Boosting Diversity in Regression Ensembles

Mathias Bourel, Jairo Cugliari, Yannig Goude, Jean-michel Poggi

Boosting Diversity in Regression Ensembles

Mathias Bourel[a], Jairo Cugliari[b], Yannig Goude[c], Jean-Michel Poggi[d]

[a]*IMERL, Facultad de Ingeniería, Universidad de la República, Uruguay*
[b]*Université de Lyon, Lyon 2, ERIC ER 3083, 5 avenue Pierre Mendès-France, F69676, Bron Cedex, France*
[c]*EDF R&D, & LMO, Université Paris-Saclay, France*
[d]*Université Paris & LMO, Université Paris-Saclay, France*

Abstract

The practical interest of using ensemble methods has been highlighted in several works. Aggregating predictors leads very often to improve the performance of a single one. A fruitful recipe is to generate several predictors from a single one by perturbing the learning set and, instead of selecting the best one, to aggregate them. Bagging, boosting and Random forests are examples of such strategies useful both for classification and regression problems. A key ingredient to properly analyse the improvement of prediction performance is the diversity of the predictors ensemble.

In the regression case, aggregation is mainly interested on how to generate individual predictors to improve quadratic prediction performance. We look for enhancing these methods by using the concept of diversity (also known as negative correlation learning). We propose an algorithm to enrich the set of original individual predictors using a gradient boosting-based method by incorporating a diversity term to guide the gradient boosting iterations. The idea is to progressively generate predictors by boosting diversity, this modification induces some kind of suboptimality of the individual learners but improve the ensemble. Then, we establish a convergence result ensuring that the associated optimisation strategy converges to a global optimum.

Finally, we show by means of numerical experiments the appropriateness of our procedure and examine not only the final predictor or the aggregated one but also the generated sequence. First, on a simulated dataset, we illustrate and study the method with respect to the family of predictors as well the parameters to be tuned (diversity weight and gradient step). Second, real-world electricity demand datasets are considered opening the application of such ideas to the forecasting context.

Keywords: Boosting, Diversity, Ensemble, Regression, Trees

1. Introduction

The practical interest of using ensemble methods has been highlighted in several works [28, 39, 45]. Ensemble methods are now used in very different domains: biology [5, 46], medecine [24, 51], electricity management [15, 26, 47], computer vision [27], physics [1], finance [14], ecology [6], insurance [34] or environmental sciences [13, 48]. Ensemble methods are also very popular for machine learning challenges, recent software libraries based on ensemble gradient boosting methods such as `XGBoost` [17], `CatBoost` [40], `LightGBM` [31] are widely used in that context.

The general idea is to build a better learner for a task by assembling several individual or base learners. Either for classification or regression tasks, these ensemble methods have proven their efficacy by controlling at least one of the two components of the classical error decomposition of the error between a variance and a bias term. An important question is how to choose and manage the different base learners. When using tree-based learners, two well-known illustrations of ensemble methods are Bagging for Bootstrap and AGGregatING [7] and Random Forests [8]. In both cases the creation of the base learners involves adding controlled randomness to produce, to some extent, independent learners.

Boosting techniques are iterative methods that consist in improving the performance of several hypothesis or base predictors of the same nature, combining them and reweighting at each step the original data sample. Freund and Schapire described Adaboost, the first boosting algorithm designed for binary classification problems and with classification trees as hypothesis [19]. Various types of extensions for boosting exist, in particular for multiclass classification and for regression and they use different approaches [45].

For a given machine learning method and/or a given set of experts, it is necessary to quantify a kind of "diversifiability" notion and it could be related to the notion of weak "learnability". The capability of a given estimation method to generate by boosting sufficiently diverse models or in other words the conditions to check for the choice of a design method to generate models, has been mainly studied in the classification case. Nevertheless, if in the classification case weak learnability is well defined, this remains essentially to be done for regression problems. A related issue, is instability. Following [30, p. 505-506], the control of bias and variance, and, hence, generalization error, is related to the idea of instability. A predictor design method is unstable if a small perturbation of the learning set may induce important changes in the resulting predictor. The instability of a predictor (or of a learning algorithm) can be used to improve accuracy. For example, by using resampling, to

stabilize a given method, as in Bagging to reduce variance and Boosting to reduce bias. Focusing on conditions to make Adaboost effective, [44] identifies some desirable properties of an effective weak learning algorithm and highlights, in the classification context, two main aspects: diversity and coverage (approximation property). Reyzin claims that with high coverage, diversity is easier to achieve while to allows diversity, and again that instability is required. We use the concept of diversity [11, 43] to propose a new algorithm to enrich the set of original individual predictors. The formulation is inspired from the Negative Correlation Learning for neural networks [35]. The significance of the Ambiguity decomposition it that the error of the mixture will be less than or equal to the average error of the individuals, and then the ensemble has lower error than the average individual error: for sufficiently accurate predictors, the larger the diversity term, the larger the reduction of ensemble error. We modify the usual L^2 cost function with the aim to find a good predictor that will be at the same time "diverse" than the mean of the predictors founded at the precedent steps, according to the diversity formula. Of course this modification induces some kind of suboptimality for the individual learner. However, the hope is that the ensemble will be benefited since the greater imposed diversity is expected to more than compensate the reduction of optimality for the individual learners.

The paper is organised as follows. After this introduction, Section 2 is devoted to methods. We first recall the usual boosting framework, then boosting diversity is defined before proving a convergence result. Section 3 contains numerical experiments. Since the methodological previous section provides a general framework, to implement it, we start by recalling the base learners considered in the sequel, from simple trees to Random forests. Then, we show by means of numerical experiments the appropriateness of our boosting diversity procedure using simulated data and real-world electricity demand datasets. Finally, Section 4 provides a short conclusion and a discussion on perspectives.

2. Methods

2.1. Boosting in the general case

In the context of machine learning methods, boosting are sequential algorithms that estimate a function $F : \mathbb{R} \to \mathbb{R}$ by minimising $C(F) = \mathbb{E}[\Psi(Y, F(X)]$, the expectation of a functional $\Psi : \mathbb{R} \times \mathbb{R} \to [0, +\infty)$ that measures the cost committed for predicting $F(X)$ instead of Y, using a training sample $\{(y_i, \mathbf{x}_i)\}_{i=1}^n$ and functional gradient descent techniques. More precisely, considering a family of functions $\mathcal{F} = \{f : \mathbb{R}^d \to \mathbb{R}\}$, the method consists in estimate F by minimisation of the the empirical expectation loss

$$C_n(F) = \frac{1}{n}\sum_{i=1}^{n} \Psi(y_i, F(\mathbf{x}_i)),$$

by looking for an additive function of the form $F_M = \sum_{m=1}^{M} \alpha_m f_m$ where $\alpha_m \in \mathbb{R}$ and $f_m \in \mathcal{F}$ for all $m = 1, \ldots, M$ [12, 20, 37]. Examples of cost functions are:

1. *Exponential cost function* or *Adaboost* cost function [19] : $\Psi(y, F) = \exp(-yF)$ if $y \in \{-1, 1\}$ for classification, with population minimizer $\frac{1}{2}\log\left(\frac{\mathbb{P}(Y=1|X=\mathbf{x})}{\mathbb{P}(Y=-1|X=\mathbf{x})}\right)$,
2. *Logit cost function* [21] : $\Psi(y, F) = \log_2(1 + e^{-2yF})$ if $y \in \{-1, 1\}$ for classification with population minimizer $\frac{1}{2}\log\left(\frac{\mathbb{P}(Y=1|X=\mathbf{x})}{\mathbb{P}(Y=-1|X=\mathbf{x})}\right)$,
3. L^2 *cost function* [21] : $\Psi(y, F) = \frac{1}{2}(y - F)^2$ if $y \in \mathbb{R}$ for regression, with population minimizer $\mathbb{E}(Y|X = \mathbf{x})$.

Let $\mathcal{L} = \{(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)\}$ be a sample and $\mathcal{F}$ a family of functions.

1. Fit an initial learner $\widehat{F}_0 \in \mathcal{F}$ such that
   $\widehat{F}_0 = \underset{f\in\mathcal{F}}{\text{Argmin}} \sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2$
2. For $m \in \{1, \ldots, M\}$:
   (a) Compute the negative gradient
   $u_i = -\left.\frac{\partial\Psi(y_i,F)}{\partial F}\right|_{F=\widehat{F}_{m-1}(\mathbf{x}_i)} \quad \forall\, i = 1, \ldots, n$ and
   $\widehat{f}_m = \underset{f\in\mathcal{F}}{\text{Argmin}} \sum_{i=1}^{n}(u_i - f(\mathbf{x}_i))^2$
   (b) Choose the best step size
   $\widehat{w}_m = \underset{w}{\text{Argmin}} \sum_{i=1}^{n} \Psi(y_i, \widehat{F}_{m-1}(\mathbf{x}_i) + w\widehat{f}_m(\mathbf{x}_i))$
   (c) Update the aggregated predictor:
   $\widehat{F}_m(x) = \widehat{F}_{m-1}(\mathbf{x}) + \widehat{w}_m\widehat{f}_m(\mathbf{x})$

Outputs: a family of experts $\widehat{f}_1, \widehat{f}_2, \ldots, \widehat{f}_M$ and the aggregated predictor $\widehat{F}_M$.

Figure 1: General Gradient Boosting method.

Searching a functional F over the linear span $lin(\mathcal{F})$ was studied mainly in [21]. At step m using the Taylor approximation $C(F_m) - C(F_m + wf) \approx -w\langle\nabla C(F_m), f\rangle_{\mu_X}$, instead of usual searching about a function $f \in \mathcal{F}$ maximizing $-\langle\nabla C(F_m), f\rangle_{\mu_X}$, where μ_X is the distribution of X and $L^2(\mu_X)$ is the set of all measurable functions such that $\int f^2 < \infty$. The empirical formulation is $\underset{f\in\mathcal{F}}{\text{Argmax}}\left\{-\frac{1}{n}\sum_{i=1}^{n}\nabla C(F_m)(\mathbf{x}_i)f(x_i)\right\}$ and we look at a least squares approximation in a class of functions $\mathcal{F}$ such that $f_{m+1} \in \underset{f\in\mathcal{F}}{\text{Argmin}}\left\{\|-\nabla C(F_m) - f\|^2_{\mu_X}\right\}$ and in the empirical setting $f_{m+1} \in \underset{f\in\mathcal{F}}{\text{Argmin}}\left\{\frac{1}{n}\sum_{i=1}^{n}(-\nabla C(F_m)(\mathbf{x}_i) - f(\mathbf{x}_i))^2\right\}$. Assuming that $\Psi(y, \cdot)$ is convex and continuously differentiable it is straightforward that $\nabla C(F_m)(\mathbf{x}_i) = \Psi_x(y_i, F_m(\mathbf{x}_i))$. General algorithm is given in Figure 1. The context of our work is in regression, with an adaptation of the L^2-cost function. In this case as $\Psi(y, x) = \frac{1}{2}(y - x)^2$ then $\nabla C(F_m)(\mathbf{x}_i) = y_i - F_m(\mathbf{x}_i)$ and the algorithm fits f_{m+1} to the residuals $y_i - F_m(\mathbf{x}_i)$, the classical residual vector [12, 21].

2.2. Boosting based on diversity decomposition

In the spirit of L^2-boost we propose a new algorithm which encourage diversity of intermediate predictors. It is based on

the negative correlation learning (NCL) framework [35] which considers cooperation and interaction among the ensemble learners. If we consider $f_1, \ldots, f_M$ different predictors and we denote the aggregated predictor as $F^*(\mathbf{x}_i) = \sum_{m=1}^{M} p_m f_m(\mathbf{x}_i) = \widehat{y_i}$, the *Diversity Formula* or *Ambiguity Decomposition* is:

$$\widehat{(y_i} - y_i)^2 = \underbrace{\sum_{m=1}^{M} p_m (f_m(\mathbf{x}_i) - y_i)^2}_{\substack{\text{weighted average error} \\ \text{of the individuals}}} - \underbrace{\sum_{m=1}^{M} p_m (f_m(\mathbf{x}_i) - \widehat{y_i})^2}_{\text{diversity term}}$$

where $\sum_m p_m = 1, p_m \geq 0$. The first term corresponds to the weighted average error of individual predictors and the second is the diversity term, $\sum_{m=1}^{M} p_m (f_m(\mathbf{x}_i) - F^*(\mathbf{x}_i))^2$ in equation 2.2, measuring the variability around the aggregated predictor. As said in [11] and in [10], the significance of the Ambiguity decomposition it that the error of the mixture will be less than or equal to the average error of the individuals, and then the ensemble has lower error than the average individual error: larger will be the diversity term, larger will be the ensemble error reduction. In a simplified framework with uniform weights $p_m = 1/M$, in [42], the authors introduced the following loss function

$$\Psi_\kappa(\mathbf{x}, y) = \frac{1}{M} \sum_{m=1}^{M} (f_m(\mathbf{x}) - y)^2 - \frac{\kappa}{M} \sum_{m=1}^{M} (f_m(\mathbf{x}) - f_m^*(\mathbf{x}))^2,$$

where $\kappa \geq 0$, $f_1, \ldots, f_M$ are regression models and $f_m^*(\mathbf{x}) = \frac{1}{m} \sum_{k=1}^{m} f_k(\mathbf{x})$. As it is possible to rewrite Ψ_κ as

$$\Psi_\kappa(\mathbf{x}, y) = (1 - \kappa) \frac{1}{M} \sum_{m=1}^{M} (f_m(\mathbf{x}) - y)^2 + \kappa (f^*(\mathbf{x}) - y)^2,$$

varying κ from 0 to 1 implies training each individual predictors independently through training the ensemble. Using this last formula, they prove that if $\kappa \in [0, 1]$ the average loss is non negative, but if $\kappa > 1$ it is arbitrarily negatively high. In this work, we modify the usual L^2-cost function with the aim to find a good predictor that will be at the same time "diverse" than the mean of the predictors founded at the precedent steps, according to the diversity formula. According to equation above, we propose as new cost function

$$\Psi(y, F) = \frac{1}{2}(y - F)^2 - \frac{\kappa}{2}(F - \overline{F})^2, \tag{1}$$

where κ is the parameter which modulate the importance given to the diversity of the predictor to $\overline{F}$, the average of the previous one with F. As $\overline{F}$ can be write as $\overline{F} = c + \alpha F$, cost function is

$$\Psi(y, F) = \frac{1}{2}(y - F)^2 - \frac{\kappa}{2}((1 - \alpha)F - c)^2,$$

where c is the average of the previous one and can be thought as a constant. In fact if $\overline{F}$ is the average of M predictors, then $\alpha = \frac{1}{M}$.

Boosting **Di**versity algorithm `BoDi` is detailed in Figure 2. With `BoDi` algorithm, as in the classical boosting, we obtain two final ensemble forecasts $F^*_{M,\kappa}$ and $F_{m,\kappa}$. Here we make the dependency to κ explicit whereas other parameters (like the gradient step, the size of the bootstrap sample) play a role. Notice that the relative weight of the terms in the new loss function varies with the iteration cycles. Consequently, it also does on the gradient direction. We make explicit this fact by notating κ_m the factor that multiplies the second term. Indeed, after m iterations κ_m vanishes and thus the `BoDi` loss function converges to the L2-Boost loss function. However, notice that a genuine gain appears before the convergence as we shown through numerical experiments (see Section 3).

2.3. Convergence of the algorithm

A recent and very elegant result from [3] proves the convergence of several gradient boosting-based methods in a very general framework. The result is stated in what follows, for assumptions discussed below.

Theorem 1. *If assumptions A_1 to A_3 hold true and if $0 < \delta < 1/(2L)$ where L is the constant of A_3 and δ the step of the Boosting Diversity Algorithm, then*

$$\lim_{t \to \infty} \mathbb{E}(\Psi(y, F_t)) = \inf_{F \in lin(\mathcal{F})} \mathbb{E}(\Psi(y, F))$$

where $\mathcal{F}$ is the family of functions we reach.

This result warranties that the optimisation strategy converges to a global optimum. However observe that this convergence result is not statistical.

The following three assumptions are required for the theorem to hold true:

A_1: C is convex and locally bounded where $C(F) = \mathbb{E}(\Psi(Y, F(X))$.

A_2: For all y $\Psi(y, \cdot)$ is $\lambda-$strongly convex[1] and then C is λ-strongly convex on $L^2(\mu_X)$ where μ_X is the distribution of X and $L^2(\mu_X)$ is the set of all measurable functions such that $\int f^2 < \infty$.

A_3: For all y the function $\Psi(y, \cdot)$ is continuously differentiable and there exists a constant $L > 0$ such that for all $(x_1, x_2) \in \mathbb{R}^2$ and for all y:

$$\left|\Psi_x(y, x_1) - \Psi_x(y, x_2)\right| \leq L|x_1 - x_2|.$$

If we require that $\kappa < \frac{1}{1-\alpha}$, then the result holds for the expectation of our convex cost function $C(F) = \mathbb{E}(\Psi(Y, F(X))$ where $\Psi(y, F) = \frac{1}{2}(y - F)^2 - \frac{\kappa}{2}((1 - \alpha)F - c)^2$. We now prove that C and Ψ satisfy the three assumptions of [3] to ensure convergence established by Theorem 1.

[1] A function f is λ-strongly convex if for all $(x_1, x_2) \in \mathbb{R}^2$ and for all $t \in [0, 1]$ we have that $f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2) - \frac{\lambda}{2}t(1-t)(x_1 - x_2)^2$.

Let $\mathcal{L} = \{(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)\}$ be a sample, $\mathcal{F}$ a family of functions, $\kappa > 0$ and $\delta > 0$. Split the data in two disjoint parts $I = I_1 \cup I_2$

1. Fit an initial learner $\widehat{F}_1 \in \mathcal{F}$ over I_1 such that $\widehat{F}_1 = \underset{f \in \mathcal{F}}{\text{Argmin}} \sum_{i \in I_1} (y_i - f(\mathbf{x}_i))^2$. Set $\widehat{F}_1^*(\mathbf{x}) = \widehat{F}_1(\mathbf{x})$.
2. For $m \in \{2, \ldots, M\}$:
   (a) $\forall\, i \in I_2$, compute the negative diversity gradient of the cost function (see equation (1)) and evaluate it at $\widehat{F}_{m-1}(\mathbf{x}_i)$:
   $$u_i = (y_i - \widehat{F}_{m-1}(\mathbf{x}_i)) + \kappa_m \left(\widehat{F}_{m-1}(\mathbf{x}_i) - \widehat{F}^*_{m-1}(\mathbf{x}_i)\right)$$
   with $\kappa_m = \kappa\left(1 - \frac{1}{m-1}\right)$ if $m > 2$, $\kappa_2 = 1$ and $\widehat{f_m} = \underset{f \in \mathcal{F}}{\text{Argmin}} \sum_{i \in I_2} (u_i - f(\mathbf{x}_i))^2$.
   (b) Update boosting predictor as $\widehat{F}_m(\mathbf{x}) = \widehat{F}_{m-1}(\mathbf{x}) + \delta \widehat{f_m}(\mathbf{x})$, compute $\widehat{F}^*_m(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^{m} \widehat{F}_i(\mathbf{x})$ and update $I_2 = I \setminus I_1$ with a new bootstrap sample I_1 of I.

Outputs: a family of experts $\widehat{f_1}, \widehat{f_2}, \ldots, \widehat{f_M}$ and the aggregated predictors $\widehat{F}_M$ and $\widehat{F}^*_M$.

Figure 2: Boosting Diversity method. The split of the original sample is not mandatory, it is only to compute the Out-of-Bag error over a test sample (in our case I_1).

Proof. 1. Looking at the derivatives of $\Psi(y, F) = \frac{1}{2}(y-F)^2 - \frac{\kappa}{2}((1-\alpha)F - c)^2$ with respect to F, then $\frac{\partial^2 \Psi}{\partial F^2}(y, F) = 1 - (1-\alpha)\kappa > 0 \Leftrightarrow \kappa < \frac{1}{1-\alpha}$ and then C is convex when $\kappa < \frac{1}{1-\alpha}$ and it is locally bounded.

2. For each y, we rewrite $\Psi(y, F)$ as a polynomial in F of the form $p(F) = \Psi(y, F) = AF^2 + BF + C$. Then,

$$\begin{aligned}
&p(tx_1 + (1-t)x_2) \\
&\quad = A(tx_1 + (1-t)x_2)^2 + B(tx_1 + (1-t)x_2) + C \\
&\quad = At^2x_1^2 + Btx_1 + tC + A(1-t)^2x_2^2 \\
&\qquad + B(1-t)x_2 + (1-t)C + 2At(1-t)x_1x_2 \\
&\quad = t(tAx_1^2 + Bx_1 + C) + (1-t)((1-t)Ax_2^2 + Bx_2 + C) \\
&\qquad + 2At(t-1)x_1x_2 \\
&\quad \leq tf(x_1) + (1-t)f(x_2) - At(1-t)(-2x_1x_2) \\
&\quad \leq tp(x_1) + (1-t)p(x_2) - \frac{1-\kappa}{2}t(1-t)(x_1 - x_2)^2
\end{aligned}$$

   Thus, p is $(1-\kappa)-$ strongly convex and therefore C is $(1-\kappa)$-strongly convex in $L^2(\mu_X)$.

3. For all $(x_1, x_2) \in \mathbb{R}^2$ and for all y:

$$\begin{aligned}
&|\Psi_x(y, x_1) - \Psi_x(y, x_2)| \\
&\quad = \left|-(y - x_1) - \kappa(x_1 - c) - (-(y - x_2) - \kappa(x_2 - c))\right| \\
&\quad = \left|(1-\kappa)(x_1 - x_2))\right| = (1-\kappa)|x_1 - x_2|
\end{aligned}$$

   And then $L > 1 - \kappa$ fulfils the requirement.

□

For the locally bounded condition of C and then its continuity, we have that $\inf_{F \in lin(\mathcal{F})} C(F) = \inf_{F \in \overline{lin(\mathcal{F})}} C(F)$ and this infimum is unique because of the strong convexity of C, i.e there exists a unique function $F^* \in lin(\mathcal{F})$ such that $C(F^*) = \inf_{F \in lin(\mathcal{F})} C(F)$. Observe that in our context, condition $A1$ means that κ must be lower than $\frac{M}{M-1}$.

Typically, the set of functions $\mathcal{F}$ could be the collection of all binary trees using axis parallel splits with k terminal nodes. In this case, each f of $\mathcal{F}$ can be written as $f(x) = \sum_{j=1}^{k} \mathbb{1}_{(x \in A_j)} a_j$. and the problem of minimizing $C(F)$ the quantity over the linear combinations of square-integrable functions of $\mathcal{F}$ is well-posed.

3. Numerical experiments

In this section, we will first introduce different tree-based base learners from very weak to quite complex ones (from trees with two leaves to Random forests) well suited to be used for boosting. Then, a simulation study focuses on booting diversity method and examines the influence of the diversity weight, the gradient step and the base learner. Finally, we evaluate the performance of our algorithm on a real data set as well as the gain with respect to classical boosting.

3.1. Learners

As mentioned in [18], boosting was originally proposed as a means for improving the performance of "weak learners" in binary classification problems. This amounts to building a model by repeatedly fitting a regression tree to the residuals. Importantly, the tree is typically quite small, involving a small number of splits, it is indeed a weak learner. Hastie et al. [28] claim that larger trees allows to introduce higher-level interaction effects among the input variables X, and then enlarge the approximation capability. So trees proved to be very efficient and flexible base learners for classical boosting (see e.g. [17] and [31] for recent powerful and highly used implementations, and [41] for a variant using boosting).

Let $\mathcal{L} = \{(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)\}$ be a learning set, consisting on n independent copies of $(y, \mathbf{x})$ supposed to be such that

$$y = f(\mathbf{x}) + \varepsilon,$$

with $E[\varepsilon|\mathbf{x}] = 0$, $\mathbf{x} = (x^1, \ldots, x^p), \mathbf{x} \in \mathbb{R}^p$ the explanatory variables and $y \in \mathbb{R}$ the response variable. In this context, f is the regression function and Random forests as well as CART trees build models providing estimators $\widehat{f}$ of f.

3.1.1. CART trees

CART [9] is a simple method to estimate f with respect to the mean square risk function, using decision trees. A regression tree is a piecewise constant function, where the splits are parallel to the original axes defined by the explanatory variables, of the following form:

$$\widehat{f_{CART}}(\mathbf{x}) = \sum_{k=1}^{K} \mathbb{1}_{(\mathbf{x}\in A_k)} a_k. \tag{2}$$

The constant value a_k in each cell A_k of the partition (or equivalently each leaf of the tree) is equal to the mean value of the response variable for the observations of the learning sample lying in A_k, that is $\frac{1}{n_k} \sum_{i:\, \mathbf{x}_i \in A_k} y_i$ with $n_k = \#\{i : \mathbf{x}_i \in A_k\}$. The construction is in two steps. A maximal tree is first obtained using splitting rules of the form $(x^j < t)$, by recursive partitioning of the input space. Since maximal trees may be very complex (too deep) and generally overfit the learning data therefore, a pruning is then performed leading to an optimal parsimonious tree.

Suboptimal trees can be considered by limiting the depth. Stumps are simple trees (of depth 1, defined by a single split), not interesting when considered alone but aggregating ensemble of such trees can be of useful to study the action of boosting diversity.

3.1.2. Random Forests

Breiman [8] proposed Random forests (abbreviated RF in the sequel) to improve a single tree and stabilize the method. It consists of aggregating a set of random trees, built over n_{tree} bootstrap samples $\mathcal{L}^1, \ldots, \mathcal{L}^{n_{tree}}$ of the training set $\mathcal{L}$. The trees of a RF are similar to CART trees but with two differences, leading to speed up the computations while preserving statistical performance. First, at each node, a fixed number of variables is randomly picked to determine the best split among them. Second, the trees are not pruned so all the trees of the forest are maximal trees. The resulting learning rule is the aggregation of all those trees, denoted by $\hat{f}_1, \ldots, \hat{f}_{n_{tree}}$. To make a prediction at a new point $\mathbf{x}$, the aggregation consists of taking the average prediction value

$$\widehat{f_{RF}}(\mathbf{x}) = \frac{1}{n_{tree}} \sum_{k=1}^{n_{tree}} \hat{f}_k(\mathbf{x}).$$

To speed up calculations and make easier the theoretical study of RF, mainly related to relax dependence of the splits on the learning sample, a lot of simplified RF have been introduced (see [4]). The simplest one is PRF (Purely RF) for which the splits of tree nodes (splitting variable and splitting value) are randomly drawn (for example uniformly) independently of the learning sample. This variant is here of special interest since it could clearly increase diversity.

In the sequel, we will compute the base-learners, stump, pure forest (PRF) and Random forests (RF) with the `R` packages `rpart` [49] for stumps and `ranger` [50] for both RF and PRF. The PRF learner is obtained using the `extratrees` option of `ranger` and for both RF and PRF the depth of the trees is set to unlimited.

In our experiments, we show that, to be able to generate some additional diversity by introducing an extra term measuring it, F has to be a more complex learner. We obtained satisfactory result choosing a Random forest as base learner.

3.2. Synthetic data

We use here a well-known simulated data set presented in [22] and already used for example in [7] to demonstrate the good performances of bagging. We use the `R` package `mlbench` of [33] to reproduce these data. The inputs are 10 independent variables uniformly distributed on the interval [0,1], only 5 out of these 10 are actually used. Outputs y are generated according to the formula:

$$y_i = 10 \sin(\pi x_{1,i} x_{2,i}) + 20(x_{3,i} - 0.5)^2 + 10 x_{4,i} + 5 x_{5,i} + \varepsilon_i,$$

where ε_i is $N(0, \sigma^2)$. As in [7] we simulated a learning set of size $n_0 = 200$ and a test set of size $n_1 = 1000$ observations, $\sigma = 1$. We replicate the simulation 100 times. The performance reported by Breiman's bagging predictor in terms of mean square error (MSE) on the test set is 6.2. The MSE reported by our Random forest learner in Figure 4 and Table 1 is about 7.1.

3.2.1. Influence of the diversity weight and the gradient step

Our first objective is to study the influence of the parameter κ which drives the diversity. The first experiment was conducted with the following inputs: base learner is a Random forest including all the 10 covariates with parameters $\mathsf{mtry} = 3$, $\mathsf{ntree} = 100$, data splitting rate $\alpha = 0.5$, gradient step $\delta = 0.08$ and diversity weight $\kappa \in \{0, 0.5, 1, 1.5\}$. The results are presented in Figure 3. For κ not too large there is a clear improvement of the diversity boosting strategy over the original Random forest learner, reducing the error by 3 after a sufficient number of iterations (at least 100). For large κ, here corresponding to 1.5 or more, the algorithm diverges after 100 iterations. In the range of reasonable values of $\kappa \leq 1$ ensuring convergence of the algorithm, we clearly see the interest of choosing κ close to 1 to encourage diversity as much as possible and improve the forecasting errors, $\kappa = 0$ corresponding to classical boosting. Curiously, even if the diversity boosting has been derived so that to improve the ensemble forecast $F^*_{m,\kappa}$ we observe that for all κ and $m < 100$ the error of $F_{m,\kappa}$ is lower than the error of $F^*_{M,\kappa}$. thus, diversity boosting could be used as a variant of classical boosting.

There is a link between the diversity weight κ and the gradient step. To illustrate that we consider a PRF base learner with again 100 trees. The mean MSE over 100 simulations of F and F^* along the boosting steps m are plotted on Figure 4. We clearly see that the best results are obtained with $\kappa = 0.9$ showing again the interest of encouraging diversity. We also

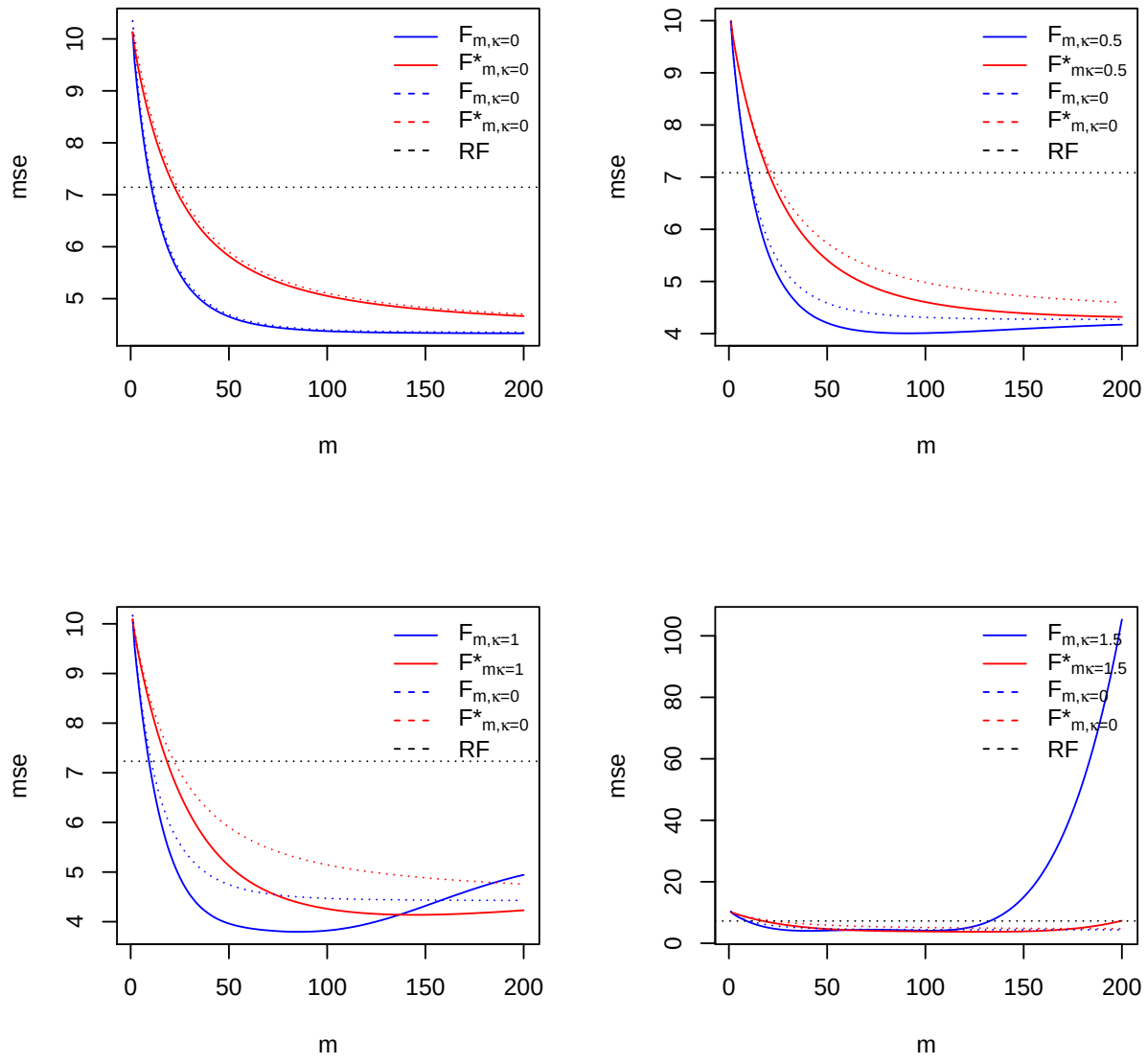

Figure 3: MSE as a function of boosting steps for $\kappa = 0, 0.5, 1, 1.5$ with Random forest (mtry=3, ntree=100) as base learner.

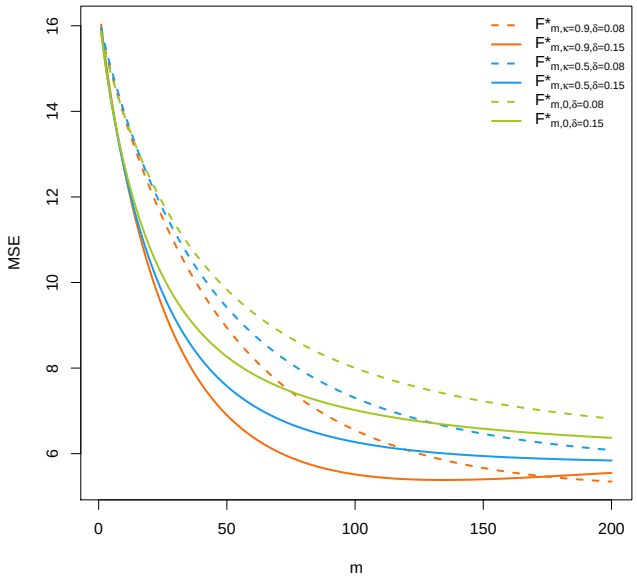

Figure 4: MSE as a function of boosting steps for different diversity values (κ) and gradient steps (δ) with PRF (ntree=100) as base learner.

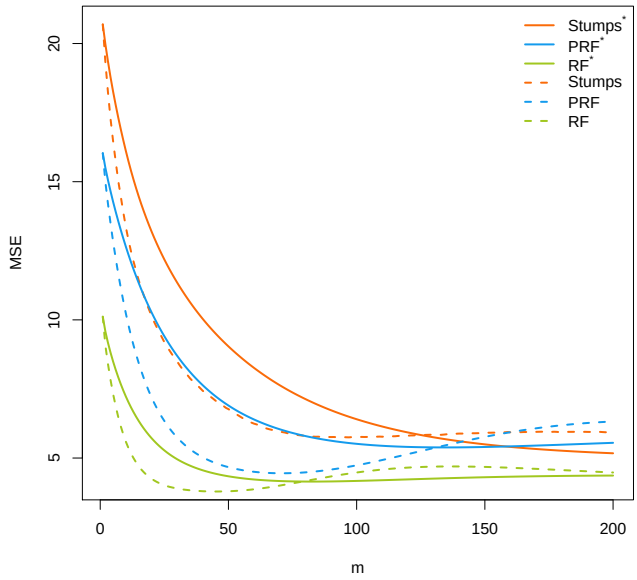

Figure 5: MSE as a function of boosting steps for 3 base learners: Stumps, PRF and RF.

see the influence of δ on the convergence speed of MSE curves similarly than in classical boosting. Choosing a small δ and a high number of boosting steps seems a good choice, especially when κ is large.

We can observe that classical boosting works well here and improves significantly the forecast of the original RF. This is also surprising as RF could be seen as a "strong" learner in the sense that it is not a weak learner as stump or other classical weak learners in boosting.

Following the first conclusion of this simulation we will compare more deeply, for different base learners, the diversity boosting and the classical boosting in the next section.

3.2.2. *Influence of the learner*

We illustrate here that the diversity boosting performance is dependant on the choice of the learner and its capacity to generate diversity. Intuitively, if a learner is too simple (e.g. a stump) the possible set of models which can be obtained on the data will be limited and so the gain induced by diversity boosting.

The results are presented in the tables 1 with columns corresponding to simple boosting ($F_{\kappa=0}$ without averaging and $F^*_{\kappa=0}$ with averaging), diversity boosting with $\kappa = 0.9$ ($F_{\kappa=0.9}$ without averaging and $F^*_{\kappa=0.9}$ with averaging) and base learners (stump, PRF and RF). The MSE are computed using 200 boosting iterations. Following the conclusions of section 3.2.1 we choose to set $\kappa = 0.9$ to encourage diversity and $\delta = 0.15$ so that each of the methods converge after 200 steps. For each Monte-Carlo run we compute the minimum MSE over gradient steps of each method and present the mean value of it, the standard deviations of this minimal MSE are also provided in brackets.

As expected, boosting and diversity boosting improves significantly over each base learner. Diversity boosting doesn't improve over boosting only for the stump learner which confirm the intuition about the limited capacity of stumps to generate diversity. The best results are obtained by the RF but the relative improvement over the original base learner is far more important for PRF. Our explanation is that PRF can generate more diversity than RF, inducing a large gain with diversity boosting.

Figure 5 represents the mean MSE (over 100 simulations) of diversity boosting (solid line corresponds to F^* and dashed lines to F) as a function of m the number of boosting steps for the 3 base learners. Note that the starting points of the curves are a bit higher that the MSE reported in Table 1 due to the fact that we subsample 50% of the data to fit the learners at each steps. This confirm the good convergence of the algorithm and its robustness regarding the choice of m.

3.3. *Electricity consumption data*

Our objective is to compare diversity boosting to classical base learners/boosting on a real data set. We evaluate the performance of our algorithm on a real data set of French electricity consumption. These data are provided by the system operator RTE (*Réseau de Transport d'Électricité*)[2]. Our dataset ranges from the 1st of January 2012 to the 15th of March 2020 with a 30 minutes sampling period. As electricity consumption depends strongly on weather conditions we add national averaged temperature from the website of the French weather forecaster Météo-France[3]. Note that we consider observed temperatures instead of forecasts so that this work is done with open data sets and the results can be easily reproduced.

[2] `https://opendata.rte-france.com`
[3] `https://donneespubliques.meteofrance.fr/`

$F_{\kappa=0}$	$F^*_{\kappa=0}$	$F_{\kappa=0.9}$	$F^*_{\kappa=0.9}$	F	Learner
4.74 (0.40)	5.72 (0.48)	5.75 (0.49)	5.17 (0.46)	19.44 (1.32)	Stump
5.71 (0.49)	6.31 (0.54)	**4.45 (0.36)**	5.38 (0.45)	14.05 (0.93)	PRF
4.38 (0.43)	4.55 (0.45)	**3.79 (0.34)**	4.15 (0.37)	7.16 (0.56)	RF

Table 1: MSE (standard deviations) for the best m with Stump, PRF and RF as a base learner.

We train the models on historical data from January 2012 to the end of August 2019. To avoid outliers we drop the holidays periods and bank holidays as well as the days before and after these periods. We consider the consumption at 8 p.m. each day and our goal is to forecast this consumption at a 24 hour horizon.

As in the simulation study we consider a RF model with 100 trees as our benchmark. Target is the Load at 20h and covariates are: *Date* a numerical variable indicating time since the beginning of the data set, *WeekDays* a categorical variable indicating the day of the week, *DLS* an indicator of winter/summer hour, *toy* a real number belonging to $[0, 1]$ indicating the position of an observation along the year from 0 (1st January) to 1 (31st December), $Temp$ the national average temperature in France, $Temp_{s95}$ and $Temp_{s99}$ exponential smoothings of this temperature with coefficient respectively of 0.95 and 0.99 to deal with thermal inertia of the buildings, $Temp_{s99_min}$ and $Temp_{s99_max}$ the min/max smoothed temperature per day, *Load*.48 lagged load (day before) and *Load*.336 lagged load (week before).

The results (mean RMSE on the test set for the best m as in section 3.2) are presented on Table 2. Figures 6 and 7 provide the mean RMSE as a function of the boosting steps for $\kappa = 0.5$ (left) and $\kappa = 0.9$ (right) for respectively learners RF and PRF. The best RMSE are obtained for F_κ, followed by F^*_κ. Interestingly, the overall best RMSE is achieved by $F_{\kappa=0.9}$ with the PRF learner which confirms on real data the conclusion of the simulation study: 1) PRF is a good base learner for diversity boosting, 2) choosing κ close to 1 gives improves forecasting performance as the learner can generate diversity. The RMSE curves of Figures 6 and 7 show that for the PRF learner and both values of κ the choice of the optimal number of boosting iterations is quite robust and choosing m sufficiently large with a small gradient step is again a good choice. For RF, the limited capacity (comparing to PRF) to generate diversity at a certain stage induces an increase of RMSE for $\kappa = 0.9$ after around 100 iterations, but even in that case the increase is slow and it could probably be eliminated by considering a smaller gradient step δ.

$F_{\kappa=0}$	$F^*_{\kappa=0}$	F_κ	F^*_κ	F	type
1295 (56)	1323 (54)	**1258 (62)**	1279 (60)	1665 (35)	RF, $\kappa = 0.5$
1301 (63)	1330 (62)	**1239 (59)**	1254 (57)	1665 (34)	RF, $\kappa = 0.9$
1298 (79)	1464 (82)	**1219 (67)**	1360 (72)	2545 (96)	PRF, $\kappa = 0.5$
1303 (76)	1470 (79)	**1147 (57)**	1258 (72)	2545 (96)	PRF, $\kappa = 0.9$

Table 2: RMSE (best m) on test set with RF and PRF as a base learner for $\kappa = 0.5$ and $\kappa = 0.9$.

To illustrate the capacity of diversity boosting to generate forecasts that are more diverse than classical boosting we plot the cumulative residuals of $PRF_{\kappa=0}$ and $PRF_{\kappa=0.9}$ in function of time on Figure 8. Given a forecast at time t denoted $\widehat{y_t}$ and observations y_t, the cumulative residuals over the interval $[1, T]$ are defined as $\bar{\varepsilon}_t = \sum_{i=1}^{t}(y_i - \widehat{y_i})$ for $t \in [1, T]$. We clearly see here that increasing κ generate trajectories of $\bar{\varepsilon}_t$ with more dispersion and this allows boosting predictors to achieve less biased forecasts: cumulative residuals of $PRF_{\kappa=0.9}$ are more centred around 0 after a few rounds of convergence (deep blue curves).

4. Conclusion and discussion

4.1. Conclusion

In this work, we propose a new boosting algorithm for regression problems based on the diversity formula. This method constructs at each step a base learner improving the diversity term of the diversity formula and then, try to reduce the mean square error. First experiments on simulated data and tree-based base learners confirm the potentiality of the method when the base learner is rich enough to generate diversity (RF and PRF). This could be considered as a surprise even if combining Random forests and boosting is a way to improve initial Random forests. Indeed, the idea of building two Random forests, the second obtained from the residuals of the first, and then add them together is a bias correction method and has proved to experimentally efficient, see for example [25] for a recent contribution on such one-step boosted forests. The authors sketched also the analysis of iterating the boosting process to continue to reduce the bias using the sum of all the estimates along the boosting steps. In addition, in our case, we are especially interested to use Random forests in the time series context often exhibiting strong temporal dependencies which potentially generate additional bias and then both boosting and diversity are welcome to improve the initial Random forest.

4.2. Improving diversity by sequential aggregation of experts

As illustrated on Figure 8, diversity boosting forecasters exhibit interesting temporal properties (see the end of section 3.3) that could be useful for online forecasting: where the observation are observed sequentially and the forecasts are updated in a data stream fashion.

Ensemble methods have already been applied for online forecasting (see [32] for a recent survey) and the questions which arise then is how to adapt these regression algorithms to non-stationary data (breaks, drift...) under memory/time constraints. Relevant approaches for that are online aggregation of experts (see [16]) where the base learners are forecasts coming from different methods or models and are called experts. The aggregation is performed in a sequential way: experts make predictions at each time instant and the forecaster must determine step

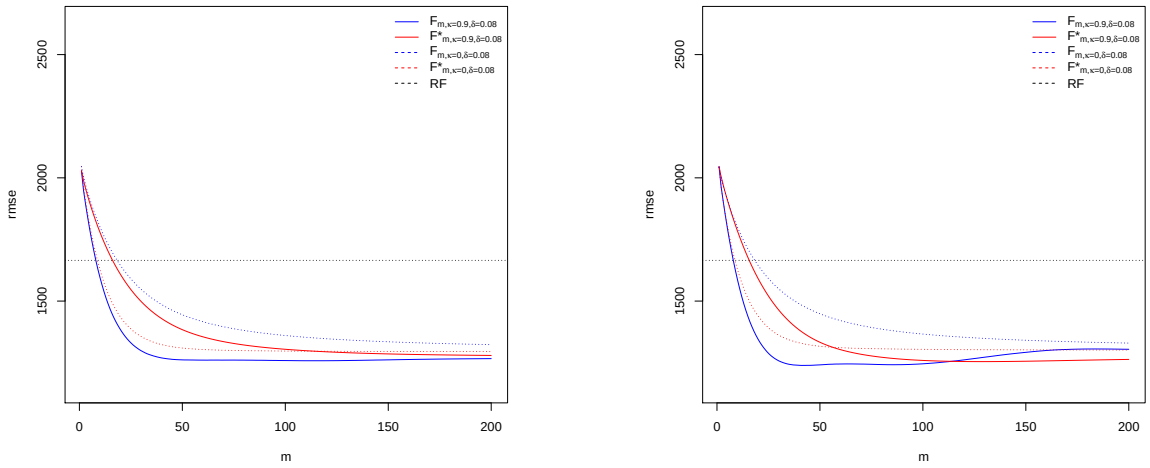

Figure 6: RMSE on test set as a function of boosting steps for $\kappa = 0.5$ (left) $\kappa = 0.9$ (right) gradient steps ($\delta = 0.08$) with RF (ntree=100, mtry=3) as base learner.

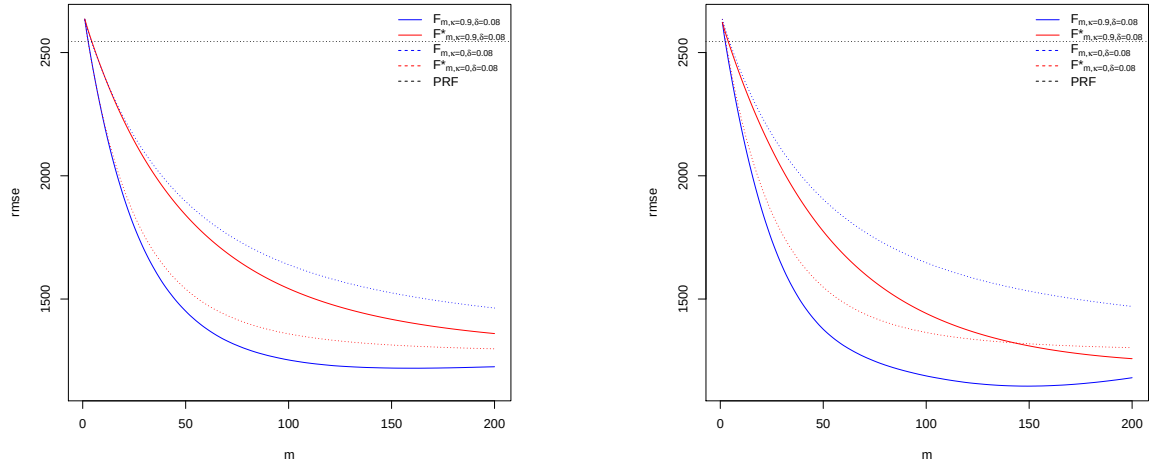

Figure 7: RMSE on test set as a function of boosting steps for $\kappa = 0.5$ (left) $\kappa = 0.9$ (right) gradient steps ($\delta = 0.08$) with PRF (ntree=100) as base learner.

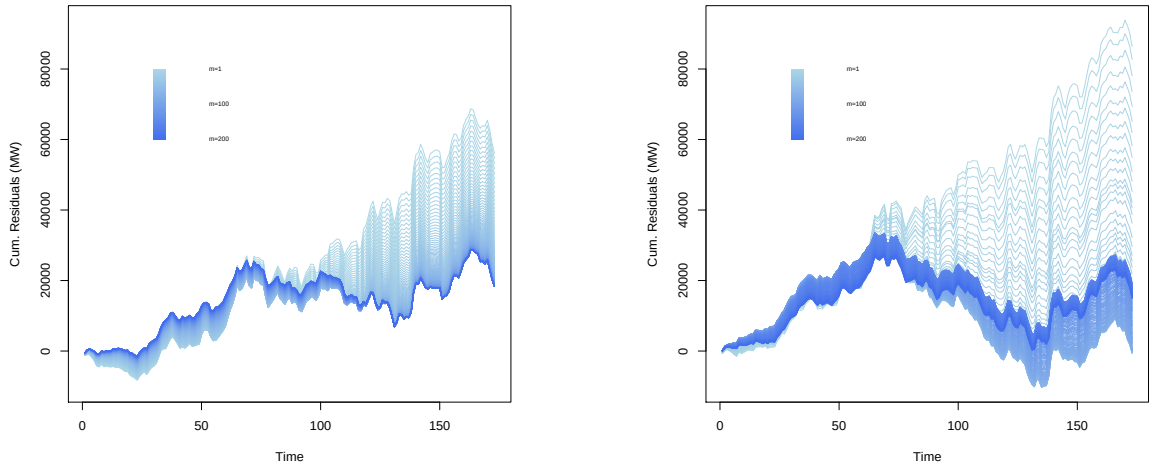

Figure 8: Cumulative residuals in function of time on the test set for $PRF_{\kappa=0}$ (left) and $PRF_{\kappa=0.9}$ (right) as a function of m

.

by step the future values of an observed time series. To build the prediction, the idea is to combine before each instant the forecasts of a finite set of experts producing a mixture. This has been successfully applied in e-commerce [29], air quality [2, 36] and electricity load forecasting [23, 26]. Also, [38] recently proposed to connect sequential expert aggregation with Mondrian Forest. Further studies could be done on how to extend diversity boosting in the online setting and connections with online aggregation of experts.

Acknowledgements. This work benefited from the support of the PGMO/IRSDI program (`https://www.fondation-hadamard.fr/en`).

References

[1] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau. The higgs boson machine learning challenge. In *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, pages 19–55, 2015.

[2] B. Auder, M. Bobbia, J.-M. Poggi, and B. Portier. Sequential aggregation of heterogeneous experts for pm10 forecasting. *Atmospheric Pollution Research*, 7(6):1101–1109, 2016.

[3] G. Biau and B. Cadre. Optimization by gradient boosting. In *Advances in Contemporary Statistics and Econometrics – Festschrift in Honour of Christine Thomas-Agnan*. Springer, 2021.

[4] G. Biau and E. Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016.

[5] A.-L. Boulesteix, S. Janitza, J. Kruppa, and I. R. König. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):493–507, 2012.

[6] M. Bourel, C. Crisci, and A. Martínez. Consensus methods based on machine learning techniques for marine phytoplankton presence–absence prediction. *Ecological Informatics*, 42:46 – 54, 2017.

[7] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[8] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[9] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[10] G. Brown, J. Wyatt, R. Harris, and X. Yao. Diversity creation methods: A survey and categorisation. *Journal of Information Fusion*, 6:5–20, 2005.

[11] G. Brown, J. L. Wyatt, and P. Tiňo. Managing diversity in regression ensembles. *J. Mach. Learn. Res.*, 6:1621–1650, Dec. 2005.

[12] P. Bühlmann and B. Yu. Boosting with the l2 loss. *Journal of the American Statistical Association*, 98(462):324–339, 2003.

[13] A. Callens, D. Morichon, S. Abadie, M. Delpey, and B. Liquet. Using random forest and gradient boosting trees to improve wave forecast at a specific location. *Applied Ocean Research*, 104:102339, 2020.

[14] P. Carmona, F. Climent, and A. Momparler. Predicting failure in the us banking sector: An extreme gradient boosting approach. *International Review of Economics & Finance*, 61:304–323, 2019.

[15] A. Castrillejo, J. Cugliari, F. Massa, and I. Ramirez. Electricity demand forecasting: The uruguayan case. In P. Drobinski, M. Mougeot, D. Picard, R. Plougonven, and P. Tankov, editors, *Renewable Energy: Forecasting and Risk Management*, pages 119–136, Cham, 2018. Springer International Publishing. ISBN 978-3-319-99052-1.

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

[17] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[18] B. Efron and T. Hastie. *Computer age statistical inference*, volume 5. Cambridge University Press, 2016.

[19] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[20] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2):337–407, 2000.

[21] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2001.

[22] J. H. Friedman et al. Multivariate adaptive regression splines. *The annals of statistics*, 19(1):1–67, 1991.

[23] P. Gaillard, Y. Goude, and R. Nedellec. Additive models and robust aggregation for gefcom2014 probabilistic electric load and electricity price forecasting. *International Journal of forecasting*, 32(3):1038–1050, 2016.

[24] R. Genuer, V. Michel, E. Eger, and B. Thirion. Random forests based feature selection for decoding fmri data. In *Proceedings Compstat*, volume 267, pages 1–8, 2010.

[25] I. Ghosal and G. Hooker. Boosting random forests to reduce bias; one-step boosted forest and its variance estimate. *Journal of Computational and Graphical Statistics*, pages 1–10, 2020.

[26] B. Goehry, Y. Goude, P. Massart, and J.-M. Poggi. Aggregation of multi-scale experts for bottom-up load forecasting. *IEEE Transactions on Smart Grid*, 11(3):1895–1904, 2019.

[27] H. Grabner and H. Bischof. On-line boosting and vision. In *2006 IEEE*

Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 1, pages 260–267. Ieee, 2006.

[28] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[29] M. Huard, R. Garnier, and G. Stoltz. Hierarchical robust aggregation of sales forecasts at aggregated levels in e-commerce, based on exponential smoothing and Holt's linear trend method. working paper or preprint, June 2020.

[30] A. J. Izenman. *Modern multivariate statistical techniques. Regression, classification and manifold learning*. Springer, 2008.

[31] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, pages 3146–3154, 2017.

[32] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak. Ensemble learning for data stream analysis: A survey. *Information Fusion*, 37:132 – 156, 2017. ISSN 1566-2535.

[33] F. Leisch and E. Dimitriadou. mlbench—a collection for artificial and realworld machine learning benchmarking problems. R package, version 0.5-6, 2001.

[34] W. Lin, Z. Wu, L. Lin, A. Wen, and J. Li. An ensemble random forest algorithm for insurance big data analysis. *Ieee access*, 5:16568–16575, 2017.

[35] Y. Liu and X. Yao. Ensemble learning via negative correlation. *Neural Netw.*, 12(10):1399–1404, Dec. 1999.

[36] V. Mallet, G. Stoltz, and B. Mauricette. Ozone ensemble forecast with machine learning algorithms. *Journal of Geophysical Research: Atmospheres*, 114(D5), 2009.

[37] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Functional Gradient Techniques for Combining Hypotheses. In *Advances in Large-Margin Classifiers*. The MIT Press, 09 2000.

[38] J. Mourtada, S. Gaïffas, and E. Scornet. Amf: Aggregated mondrian forests for online learning. *arXiv preprint arXiv:1906.10529*, 2019.

[39] R. Polikar. Ensemble learning. In *In: Zhang C., Ma Y. (eds) Ensemble Machine Learning.*, pages 1–34. Springer, Boston, MA., 2012.

[40] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. Catboost: unbiased boosting with categorical features. In *Advances in neural information processing systems*, pages 6638–6648, 2018.

[41] N. V. Queipo and E. Nava. A gradient boosting approach with diversity promoting measures for the ensemble of surrogates in engineering. *Structural and Multidisciplinary Optimization*, 60(4):1289–1311, 2019.

[42] H. Reeve and G. Brown. Modular autoencoders for ensemble feature extraction. In D. Storcheus, A. Rostamizadeh, and S. Kumar, editors, *Feature Extraction: Modern Questions and Challenges*, volume 44 of *Proceedings of Machine Learning Research*, pages 242–259, Montreal, Canada, 11 Dec 2015. PMLR.

[43] H. W. Reeve and G. Brown. Diversity and degrees of freedom in regression ensembles. *Neurocomputing*, 298:55 – 68, 2018.

[44] L. Reyzin. On boosting sparse parities. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[45] R. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. Adaptive Computation and Machine Learning Series. Mit Press, 2012.

[46] S. Seifert. Application of random forest based approaches to surface-enhanced raman scattering data. *Scientific reports*, 10(1):1–11, 2020.

[47] S. B. Taieb, R. Huser, R. J. Hyndman, and M. G. Genton. Forecasting uncertainty in electricity smart meter data by boosting additive quantile regression. *IEEE Transactions on Smart Grid*, 7(5):2448–2455, 2016.

[48] M. Taillardat, O. Mestre, M. Zamo, and P. Naveau. Calibrated ensemble forecasts using quantile regression forests and ensemble model output statistics. *Monthly Weather Review*, 144(6):2375–2393, 2016.

[49] T. M. Therneau and E. J. Atkinson. An introduction to recursive partitioning using the rpart routines. Technical report, Technical report Mayo Foundation, 2019.

[50] M. N. Wright and A. Ziegler. ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1):1–17, 2017.

[51] W. Xu, J. Zhang, Q. Zhang, and X. Wei. Risk prediction of type 2 diabetes based on random forest model. In *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pages 382–386. IEEE, 2017.