varepsilon_{k,t}$ are called the *residuals* or *errors* of each weak learner $f_{k,t}$ at time t. At any time $t = 1, \ldots, T$, each weak learner $f_{k,t}$ aims to best correct the *residuals* $\varepsilon_{1,t}, \ldots, \varepsilon_{K,t}$ of $\{f_{1,t}, \ldots, f_{K,t}\} \backslash \{f_{k,t}\}$.

2.2 How to bound the regret (1)?

We can decompose the regret (1) as a sum of the following 2 stage regrets:

$$R_T^{(1)} = \sum_{t=1}^{T} \ell_t(\hat{f}_{1,t}, \ldots, \hat{f}_{K,t}) - \inf_{f_1, \ldots, f_K \in \mathcal{W}} \sum_{t=1}^{T} \ell_t(f_1, \ldots, f_K) \tag{3}$$

$$R_T^{(2)} = \inf_{f_1, \ldots, f_K \in \mathcal{W}} \sum_{t=1}^{T} \ell_t(f_1, \ldots, f_K) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \ell_t(f(x_t), y_t) \tag{4}$$

Assume we consider an online version of gradient descent, *online gradient descent (OGD)* (introduced in [8], and which can be applied to any convex and differentiable loss function), to perform (2) sequentially for any $k = 1, \ldots, K$ and any $t \geq 1$ as

$$\hat{f}_{k,t+1} \leftarrow f_{k,t} - \eta_{k,t} g_{k,t} \tag{5}$$

Algorithm (1) verifies the following result for bounding $R_T^{(1)}$:

Theorem 2.1. *Assume losses $\ell_t : \mathbb{R}^K \to \mathbb{R}$ are differentiable and σ_k-strongly convex in each coordinate $k = 1, \dots, K$. Also assume that for any $k = 1, \dots, K, \sup_t |\nabla_{\hat{\theta}_{k,t}} \ell_t| \leq G_k$. Then, Algorithm 1 with (5) and $\eta_{k,t} = \frac{1}{\sigma_k t}$ has the following regret:*

$$\sum_{t=1}^{T} \ell_t(\hat{f}_{1,t}, \dots, \hat{f}_{K,t}) - \inf_{f_1,\dots,f_K \in \mathcal{W}} \sum_{t=1}^{T} \ell_t(f_1, \dots, f_K) \leq \sum_{k=1}^{K} \frac{G_k^2}{2\sigma_k} \ln(T_k + 1) \leq \frac{G^2}{2\sigma} K \ln(T+1) \quad (6)$$

with $T_k = |\{t : x_t \in I_k\}|, G = \sup_{1 \leq k \leq K} G_k$ and $\sigma = \inf_{1 \leq k \leq K} \sigma_k$.

Additionally, we will explore other minimization procedure step (2) within Algorithm 1.

Remember that our goal is to establish a bound for regret (1), necessitating an analysis of the second regret outlined in (4). Specifically, we introduce a *boosting/learning condition* for weak learners in $\mathcal{W}$ with respect to an appropriate class of functions $\mathcal{G}$ (e.g. the set of piecewise constant functions for weak learners in $\mathcal{W}_1$).

Assumption 2.1. *For any $k \geq 1, \exists \rho_k = \rho_k(\mathcal{W}, \mathcal{G}) \in [0, 1]$ such that*

$$\inf_{f_1,\dots,f_{k+1} \in \mathcal{W}} \sum_{t=1}^{T} \ell_t(f_1, \dots, f_k, f_{k+1}) - \inf_{g \in \mathcal{G}} \sum_{t=1}^{T} \ell_t(g(x_t), y_t) \leq \rho_k \left(\inf_{f_1,\dots,f_k \in \mathcal{W}} \sum_{t=1}^{T} \ell_t(f_1, \dots, f_k) - \inf_{g \in \mathcal{G}} \sum_{t=1}^{T} \ell_t(g(x_t), y_t) \right) \quad (7)$$

The latter stems from a common assumption prevalent in batch scenarios, notably introduced and scrutinized by Jiang [6] within both regression and classification contexts.

This hypothesis will serve us in managing the second term of regret $R_T^{(2)}$ (4) recursively as we advance through k. Following this, it becomes necessary to break down $R_T^{(2)}$ into a regret against a suitable class of functions $\mathcal{G}$ (e.g. $\mathcal{G} = \mathcal{F}_{K'}$), ensuring the existence of a learning coefficient $\rho(\mathcal{W}, \mathcal{G})$.

3 Experiments

Consider the standard regression problem,

$$\forall 1 \leq t \leq T, \quad y_t = g(x_t) + W_t$$

where $W_t \sim \mathcal{N}(0, \sigma^2), \sigma > 0, g : \mathbb{R} \to \mathbb{R}$ and $\mathcal{X} = [0, 1]$.

In experiments below, we aim to reconstruct $g : x \mapsto \cos(3\pi x) - \sin(3x)$ using $K = 10$ weak learners with random supports (of type given in the example) and we define ℓ_t as the square loss function.
Recall that $\mathcal{F}_{K'}$ represents the set of functions that remain constant across $K' = 2K+1$ intervals. Below, we depict the progression of $\text{Reg}_T(\mathcal{F}_{K'})$ over time, indicating the regret of our algorithm employing regression trees compared to the optimal piecewise constant functions of similar order. Additionally, we present in a separate graph the final prediction generated by our algorithm, juxtaposed with that of its competitor.

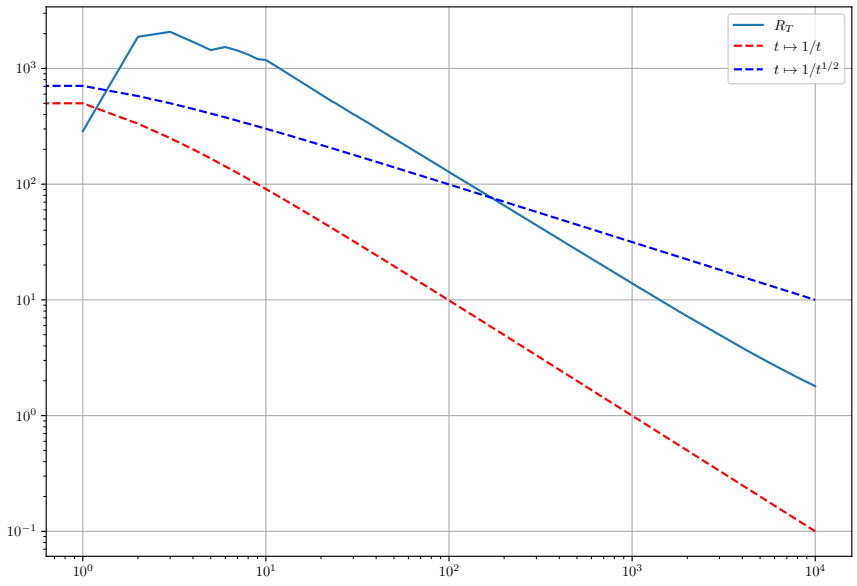


Figure 1: $\mathrm{Reg}_T(\mathcal{F}_{K'})/T$ as a function of T.

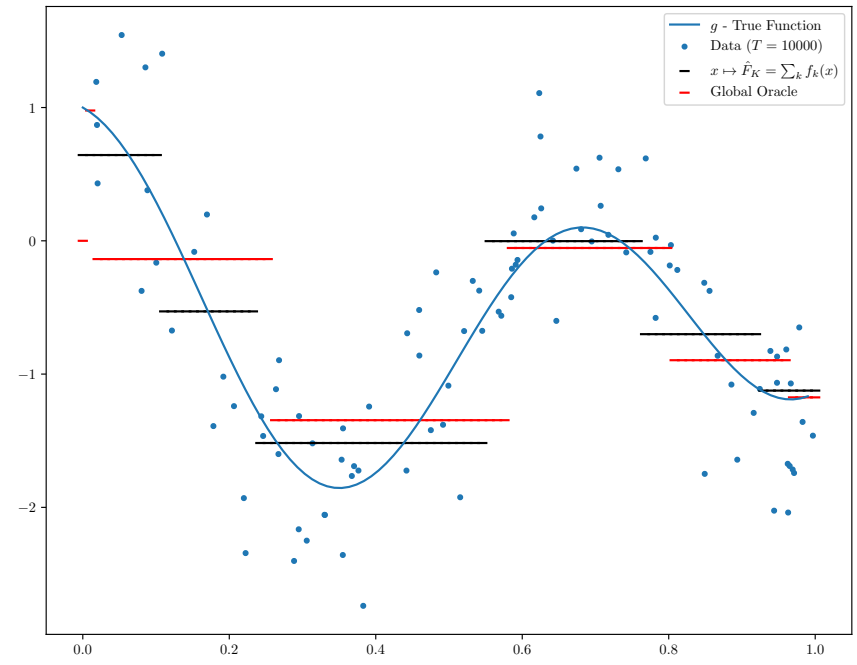


Figure 2: Final prediction

References

[1] A. Beygelzimer, E. Hazan, S. Kale, and H. Luo. Online gradient boosting. *Advances in neural information processing systems*, 28, 2015.

[2] N. Brukhim and E. Hazan. Online boosting with bandit feedback. In *Algorithmic Learning Theory*, pages 397–420. PMLR, 2021.

[3] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[4] Y. Freund, R. Schapire, and N. Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[5] P. Gaillard and S. Gerchinovitz. A chaining algorithm for online nonparametric regression. In *Conference on Learning Theory*, pages 764–796. PMLR, 2015.

[6] W. Jiang. On weak base hypotheses and their implications for boosting regression and classification. *The Annals of Statistics*, 30(1):51–73, 2002.

[7] A. Rakhlin and K. Sridharan. Online non-parametric regression. In *Conference on Learning Theory*, pages 1232–1264. PMLR, 2014.

[8] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.