Optimization by gradient boosting

Gérard Biau

Nantes, March 2018

Coauthors

Benoît Cadre (ENS Rennes) **Laurent Rouvière** (University Rennes 2)

Outline

1. Boosting and gradient boosting
2. Mathematical context
3. Two algorithms
4. Convergence
5. Large sample properties
6. Boosting gradient boosting

Boosting and gradient boosting

Boosting at a glance

- Boosting: algorithms that convert weak learners to strong ones.

Boosting at a glance

- Boosting: algorithms that convert weak learners to strong ones.
- Idea: combine simple predictors to produce a weighted committee.

Boosting at a glance

- Boosting: algorithms that convert weak learners to strong ones.
- Idea: combine simple predictors to produce a weighted committee.
- One of the most powerful learning ideas introduced in modern times.

Boosting at a glance

- Boosting: algorithms that convert weak learners to strong ones.
- Idea: combine simple predictors to produce a weighted committee.
- One of the most powerful learning ideas introduced in modern times.
- Considerable impact in statistics and machine learning.

A brief history of boosting

1990-1997: Freund and Schapire's Adaboost.

A brief history of boosting

1990-1997: Freund and Schapire's Adaboost.

- Adaboost is an iterative classification algorithm.

A brief history of boosting

1990-1997: Freund and Schapire's Adaboost.

- Adaboost is an iterative classification algorithm.
- For a fixed number of iterations, do:

A brief history of boosting

1990-1997: Freund and Schapire's Adaboost.

- Adaboost is an iterative classification algorithm.
- For a fixed number of iterations, do:
  - ▷ At each iteration, select a base classifier and assign a weight to it;

A brief history of boosting

1990-1997: Freund and Schapire's Adaboost.

- Adaboost is an iterative classification algorithm.
- For a fixed number of iterations, do:
  - ▷ At each iteration, select a base classifier and assign a weight to it;
  - ▷ Misclassified observations have their weights increased;

A brief history of boosting

1990-1997: Freund and Schapire's Adaboost.

- Adaboost is an iterative classification algorithm.
- For a fixed number of iterations, do:
  - ▷ At each iteration, select a base classifier and assign a weight to it;
  - ▷ Misclassified observations have their weights increased;
  - ▷ Output the weighted majority vote of the chosen classifiers.

A brief history of boosting

1990-1997: Freund and Schapire's Adaboost.

- Adaboost is an iterative classification algorithm.
- For a fixed number of iterations, do:
  - ▷ At each iteration, select a base classifier and assign a weight to it;
  - ▷ Misclassified observations have their weights increased;
  - ▷ Output the weighted majority vote of the chosen classifiers.

1997-2004: Breiman's papers and technical reports.

A brief history of boosting

1990-1997: Freund and Schapire's Adaboost.

- Adaboost is an iterative classification algorithm.
- For a fixed number of iterations, do:
    - ▷ At each iteration, select a base classifier and assign a weight to it;
    - ▷ Misclassified observations have their weights increased;
    - ▷ Output the weighted majority vote of the chosen classifiers.

1997-2004: Breiman's papers and technical reports.

- AdaBoost is a gradient-descent-type algorithm in a function space.

A brief history of boosting

1990-1997: Freund and Schapire's Adaboost.

- Adaboost is an iterative classification algorithm.
- For a fixed number of iterations, do:
  - ▷ At each iteration, select a base classifier and assign a weight to it;
  - ▷ Misclassified observations have their weights increased;
  - ▷ Output the weighted majority vote of the chosen classifiers.

1997-2004: Breiman's papers and technical reports.

- AdaBoost is a gradient-descent-type algorithm in a function space.
- Boosting is at the frontier of numerical optimization and statistics.

A brief history of boosting

2001-2002: Friedman's gradient boosting.

A brief history of boosting

2001-2002: Friedman's gradient boosting.

- A general statistical framework for boosting.

A brief history of boosting

2001-2002: Friedman's gradient boosting.

- A general statistical framework for boosting.
- Interpretation as optimization in a function space.

A brief history of boosting

2001-2002: Friedman's gradient boosting.

- A general statistical framework for boosting.
- Interpretation as optimization in a function space.
- Arbitrary loss functions, for classification and regression.

A brief history of boosting

2001-2002: Friedman's gradient boosting.

- A general statistical framework for boosting.
- Interpretation as optimization in a function space.
- Arbitrary loss functions, for classification and regression.
- Special attention paid to decision trees as weak learners.

A brief history of boosting

2001-2002: Friedman's gradient boosting.

- A general statistical framework for boosting.
- Interpretation as optimization in a function space.
- Arbitrary loss functions, for classification and regression.
- Special attention paid to decision trees as weak learners.

2000: Mason et al. analysis.

A brief history of boosting

2001-2002: Friedman's gradient boosting.

- A general statistical framework for boosting.
- Interpretation as optimization in a function space.
- Arbitrary loss functions, for classification and regression.
- Special attention paid to decision trees as weak learners.

2000: Mason et al. analysis.

- Boosting is a principle to optimize a convex risk in a function space.

A brief history of boosting

2001-2002: Friedman's gradient boosting.

- A general statistical framework for boosting.
- Interpretation as optimization in a function space.
- Arbitrary loss functions, for classification and regression.
- Special attention paid to decision trees as weak learners.

2000: Mason et al. analysis.

- Boosting is a principle to optimize a convex risk in a function space.
- The increments point in the negative gradient direction.

A brief history of boosting

2001-2002: Friedman's gradient boosting.

- A general statistical framework for boosting.
- Interpretation as optimization in a function space.
- Arbitrary loss functions, for classification and regression.
- Special attention paid to decision trees as weak learners.

2000: Mason et al. analysis.

- Boosting is a principle to optimize a convex risk in a function space.
- The increments point in the negative gradient direction.
- First attempt to understand the mathematical forces of boosting.

A brief history of boosting

2003-2007: Boosting from a statistical perspective.

A brief history of boosting

2003-2007: Boosting from a statistical perspective.

- Empirical risk minimization with a convex loss.

A brief history of boosting

2003-2007: Boosting from a statistical perspective.

- Empirical risk minimization with a convex loss.
- Examples: Blanchard et al. (2003), Lugosi and Vayatis (2004).

A brief history of boosting

2003-2007: Boosting from a statistical perspective.

- Empirical risk minimization with a convex loss.
- Examples: Blanchard et al. (2003), Lugosi and Vayatis (2004).
- Idealized models: statistical properties but no optimization.

A brief history of boosting

2003-2007: Boosting from a statistical perspective.

- Empirical risk minimization with a convex loss.
- Examples: Blanchard et al. (2003), Lugosi and Vayatis (2004).
- Idealized models: statistical properties but no optimization.
- Regularization via early stopping.

A brief history of boosting

2003-2007: Boosting from a statistical perspective.

- Empirical risk minimization with a convex loss.
- Examples: Blanchard et al. (2003), Lugosi and Vayatis (2004).
- Idealized models: statistical properties but no optimization.
- Regularization via early stopping.
- Examples: Bühlmann and Yu (2003), Mannor et al. (2003), Zhang and Yu (2005), Bickel et al. (2006), Bartlett and Traskin (2007).

A brief history of boosting

2003-2007: Boosting from a statistical perspective.

- Empirical risk minimization with a convex loss.
- Examples: Blanchard et al. (2003), Lugosi and Vayatis (2004).
- Idealized models: statistical properties but no optimization.
- Regularization via early stopping.
- Examples: Bühlmann and Yu (2003), Mannor et al. (2003), Zhang and Yu (2005), Bickel et al. (2006), Bartlett and Traskin (2007).

2016: XGBoost of Chen and Guestrin.

A brief history of boosting

2003-2007: Boosting from a statistical perspective.

- Empirical risk minimization with a convex loss.
- Examples: Blanchard et al. (2003), Lugosi and Vayatis (2004).
- Idealized models: statistical properties but no optimization.
- Regularization via early stopping.
- Examples: Bühlmann and Yu (2003), Mannor et al. (2003), Zhang and Yu (2005), Bickel et al. (2006), Bartlett and Traskin (2007).

2016: XGBoost of Chen and Guestrin.

- A scalable implementation of gradient tree boosting.

A brief history of boosting

2003-2007: Boosting from a statistical perspective.

- Empirical risk minimization with a convex loss.
- Examples: Blanchard et al. (2003), Lugosi and Vayatis (2004).
- Idealized models: statistical properties but no optimization.
- Regularization via early stopping.
- Examples: Bühlmann and Yu (2003), Mannor et al. (2003), Zhang and Yu (2005), Bickel et al. (2006), Bartlett and Traskin (2007).

2016: XGBoost of Chen and Guestrin.

- A scalable implementation of gradient tree boosting.
- Inspired by Friedman's principles.

A brief history of boosting

2003-2007: Boosting from a statistical perspective.

- Empirical risk minimization with a convex loss.
- Examples: Blanchard et al. (2003), Lugosi and Vayatis (2004).
- Idealized models: statistical properties but no optimization.
- Regularization via early stopping.
- Examples: Bühlmann and Yu (2003), Mannor et al. (2003), Zhang and Yu (2005), Bickel et al. (2006), Bartlett and Traskin (2007).

2016: XGBoost of Chen and Guestrin.

- A scalable implementation of gradient tree boosting.
- Inspired by Friedman's principles.
- Outstanding results in numerous data challenges.

A brief history of boosting

2003-2007: Boosting from a statistical perspective.

- Empirical risk minimization with a convex loss.
- Examples: Blanchard et al. (2003), Lugosi and Vayatis (2004).
- Idealized models: statistical properties but no optimization.
- Regularization via early stopping.
- Examples: Bühlmann and Yu (2003), Mannor et al. (2003), Zhang and Yu (2005), Bickel et al. (2006), Bartlett and Traskin (2007).

2016: XGBoost of Chen and Guestrin.

- A scalable implementation of gradient tree boosting.
- Inspired by Friedman's principles.
- Outstanding results in numerous data challenges.
- The objective is regularized to avoid overfitting.

Agenda

- There is to date no sound theory of gradient boosting.

Agenda

- There is to date no sound theory of gradient boosting.
- Optimization is the natural environment for gradient-type methods.

Agenda

- There is to date no sound theory of gradient boosting.
- Optimization is the natural environment for gradient-type methods.
- Our objective today:

Agenda

- There is to date no sound theory of gradient boosting.
- Optimization is the natural environment for gradient-type methods.
- Our objective today:
    - ▷ Clarify the mathematical principles of the algorithms;

Agenda

- There is to date no sound theory of gradient boosting.
- Optimization is the natural environment for gradient-type methods.
- Our objective today:
  - ▷ Clarify the mathematical principles of the algorithms;
  - ▷ Adopt the point of view of functional optimization in L^2;

Agenda

- There is to date no sound theory of gradient boosting.
- Optimization is the natural environment for gradient-type methods.
- Our objective today:
    - ▷ Clarify the mathematical principles of the algorithms;
    - ▷ Adopt the point of view of functional optimization in L^2;
    - ▷ Prove convergence as the number of iterations tends to infinity;

Agenda

- There is to date no sound theory of gradient boosting.
- Optimization is the natural environment for gradient-type methods.
- Our objective today:
  - ▷ Clarify the mathematical principles of the algorithms;
  - ▷ Adopt the point of view of functional optimization in L^2;
  - ▷ Prove convergence as the number of iterations tends to infinity;
  - ▷ Introduce a reasonable statistical framework for consistency properties.

Mathematical context

Notation

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.

Notation

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.
- $\mathscr{Y}$ is either a finite set (classification) or a subset of $\mathbb{R}$ (regression).

Notation

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.
- $\mathscr{Y}$ is either a finite set (classification) or a subset of $\mathbb{R}$ (regression).
- Goal: construct a predictor $F : \mathscr{X} \to \mathbb{R}$.

Notation

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.
- $\mathscr{Y}$ is either a finite set (classification) or a subset of $\mathbb{R}$ (regression).
- Goal: construct a predictor $F : \mathscr{X} \to \mathbb{R}$.

☞ In ± 1-classification, the final rule is $+1$ if $F(x) > 0$ and -1 otherwise.

Notation

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.
- $\mathscr{Y}$ is either a finite set (classification) or a subset of $\mathbb{R}$ (regression).
- Goal: construct a predictor $F : \mathscr{X} \to \mathbb{R}$.

☞ In ± 1-classification, the final rule is $+1$ if $F(x) > 0$ and -1 otherwise.

- $\mathscr{F}$ = class of functions $f : \mathscr{X} \to \mathbb{R}$ (the weak learners).

Notation

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.
- $\mathscr{Y}$ is either a finite set (classification) or a subset of $\mathbb{R}$ (regression).
- Goal: construct a predictor $F : \mathscr{X} \to \mathbb{R}$.

☞ In ± 1-classification, the final rule is $+1$ if $F(x) > 0$ and -1 otherwise.

- $\mathscr{F}$ = class of functions $f : \mathscr{X} \to \mathbb{R}$ (the weak learners).
- Objective: minimize over $\text{lin}(\mathscr{F})$ the empirical risk functional

$$C_n(F) = \frac{1}{n} \sum_{i=1}^{n} \psi(F(X_i), Y_i).$$

Notation

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.
- $\mathscr{Y}$ is either a finite set (classification) or a subset of $\mathbb{R}$ (regression).
- Goal: construct a predictor $F : \mathscr{X} \to \mathbb{R}$.

☞ In ± 1-classification, the final rule is $+1$ if $F(x) > 0$ and -1 otherwise.

- $\mathscr{F}$ = class of functions $f : \mathscr{X} \to \mathbb{R}$ (the weak learners).
- Objective: minimize over $\text{lin}(\mathscr{F})$ the empirical risk functional

$$C_n(F) = \frac{1}{n} \sum_{i=1}^{n} \psi(F(X_i), Y_i).$$

- The loss function $\psi : \mathbb{R} \times \mathscr{Y} \to \mathbb{R}_+$ is convex in its first argument.

Notation

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.
- $\mathscr{Y}$ is either a finite set (classification) or a subset of $\mathbb{R}$ (regression).
- Goal: construct a predictor $F : \mathscr{X} \to \mathbb{R}$.

☞ In ± 1-classification, the final rule is $+1$ if $F(x) > 0$ and -1 otherwise.

- $\mathscr{F}$ = class of functions $f : \mathscr{X} \to \mathbb{R}$ (the weak learners).
- Objective: minimize over $\text{lin}(\mathscr{F})$ the empirical risk functional

$$C_n(F) = \frac{1}{n} \sum_{i=1}^{n} \psi(F(X_i), Y_i).$$

- The loss function $\psi : \mathbb{R} \times \mathscr{Y} \to \mathbb{R}_+$ is convex in its first argument.
- Example: $\psi(x, y) = (y - x)^2$ and

$$C_n(F) = \frac{1}{n} \sum_{i=1}^{n} (Y_i - F(X_i))^2.$$

A more general problem

- Clearly,

$$C_n(F) = \mathbb{E}\psi(F(X), Y),$$

where (X, Y) is a random pair with distribution μ_n.

A more general problem

- Clearly,

$$C_n(F) = \mathbb{E}\psi(F(X), Y),$$

where (X, Y) is a random pair with distribution μ_n.

- The population version of C_n is

$$C(F) = \mathbb{E}\psi(F(X_1), Y_1).$$

A more general problem

- Clearly,
$$C_n(F) = \mathbb{E}\psi(F(X), Y),$$
where (X, Y) is a random pair with distribution μ_n.
- The population version of C_n is
$$C(F) = \mathbb{E}\psi(F(X_1), Y_1).$$
- General context: (X, Y) is a generic pair with distribution $\mu_{X,Y}$

A more general problem

- Clearly,

$$C_n(F) = \mathbb{E}\psi(F(X), Y),$$

  where (X, Y) is a random pair with distribution μ_n.
- The population version of C_n is

$$C(F) = \mathbb{E}\psi(F(X_1), Y_1).$$

- General context: (X, Y) is a generic pair with distribution $\mu_{X,Y}$
  - ▷ $\mu_{X,Y}$ = distribution of (X_1, Y_1) (theoretical risk);
  - ▷ $\mu_{X,Y}$ = standard empirical measure μ_n (empirical risk);
  - ▷ $\mu_{X,Y}$ = any smoothed version of μ_n (smoothed empirical risk).

A more general problem

- Clearly,

$$C_n(F) = \mathbb{E}\psi(F(X), Y),$$

where (X, Y) is a random pair with distribution μ_n.

- The population version of C_n is

$$C(F) = \mathbb{E}\psi(F(X_1), Y_1).$$

- General context: (X, Y) is a generic pair with distribution $\mu_{X,Y}$
  - ▷ $\mu_{X,Y}$ = distribution of (X_1, Y_1) (theoretical risk);
  - ▷ $\mu_{X,Y}$ = standard empirical measure μ_n (empirical risk);
  - ▷ $\mu_{X,Y}$ = any smoothed version of μ_n (smoothed empirical risk).

Objective

Minimize $C(F) = \mathbb{E}\psi(F(X), Y)$ over $\text{lin}(\mathscr{F})$, where $\mathscr{F} \subset L^2(\mu_X)$.

A more general problem

- Clearly,

$$C_n(F) = \mathbb{E}\psi(F(X), Y),$$

  where (X, Y) is a random pair with distribution μ_n.
- The population version of C_n is

$$C(F) = \mathbb{E}\psi(F(X_1), Y_1).$$

- General context: (X, Y) is a generic pair with distribution $\mu_{X,Y}$
  - ▷ $\mu_{X,Y}$ = distribution of (X_1, Y_1) (theoretical risk);
  - ▷ $\mu_{X,Y}$ = standard empirical measure μ_n (empirical risk);
  - ▷ $\mu_{X,Y}$ = any smoothed version of μ_n (smoothed empirical risk).

Objective

Minimize $C(F) = \mathbb{E}\psi(F(X), Y)$ over $\text{lin}(\mathscr{F})$, where $\mathscr{F} \subset L^2(\mu_X)$.

- Typical $\mathscr{F}$: decision trees in $\mathbb{R}^d$ with k terminal nodes.

A more general problem

- Clearly,

$$C_n(F) = \mathbb{E}\psi(F(X), Y),$$

where (X, Y) is a random pair with distribution μ_n.

- The population version of C_n is

$$C(F) = \mathbb{E}\psi(F(X_1), Y_1).$$

- General context: (X, Y) is a generic pair with distribution $\mu_{X,Y}$
  - ▷ $\mu_{X,Y}$ = distribution of (X_1, Y_1) (theoretical risk);
  - ▷ $\mu_{X,Y}$ = standard empirical measure μ_n (empirical risk);
  - ▷ $\mu_{X,Y}$ = any smoothed version of μ_n (smoothed empirical risk).

Objective

Minimize $C(F) = \mathbb{E}\psi(F(X), Y)$ over $\mathrm{lin}(\mathscr{F})$, where $\mathscr{F} \subset L^2(\mu_X)$.

- Typical $\mathscr{F}$: decision trees in $\mathbb{R}^d$ with k terminal nodes.
- Each $f \in \mathscr{F}$ takes the form $f = \sum_{j=1}^{k} \beta_j \mathbb{1}_{A_j}$.

Some assumptions

Subgradient

$\xi(\cdot, y)$ is a subgradient of the convex function $\psi(\cdot, y)$. Recall that

1. $\xi(x, y) \in [\partial_x^- \psi(x, y); \partial_x^+ \psi(x, y)]$.
2. $\psi(x_1, y) \geq \psi(x_2, y) + \xi(x_2, y)(x_1 - x_2)$.

Some assumptions

Subgradient

$\xi(\cdot, y)$ is a subgradient of the convex function $\psi(\cdot, y)$. Recall that

1. $\xi(x, y) \in [\partial_x^- \psi(x, y); \partial_x^+ \psi(x, y)]$.
2. $\psi(x_1, y) \geq \psi(x_2, y) + \xi(x_2, y)(x_1 - x_2)$.

Assumption $\mathbf{A_1}$

One has $\mathbb{E}\psi(0, Y) < \infty$. In addition, for all $F \in L^2(\mu_X)$, there exists $\delta > 0$ such that

$$\sup_{G \in L^2(\mu_X) : \|G - F\|_{\mu_X} \leq \delta} \left(\mathbb{E}|\partial_x^- \psi(G(X), Y)|^2 + \mathbb{E}|\partial_x^+ \psi(G(X), Y)|^2\right) < \infty.$$

Some assumptions

Subgradient

$\xi(\cdot, y)$ is a subgradient of the convex function $\psi(\cdot, y)$. Recall that

1. $\xi(x, y) \in [\partial_x^- \psi(x, y); \partial_x^+ \psi(x, y)]$.
2. $\psi(x_1, y) \geq \psi(x_2, y) + \xi(x_2, y)(x_1 - x_2)$.

Assumption $\mathbf{A_1}$

One has $\mathbb{E}\psi(0, Y) < \infty$. In addition, for all $F \in L^2(\mu_X)$, there exists $\delta > 0$ such that

$$\sup_{G \in L^2(\mu_X) : \|G - F\|_{\mu_X} \leq \delta} \left(\mathbb{E}|\partial_x^- \psi(G(X), Y)|^2 + \mathbb{E}|\partial_x^+ \psi(G(X), Y)|^2 \right) < \infty.$$

Interpretation

$C(F) < \infty$ for all $F \in L^2(\mu_X)$ and C is continuous.

Some assumptions

Assumption A_2

There exists $\alpha > 0$ such that, for all $y \in \mathscr{Y}$, the function $\psi(\cdot, y)$ is α-strongly convex, i.e., for all $(x_1, x_2) \in \mathbb{R}^2$ and $t \in [0, 1]$,

$$\psi(tx_1 + (1-t)x_2, y) \leq t\psi(x_1, y) + (1-t)\psi(x_2, y) - \frac{\alpha}{2} t(1-t)(x_1 - x_2)^2.$$

Some assumptions

Assumption A_2

There exists $\alpha > 0$ such that, for all $y \in \mathscr{Y}$, the function $\psi(\cdot, y)$ is α-strongly convex, i.e., for all $(x_1, x_2) \in \mathbb{R}^2$ and $t \in [0, 1]$,

$$\psi(tx_1 + (1-t)x_2, y) \leq t\psi(x_1, y) + (1-t)\psi(x_2, y) - \frac{\alpha}{2}t(1-t)(x_1 - x_2)^2.$$

Interpretation

One has

$$\psi(x_1, y) \geq \psi(x_2, y) + \xi(x_2, y)(x_1 - x_2) + \frac{\alpha}{2}(x_1 - x_2)^2$$

instead of

$$\psi(x_1, y) \geq \psi(x_2, y) + \xi(x_2, y)(x_1 - x_2).$$

Some assumptions

Assumption $\mathbf{A_3}$

There exists a positive constant L such that, for all $(x_1, x_2) \in \mathbb{R}^2$,

$$|\mathbb{E}(\xi(x_1, Y) - \xi(x_2, Y) \mid X)| \leq L|x_1 - x_2|.$$

Some assumptions

Assumption A_3

There exists a positive constant L such that, for all $(x_1, x_2) \in \mathbb{R}^2$,

$$|\mathbb{E}(\xi(x_1, Y) - \xi(x_2, Y) \,|\, X)| \leq L|x_1 - x_2|.$$

A more digest Assumption A_3'

For all $y \in \mathscr{Y}$, the function $\psi(\cdot, y)$ is continuously differentiable, and there exists a positive constant L such that

$$|\partial_x \psi(x_1, y) - \partial_x \psi(x_2, y)| \leq L|x_1 - x_2|.$$

Some assumptions

Assumption $\mathbf{A_3}$

There exists a positive constant L such that, for all $(x_1, x_2) \in \mathbb{R}^2$,

$$|\mathbb{E}(\xi(x_1, Y) - \xi(x_2, Y) \,|\, X)| \leq L|x_1 - x_2|.$$

A more digest Assumption $\mathbf{A_3'}$

For all $y \in \mathscr{Y}$, the function $\psi(\cdot, y)$ is continuously differentiable, and there exists a positive constant L such that

$$|\partial_x \psi(x_1, y) - \partial_x \psi(x_2, y)| \leq L|x_1 - x_2|.$$

Interpretation

The functional C is differentiable at any $F \in L^2(\mu_X)$ with

$$dC(F; G) = \langle \nabla C(F), G \rangle_{\mu_X},$$

where $\nabla C(F)(x) := \int \partial_x \psi(F(x), y) \mu_{Y|X=x}(\mathrm{d}y)$.

Examples in regression analysis

- Squared error loss: $\psi(x, y) = (y - x)^2$.

Examples in regression analysis

- Squared error loss: $\psi(x, y) = (y - x)^2$.
  - ▷ Assumption $\mathbf{A_1}$: $\mathbb{E}Y^2 < \infty$ ✔

Examples in regression analysis

- Squared error loss: $\psi(x, y) = (y - x)^2$.
  - ▷ Assumption $\mathbf{A_1}$: $\mathbb{E}Y^2 < \infty$ ✔
  - ▷ Assumption $\mathbf{A_2}$: 2-strongly convex ✔

Examples in regression analysis

- Squared error loss: $\psi(x, y) = (y - x)^2$.
  - ▷ Assumption $\mathbf{A_1}$: $\mathbb{E}Y^2 < \infty$ ✔
  - ▷ Assumption $\mathbf{A_2}$: 2-strongly convex ✔
  - ▷ Assumption $\mathbf{A_3'}$: $\partial_x \psi(x, y) = 2(x - y)$ and $L = 2$ ✔

Examples in regression analysis

- Squared error loss: $\psi(x, y) = (y - x)^2$.
  - ▷ Assumption $\mathbf{A_1}$: $\mathbb{E}Y^2 < \infty$ ✔
  - ▷ Assumption $\mathbf{A_2}$: 2-strongly convex ✔
  - ▷ Assumption $\mathbf{A_3'}$: $\partial_x \psi(x, y) = 2(x - y)$ and $L = 2$ ✔
- Absolute error loss: $\psi(x, y) = |y - x|$.

Examples in regression analysis

- Squared error loss: $\psi(x, y) = (y - x)^2$.
  - ▷ Assumption $\mathbf{A_1}$: $\mathbb{E}Y^2 < \infty$ ✔
  - ▷ Assumption $\mathbf{A_2}$: 2-strongly convex ✔
  - ▷ Assumption $\mathbf{A}'_3$: $\partial_x \psi(x, y) = 2(x - y)$ and $L = 2$ ✔
- Absolute error loss: $\psi(x, y) = |y - x|$.
  - ▷ Assumption $\mathbf{A_1}$: $\mathbb{E}|Y| < \infty$ ✔

Examples in regression analysis

- Squared error loss: $\psi(x,y) = (y-x)^2$.
  - ▷ Assumption $\mathbf{A_1}$: $\mathbb{E}Y^2 < \infty$ ✔
  - ▷ Assumption $\mathbf{A_2}$: 2-strongly convex ✔
  - ▷ Assumption $\mathbf{A_3'}$: $\partial_x \psi(x,y) = 2(x-y)$ and $L = 2$ ✔
- Absolute error loss: $\psi(x,y) = |y-x|$.
  - ▷ Assumption $\mathbf{A_1}$: $\mathbb{E}|Y| < \infty$ ✔
  - ▷ Assumption $\mathbf{A_2}$: convex but not strongly convex ✖

Examples in regression analysis

- Squared error loss: $\psi(x, y) = (y - x)^2$.
  - ▷ Assumption $\mathbf{A_1}$: $\mathbb{E}Y^2 < \infty$ ✔
  - ▷ Assumption $\mathbf{A_2}$: 2-strongly convex ✔
  - ▷ Assumption $\mathbf{A_3'}$: $\partial_x \psi(x, y) = 2(x - y)$ and $L = 2$ ✔
- Absolute error loss: $\psi(x, y) = |y - x|$.
  - ▷ Assumption $\mathbf{A_1}$: $\mathbb{E}|Y| < \infty$ ✔
  - ▷ Assumption $\mathbf{A_2}$: convex but not strongly convex ✖
  - ▷ Solution: regularization via

$$\psi(x, y) = |y - x| + \gamma x^2,$$

which is (2γ)-strongly convex in x ✔

Examples in regression analysis

- Squared error loss: $\psi(x, y) = (y - x)^2$.
  - ▷ Assumption $\mathbf{A_1}$: $\mathbb{E}Y^2 < \infty$ ✔
  - ▷ Assumption $\mathbf{A_2}$: 2-strongly convex ✔
  - ▷ Assumption $\mathbf{A}_3'$: $\partial_x \psi(x, y) = 2(x - y)$ and $L = 2$ ✔
- Absolute error loss: $\psi(x, y) = |y - x|$.
  - ▷ Assumption $\mathbf{A_1}$: $\mathbb{E}|Y| < \infty$ ✔
  - ▷ Assumption $\mathbf{A_2}$: convex but not strongly convex ✖
  - ▷ Solution: regularization via

    $$\psi(x, y) = |y - x| + \gamma x^2,$$

    which is (2γ)-strongly convex in x ✔
  - ▷ Assumption $\mathbf{A}_3'$: $\psi(\cdot, y)$ is not differentiable at y ✖

Examples in regression analysis

- Squared error loss: $\psi(x, y) = (y - x)^2$.
  - Assumption $\mathbf{A}_1$: $\mathbb{E}Y^2 < \infty$ ✔
  - Assumption $\mathbf{A}_2$: 2-strongly convex ✔
  - Assumption $\mathbf{A}_3'$: $\partial_x \psi(x, y) = 2(x - y)$ and $L = 2$ ✔
- Absolute error loss: $\psi(x, y) = |y - x|$.
  - Assumption $\mathbf{A}_1$: $\mathbb{E}|Y| < \infty$ ✔
  - Assumption $\mathbf{A}_2$: convex but not strongly convex ✖
  - Solution: regularization via

$$\psi(x, y) = |y - x| + \gamma x^2,$$

    which is (2γ)-strongly convex in x ✔
  - Assumption $\mathbf{A}_3'$: $\psi(\cdot, y)$ is not differentiable at y ✖
  - If $\mu_{Y|X}$ has a bounded density, then Assumption $\mathbf{A}_3$ ✔, with

$$|\mathbb{E}(\xi(x_1, Y) - \xi(x_2, Y) \,|\, X)| \leq 2(B + \gamma)|x_1 - x_2|.$$

Examples in ± 1-classification

- Often, $\psi(x, y) = \phi(yx)$, where $\phi : \mathbb{R} \to \mathbb{R}_+$ is convex.

Examples in ± 1-classification

- Often, $\psi(x, y) = \phi(yx)$, where $\phi : \mathbb{R} \to \mathbb{R}_+$ is convex.
- Logit loss: $\phi(u) = \ln_2(1 + e^{-u})$.

Examples in ± 1-classification

- Often, $\psi(x, y) = \phi(yx)$, where $\phi : \mathbb{R} \to \mathbb{R}_+$ is convex.
- Logit loss: $\phi(u) = \ln_2(1 + e^{-u})$.
- Not strongly convex $\to$ regularization via $\psi(x, y) = \phi(yx) + \gamma x^2$.

Examples in ± 1-classification

- Often, $\psi(x, y) = \phi(yx)$, where $\phi : \mathbb{R} \to \mathbb{R}_+$ is convex.
- Logit loss: $\phi(u) = \ln_2(1 + e^{-u})$.
- Not strongly convex $\to$ regularization via $\psi(x, y) = \phi(yx) + \gamma x^2$.
- Assumptions $\mathbf{A_1}$, $\mathbf{A_2}$, and $\mathbf{A_3'}$ ✔

Examples in ± 1-classification

- Often, $\psi(x, y) = \phi(yx)$, where $\phi : \mathbb{R} \to \mathbb{R}_+$ is convex.
- Logit loss: $\phi(u) = \ln_2(1 + e^{-u})$.
- Not strongly convex $\to$ regularization via $\psi(x, y) = \phi(yx) + \gamma x^2$.
- Assumptions $\mathbf{A_1}$, $\mathbf{A_2}$, and $\mathbf{A_3'}$ ✔
- Other examples:
  - ▷ Penalized sigmoid loss: $\psi(x, y) = (1 - \tanh(\beta yx)) + \gamma x^2$.

Examples in ± 1-classification

- Often, $\psi(x, y) = \phi(yx)$, where $\phi : \mathbb{R} \to \mathbb{R}_+$ is convex.
- Logit loss: $\phi(u) = \ln_2(1 + e^{-u})$.
- Not strongly convex $\to$ regularization via $\psi(x, y) = \phi(yx) + \gamma x^2$.
- Assumptions $\mathbf{A_1}$, $\mathbf{A_2}$, and $\mathbf{A_3'}$ ✔
- Other examples:
  - ▷ Penalized sigmoid loss: $\psi(x, y) = (1 - \tanh(\beta yx)) + \gamma x^2$.
  - ▷ $2(\gamma - \beta^2)$-strongly convex as soon as $\beta < \sqrt{\gamma}$.

Examples in ± 1-classification

- Often, $\psi(x, y) = \phi(yx)$, where $\phi : \mathbb{R} \to \mathbb{R}_+$ is convex.
- Logit loss: $\phi(u) = \ln_2(1 + e^{-u})$.
- Not strongly convex $\to$ regularization via $\psi(x, y) = \phi(yx) + \gamma x^2$.
- Assumptions $\mathbf{A_1}$, $\mathbf{A_2}$, and $\mathbf{A_3'}$ ✔
- Other examples:
  - Penalized sigmoid loss: $\psi(x, y) = (1 - \tanh(\beta yx)) + \gamma x^2$.
  - $2(\gamma - \beta^2)$-strongly convex as soon as $\beta < \sqrt{\gamma}$.
  - Assumptions $\mathbf{A_1}$, $\mathbf{A_2}$, and $\mathbf{A_3'}$ ✔

Examples in ± 1-classification

- Often, $\psi(x, y) = \phi(yx)$, where $\phi : \mathbb{R} \to \mathbb{R}_+$ is convex.
- Logit loss: $\phi(u) = \ln_2(1 + e^{-u})$.
- Not strongly convex $\to$ regularization via $\psi(x, y) = \phi(yx) + \gamma x^2$.
- Assumptions $\mathbf{A_1}$, $\mathbf{A_2}$, and $\mathbf{A_3'}$ ✔
- Other examples:
  - ▷ Penalized sigmoid loss: $\psi(x, y) = (1 - \tanh(\beta yx)) + \gamma x^2$.
  - ▷ $2(\gamma - \beta^2)$-strongly convex as soon as $\beta < \sqrt{\gamma}$.
  - ▷ Assumptions $\mathbf{A_1}$, $\mathbf{A_2}$, and $\mathbf{A_3'}$ ✔
  - ▷ Penalized lin-exp loss: $\psi(x, y) = \phi(yx) + \gamma x^2$, where

$$\phi(u) = \begin{cases} -u + 1 & \text{if } u \leq 0 \\ e^{-u} & \text{if } u > 0. \end{cases}$$

Examples in ± 1-classification

- Often, $\psi(x, y) = \phi(yx)$, where $\phi : \mathbb{R} \to \mathbb{R}_+$ is convex.
- Logit loss: $\phi(u) = \ln_2(1 + e^{-u})$.
- Not strongly convex $\to$ regularization via $\psi(x, y) = \phi(yx) + \gamma x^2$.
- Assumptions $\mathbf{A_1}$, $\mathbf{A_2}$, and $\mathbf{A_3'}$ ✔
- Other examples:
  - ▷ Penalized sigmoid loss: $\psi(x, y) = (1 - \tanh(\beta yx)) + \gamma x^2$.
  - ▷ $2(\gamma - \beta^2)$-strongly convex as soon as $\beta < \sqrt{\gamma}$.
  - ▷ Assumptions $\mathbf{A_1}$, $\mathbf{A_2}$, and $\mathbf{A_3'}$ ✔
  - ▷ Penalized lin-exp loss: $\psi(x, y) = \phi(yx) + \gamma x^2$, where

$$\phi(u) = \begin{cases} -u + 1 & \text{if } u \leq 0 \\ e^{-u} & \text{if } u > 0. \end{cases}$$

  - ▷ Assumptions $\mathbf{A_1}$, $\mathbf{A_2}$, and $\mathbf{A_3'}$ ✔

Take-home message

1. Our assumptions include a large variety of learning problems.

Take-home message

1. Our assumptions include a large variety of learning problems.
2. Regularization is important.

Take-home message

1. Our assumptions include a large variety of learning problems.
2. Regularization is important.
3. Regularized objectives are in action in the XGBoost system.

Two algorithms

The idea of gradient boosting

- Finding the infimum of the functional C over $\mathrm{lin}(\mathscr{F})$ is challenging.

The idea of gradient boosting

- Finding the infimum of the functional C over $\text{lin}(\mathscr{F})$ is challenging.
- It is an infinite-dimensional optimization problem.

The idea of gradient boosting

- Finding the infimum of the functional C over $\mathrm{lin}(\mathscr{F})$ is challenging.
- It is an infinite-dimensional optimization problem.

Gradient boosting algorithm

Locate the infimum by sequentially producing a linear combination of weak learners via a gradient-descent-type algorithm in $L^2(\mu_X)$.

The idea of gradient boosting

- Finding the infimum of the functional C over $\mathrm{lin}(\mathscr{F})$ is challenging.
- It is an infinite-dimensional optimization problem.

Gradient boosting algorithm

Locate the infimum by sequentially producing a linear combination of weak learners via a gradient-descent-type algorithm in $L^2(\mu_X)$.

- Fact 1: Under Assumption $\mathbf{A_1}$,

$$\inf_{F \in \mathrm{lin}(\mathscr{F})} C(F) = \inf_{F \in \overline{\mathrm{lin}(\mathscr{F})}} C(F).$$

The idea of gradient boosting

- Finding the infimum of the functional C over $\mathrm{lin}(\mathscr{F})$ is challenging.
- It is an infinite-dimensional optimization problem.

Gradient boosting algorithm

Locate the infimum by sequentially producing a linear combination of weak learners via a gradient-descent-type algorithm in $L^2(\mu_X)$.

- Fact 1: Under Assumption $\mathbf{A_1}$,

$$\inf_{F \in \mathrm{lin}(\mathscr{F})} C(F) = \inf_{F \in \overline{\mathrm{lin}(\mathscr{F})}} C(F).$$

- Fact 2: Under Assumption $\mathbf{A_2}$, there exists a unique $\bar{F} \in \overline{\mathrm{lin}(\mathscr{F})}$ (the boosting predictor) such that

$$C(\bar{F}) = \inf_{F \in \mathrm{lin}(\mathscr{F})} C(F).$$

Approach 1 (Mason et al., 2000)

- $\mathscr{F}$ = functions $f : \mathscr{X} \to \mathbb{R}$ such that $0 \in \mathscr{F}$, $f \in \mathscr{F} \Leftrightarrow -f \in \mathscr{F}$, and $\|f\|_{\mu_X} = 1$ for $f \neq 0$.

Approach 1 (Mason et al., 2000)

- $\mathscr{F}$ = functions $f : \mathscr{X} \to \mathbb{R}$ such that $0 \in \mathscr{F}$, $f \in \mathscr{F} \Leftrightarrow -f \in \mathscr{F}$, and $\|f\|_{\mu_X} = 1$ for $f \neq 0$.
- Example: all ± 1-trees in $\mathbb{R}^d$ with k terminal nodes (plus zero).

Approach 1 (Mason et al., 2000)

- $\mathscr{F}$ = functions $f : \mathscr{X} \to \mathbb{R}$ such that $0 \in \mathscr{F}$, $f \in \mathscr{F} \Leftrightarrow -f \in \mathscr{F}$, and $\|f\|_{\mu_X} = 1$ for $f \neq 0$.
- Example: all ± 1-trees in $\mathbb{R}^d$ with k terminal nodes (plus zero).
- Start with $F \in \text{lin}(\mathscr{F})$.

Approach 1 (Mason et al., 2000)

- $\mathscr{F}$ = functions $f : \mathscr{X} \to \mathbb{R}$ such that $0 \in \mathscr{F}$, $f \in \mathscr{F} \Leftrightarrow -f \in \mathscr{F}$, and $\|f\|_{\mu_X} = 1$ for $f \neq 0$.
- Example: all ± 1-trees in $\mathbb{R}^d$ with k terminal nodes (plus zero).
- Start with $F \in \mathrm{lin}(\mathscr{F})$.

? Which $f \in \mathscr{F}$ to add to F so that $C(F + wf)$ decreases at most?

Approach 1 (Mason et al., 2000)

- $\mathscr{F}$ = functions $f : \mathscr{X} \to \mathbb{R}$ such that $0 \in \mathscr{F}$, $f \in \mathscr{F} \Leftrightarrow -f \in \mathscr{F}$, and $\|f\|_{\mu_X} = 1$ for $f \neq 0$.
- Example: all ± 1-trees in $\mathbb{R}^d$ with k terminal nodes (plus zero).
- Start with $F \in \text{lin}(\mathscr{F})$.

? Which $f \in \mathscr{F}$ to add to F so that $C(F + wf)$ decreases at most?

- Knee-jerk reaction: take the opposite of the gradient of C at F.

Approach 1 (Mason et al., 2000)

- $\mathscr{F}$ = functions $f : \mathscr{X} \to \mathbb{R}$ such that $0 \in \mathscr{F}$, $f \in \mathscr{F} \Leftrightarrow -f \in \mathscr{F}$, and $\|f\|_{\mu_X} = 1$ for $f \neq 0$.
- Example: all ± 1-trees in $\mathbb{R}^d$ with k terminal nodes (plus zero).
- Start with $F \in \text{lin}(\mathscr{F})$.

? Which $f \in \mathscr{F}$ to add to F so that $C(F + wf)$ decreases at most?

- Knee-jerk reaction: take the opposite of the gradient of C at F.

✖ Impossible, since our new function has to live in $\mathscr{F}$.

Approach 1 (Mason et al., 2000)

- $\mathscr{F}$ = functions $f : \mathscr{X} \to \mathbb{R}$ such that $0 \in \mathscr{F}$, $f \in \mathscr{F} \Leftrightarrow -f \in \mathscr{F}$, and $\|f\|_{\mu_X} = 1$ for $f \neq 0$.
- Example: all ± 1-trees in $\mathbb{R}^d$ with k terminal nodes (plus zero).
- Start with $F \in \text{lin}(\mathscr{F})$.

? Which $f \in \mathscr{F}$ to add to F so that $C(F + wf)$ decreases at most?

- Knee-jerk reaction: take the opposite of the gradient of C at F.

✖ Impossible, since our new function has to live in $\mathscr{F}$.

- Solution: start from the approximate identity

$$C(F) - C(F + wf) \approx -w\langle \nabla C(F), f\rangle_{\mu_X}$$

and choose $f \in \mathscr{F}$ that maximizes $-\langle \nabla C(F), f\rangle_{\mu_X}$.

Approach 1 (Mason et al., 2000)

- $\mathscr{F}$ = functions $f : \mathscr{X} \to \mathbb{R}$ such that $0 \in \mathscr{F}$, $f \in \mathscr{F} \Leftrightarrow -f \in \mathscr{F}$, and $\|f\|_{\mu_X} = 1$ for $f \neq 0$.
- Example: all ± 1-trees in $\mathbb{R}^d$ with k terminal nodes (plus zero).
- Start with $F \in \text{lin}(\mathscr{F})$.

? Which $f \in \mathscr{F}$ to add to F so that $C(F + wf)$ decreases at most?

- Knee-jerk reaction: take the opposite of the gradient of C at F.

✖ Impossible, since our new function has to live in $\mathscr{F}$.

- Solution: start from the approximate identity

$$C(F) - C(F + wf) \approx -w\langle \nabla C(F), f\rangle_{\mu_X}$$

and choose $f \in \mathscr{F}$ that maximizes $-\langle \nabla C(F), f\rangle_{\mu_X}$.

✔ General case: choose $f \in \mathscr{F}$ that maximizes $-\mathbb{E}\xi(F(X), Y)f(X)$.

Gradient boosting Algorithm 1

1: **Require** $(w_t)_t$ a sequence of positive real numbers.

2: **Set** $t = 0$ and start with $F_0 \in \mathscr{F}$.

3: **Compute**

$$f_{t+1} \in \arg\max_{f \in \mathscr{F}} - \mathbb{E}\xi(F_t(X), Y)f(X)$$

and **let** $F_{t+1} = F_t + w_{t+1} f_{t+1}$.

4: **Take** $t \leftarrow t + 1$ and **go** to step 3.

Some comments

- The algorithm performs a gradient-type descent in $L^2(\mu_X)$.

Some comments

- The algorithm performs a gradient-type descent in $L^2(\mu_X)$.
- Difference: the descent direction belongs to $\mathscr{F}$.

Some comments

- The algorithm performs a gradient-type descent in $L^2(\mu_X)$.
- Difference: the descent direction belongs to $\mathscr{F}$.
- If ψ is continuously differentiable in its first argument, then

$$-\mathbb{E}\xi(F_t(X), Y)f(X) = -\langle \nabla C(F_t), f \rangle_{\mu_X},$$

and, for $\nabla C(F_t) \neq 0$,

$$\frac{-\nabla C(F_t)}{\|\nabla C(F_t)\|_{\mu_X}} = \arg\max_{F \in L^2(\mu_X) : \|F\|_{\mu_X} = 1} - \langle \nabla C(F_t), F \rangle_{\mu_X}.$$

Some comments

- The algorithm performs a gradient-type descent in $L^2(\mu_X)$.
- Difference: the descent direction belongs to $\mathscr{F}$.
- If ψ is continuously differentiable in its first argument, then

$$-\mathbb{E}\xi(F_t(X), Y)f(X) = -\langle \nabla C(F_t), f\rangle_{\mu_X},$$

and, for $\nabla C(F_t) \neq 0$,

$$\frac{-\nabla C(F_t)}{\|\nabla C(F_t)\|_{\mu_X}} = \arg\max_{F \in L^2(\mu_X) : \|F\|_{\mu_X} = 1} - \langle \nabla C(F_t), F\rangle_{\mu_X}.$$

- Rationale: at each step, Algorithm 1 mimics the computation of the negative gradient:

$$f_{t+1} \in \arg\max_{f \in \mathscr{F}} - \langle \nabla C(F_t), f\rangle_{\mu_X}.$$

Some comments

- Empirical case: the descent step takes the form

$$f_{t+1} \in \arg\max_{f \in \mathscr{F}} - \frac{1}{n} \sum_{i=1}^{n} \nabla C(F_t)(X_i) \cdot f(X_i).$$

Some comments

- Empirical case: the descent step takes the form

$$f_{t+1} \in \arg\max_{f \in \mathscr{F}} - \frac{1}{n} \sum_{i=1}^{n} \nabla C(F_t)(X_i) \cdot f(X_i).$$

- Finding this optimum is a non-trivial problem $\rightarrow$ CART strategy.

Some comments

- Empirical case: the descent step takes the form

$$f_{t+1} \in \arg\max_{f \in \mathscr{F}} -\frac{1}{n}\sum_{i=1}^{n} \nabla C(F_t)(X_i) \cdot f(X_i).$$

- Finding this optimum is a non-trivial problem $\rightarrow$ CART strategy.
- The sequence $(w_t)_t$ should be carefully chosen for convergence.

Some comments

- Empirical case: the descent step takes the form

$$f_{t+1} \in \arg\max_{f \in \mathscr{F}} - \frac{1}{n} \sum_{i=1}^{n} \nabla C(F_t)(X_i) \cdot f(X_i).$$

- Finding this optimum is a non-trivial problem $\rightarrow$ CART strategy.
- The sequence $(w_t)_t$ should be carefully chosen for convergence.
- The algorithm is run forever: no stopping at this stage.

Some comments

- Empirical case: the descent step takes the form

$$f_{t+1} \in \arg\max_{f \in \mathscr{F}} -\frac{1}{n}\sum_{i=1}^{n} \nabla C(F_t)(X_i) \cdot f(X_i).$$

- Finding this optimum is a non-trivial problem $\to$ CART strategy.
- The sequence $(w_t)_t$ should be carefully chosen for convergence.
- The algorithm is run forever: no stopping at this stage.
- Question: is it true that

$$\lim_{t\to\infty} C(F_t) = \inf_{F \in \mathrm{lin}(\mathscr{F})} C(F) \quad ?$$

Approach 2 (Friedman, 2001)

- $\mathscr{P}$ = functions $f : \mathscr{X} \to \mathbb{R}$ such that $f \in \mathscr{P} \Leftrightarrow -f \in \mathscr{P}$, and $af \in \mathscr{P}$ for all $(a, f) \in \mathbb{R} \times \mathscr{P}$.

Approach 2 (Friedman, 2001)

- $\mathscr{P}$ = functions $f : \mathscr{X} \to \mathbb{R}$ such that $f \in \mathscr{P} \Leftrightarrow -f \in \mathscr{P}$, and $af \in \mathscr{P}$ for all $(a, f) \in \mathbb{R} \times \mathscr{P}$.
- Example: all trees in $\mathbb{R}^d$ with k terminal nodes.

Approach 2 (Friedman, 2001)

- $\mathscr{P}$ = functions $f : \mathscr{X} \to \mathbb{R}$ such that $f \in \mathscr{P} \Leftrightarrow -f \in \mathscr{P}$, and $af \in \mathscr{P}$ for all $(a, f) \in \mathbb{R} \times \mathscr{P}$.
- Example: all trees in $\mathbb{R}^d$ with k terminal nodes.
- Key idea: replace

$$f_{t+1} \in \arg\max_{f \in \mathscr{F}} -\mathbb{E}\xi(F_t(X), Y)f(X)$$

Approach 2 (Friedman, 2001)

- $\mathscr{P}$ = functions $f : \mathscr{X} \to \mathbb{R}$ such that $f \in \mathscr{P} \Leftrightarrow -f \in \mathscr{P}$, and $af \in \mathscr{P}$ for all $(a, f) \in \mathbb{R} \times \mathscr{P}$.
- Example: all trees in $\mathbb{R}^d$ with k terminal nodes.
- Key idea: replace

$$f_{t+1} \in \arg\max_{f \in \mathscr{F}} -\mathbb{E}\xi(F_t(X), Y)f(X)$$

by

$$f_{t+1} \in \arg\min_{f \in \mathscr{P}} \mathbb{E}(-\xi(F_t(X), Y) - f(X))^2.$$

Approach 2 (Friedman, 2001)

- $\mathscr{P}$ = functions $f : \mathscr{X} \to \mathbb{R}$ such that $f \in \mathscr{P} \Leftrightarrow -f \in \mathscr{P}$, and $af \in \mathscr{P}$ for all $(a, f) \in \mathbb{R} \times \mathscr{P}$.
- Example: all trees in $\mathbb{R}^d$ with k terminal nodes.
- Key idea: replace

$$f_{t+1} \in \arg\max_{f \in \mathscr{F}} -\mathbb{E}\xi(F_t(X), Y)f(X)$$

by

$$f_{t+1} \in \arg\min_{f \in \mathscr{P}} \mathbb{E}(-\xi(F_t(X), Y) - f(X))^2.$$

- Equivalently,

$$f_{t+1} \in \arg\min_{f \in \mathscr{P}} \big(2\mathbb{E}\xi(F_t(X), Y)f(X) + \|f\|^2_{\mu_X}\big).$$

Gradient boosting Algorithm 2

1: **Require** ν a positive real number.

2: **Set** $t = 0$ and start with $F_0 \in \mathscr{P}$.

3: **Compute**

$$f_{t+1} \in \arg\min_{f \in \mathscr{P}} \big(2\mathbb{E}\xi(F_t(X), Y) f(X) + \|f\|^2_{\mu_X}\big)$$

and **let** $F_{t+1} = F_t + \nu f_{t+1}$.

4: **Take** $t \leftarrow t + 1$ and **go** to step 3.

Some comments

- The step size ν is kept fixed during the iterations.

Some comments

- The step size ν is kept fixed during the iterations.
- Empirical setting with ψ continuously differentiable:

$$f_{t+1} \in \arg\min_{f \in \mathscr{P}} \frac{1}{n} \sum_{i=1}^{n} (-\nabla C(F_t)(X_i) - f(X_i))^2.$$

Some comments

- The step size ν is kept fixed during the iterations.
- Empirical setting with ψ continuously differentiable:

$$f_{t+1} \in \arg\min_{f \in \mathscr{P}} \frac{1}{n} \sum_{i=1}^{n} (-\nabla C(F_t)(X_i) - f(X_i))^2.$$

- f_{t+1} is fitted to the negative gradient instances $-\nabla C(F_t)(X_i)$.

Some comments

- The step size ν is kept fixed during the iterations.
- Empirical setting with ψ continuously differentiable:

$$f_{t+1} \in \arg\min_{f \in \mathscr{P}} \frac{1}{n} \sum_{i=1}^{n} (-\nabla C(F_t)(X_i) - f(X_i))^2.$$

- f_{t+1} is fitted to the negative gradient instances $-\nabla C(F_t)(X_i)$.
- Example: when $\psi(x, y) = (y - x)^2/2$, then

$$-\nabla C(F_t)(X_i) = Y_i - F_t(X_i).$$

Some comments

- The step size ν is kept fixed during the iterations.
- Empirical setting with ψ continuously differentiable:

$$f_{t+1} \in \arg\min_{f \in \mathscr{P}} \frac{1}{n} \sum_{i=1}^{n} (-\nabla C(F_t)(X_i) - f(X_i))^2.$$

- f_{t+1} is fitted to the negative gradient instances $-\nabla C(F_t)(X_i)$.
- Example: when $\psi(x, y) = (y - x)^2/2$, then

$$-\nabla C(F_t)(X_i) = Y_i - F_t(X_i).$$

- This is at the origin of gradient boosting.

Convergence

Algorithm 1

Step sizes: we take $w_0 > 0$ arbitrarily and set

$$w_{t+1} = \min\big(w_t, -(2L)^{-1}\mathbb{E}\xi(F_t(X), Y) f_{t+1}(X)\big), \quad t \geq 0.$$

Algorithm 1

Step sizes: we take $w_0 > 0$ arbitrarily and set

$$w_{t+1} = \min\big(w_t, -(2L)^{-1}\mathbb{E}\xi(F_t(X), Y)f_{t+1}(X)\big), \quad t \geq 0.$$

Theorem

Assume that Assumptions $\mathbf{A_1}$ *and* $\mathbf{A_3}$ *are satisfied. Then*

$$\lim_{t\to\infty} C(F_t) = \inf_{F\in\mathrm{lin}(\mathscr{F})} C(F).$$

Algorithm 1

Step sizes: we take $w_0 > 0$ arbitrarily and set

$$w_{t+1} = \min\left(w_t, -(2L)^{-1}\mathbb{E}\xi(F_t(X), Y)f_{t+1}(X)\right), \quad t \geq 0.$$

Theorem

Assume that Assumptions $\mathbf{A_1}$ *and* $\mathbf{A_3}$ *are satisfied. Then*

$$\lim_{t\to\infty} C(F_t) = \inf_{F\in\mathrm{lin}(\mathscr{F})} C(F).$$

- The result holds without Assumption $\mathbf{A_2}$.

Algorithm 1

Step sizes: we take $w_0 > 0$ arbitrarily and set

$$w_{t+1} = \min\left(w_t, -(2L)^{-1}\mathbb{E}\xi(F_t(X), Y)f_{t+1}(X)\right), \quad t \geq 0.$$

Theorem

Assume that Assumptions $\mathbf{A_1}$ *and* $\mathbf{A_3}$ *are satisfied. Then*

$$\lim_{t\to\infty} C(F_t) = \inf_{F\in\mathrm{lin}(\mathscr{F})} C(F).$$

- The result holds without Assumption $\mathbf{A_2}$.
- With $\mathbf{A_2}$, there is a unique boosting predictor $\bar{F} \in \overline{\mathrm{lin}(\mathscr{F})}$ such that

$$C(\bar{F}) = \inf_{F\in\mathrm{lin}(\mathscr{F})} C(F).$$

Algorithm 1

Step sizes: we take $w_0 > 0$ arbitrarily and set

$$w_{t+1} = \min\left(w_t, -(2L)^{-1}\mathbb{E}\xi(F_t(X), Y)f_{t+1}(X)\right), \quad t \geq 0.$$

Theorem

Assume that Assumptions $\mathbf{A_1}$ *and* $\mathbf{A_3}$ *are satisfied. Then*

$$\lim_{t\to\infty} C(F_t) = \inf_{F\in \mathrm{lin}(\mathscr{F})} C(F).$$

- The result holds without Assumption $\mathbf{A_2}$.
- With $\mathbf{A_2}$, there is a unique boosting predictor $\bar{F} \in \overline{\mathrm{lin}(\mathscr{F})}$ such that

$$C(\bar{F}) = \inf_{F\in \mathrm{lin}(\mathscr{F})} C(F).$$

- The theorem guarantees that $\lim_{t\to\infty} C(F_t) = C(\bar{F})$.

Mathematical machinery

Lemma

Assume that Assumptions $\mathbf{A_1}$ *and* $\mathbf{A_3}$ *are satisfied. Then*

$$C(F_t) - C(F_{t+1}) \geq L w_{t+1}^2.$$

In particular, $\lim_{t\to\infty} C(F_t) = \inf_k C(F_k)$.

Mathematical machinery

Lemma

Assume that Assumptions $\mathbf{A_1}$ *and* $\mathbf{A_3}$ *are satisfied. Then*

$$C(F_t) - C(F_{t+1}) \geq L w_{t+1}^2.$$

In particular, $\lim_{t\to\infty} C(F_t) = \inf_k C(F_k)$.

Corollary

Assume that $\overline{\mathrm{lin}(\mathscr{F})} = L^2(\mu_X)$*. Assume, in addition, that Assumptions* $\mathbf{A_1}$, $\mathbf{A_2}$, *and* $\mathbf{A_3'}$ *are satisfied. Then*

$$\lim_{t\to\infty} \|F_t - \bar{F}\|_{\mu_X} = 0,$$

where

$$\bar{F} = \arg\min_{F \in L^2(\mu_X)} C(F).$$

Algorithm 2

Theorem

Assume that Assumptions $\mathbf{A_1}$-$\mathbf{A_3}$ *are satisfied, with* $0 < \nu < 1/(2L)$. *Then*

$$\lim_{t \to \infty} C(F_t) = \inf_{F \in \mathrm{lin}(\mathscr{P})} C(F).$$

Algorithm 2

Theorem

Assume that Assumptions $\mathbf{A_1}$*-*$\mathbf{A_3}$ *are satisfied, with* $0 < \nu < 1/(2L)$*. Then*

$$\lim_{t \to \infty} C(F_t) = \inf_{F \in \mathrm{lin}(\mathscr{P})} C(F).$$

- ▷ The result requires Assumption $\mathbf{A_2}$.

Algorithm 2

Theorem

Assume that Assumptions $\mathbf{A_1}$*-*$\mathbf{A_3}$ *are satisfied, with* $0 < \nu < 1/(2L)$*. Then*

$$\lim_{t\to\infty} C(F_t) = \inf_{F\in \mathrm{lin}(\mathscr{P})} C(F).$$

- The result requires Assumption $\mathbf{A_2}$.
- The theorem guarantees that $\lim_{t\to\infty} C(F_t) = C(\bar{F})$.

Algorithm 2

Theorem

Assume that Assumptions $\mathbf{A_1}$-$\mathbf{A_3}$ *are satisfied, with* $0 < \nu < 1/(2L)$. *Then*

$$\lim_{t\to\infty} C(F_t) = \inf_{F\in \mathrm{lin}(\mathscr{P})} C(F).$$

- The result requires Assumption $\mathbf{A_2}$.
- The theorem guarantees that $\lim_{t\to\infty} C(F_t) = C(\bar{F})$.
- If $\overline{\mathrm{lin}(\mathscr{P})} = L^2(\mu_X)$ and $\mathbf{A}_3'$ is satisfied, then

$$\lim_{t\to\infty} \|F_t - \bar{F}\|_{\mu_X} = 0,$$

  where

$$\bar{F} = \arg\min_{F\in L^2(\mu_X)} C(F).$$

Discussion

- Empirical setting: both algorithms track the infimum of

$$C_n(F) = \frac{1}{n}\sum_{i=1}^{n}\psi(F(X_i), Y_i)$$

over the linear combinations of weak learners.

Discussion

- Empirical setting: both algorithms track the infimum of

$$C_n(F) = \frac{1}{n}\sum_{i=1}^{n}\psi(F(X_i), Y_i)$$

  over the linear combinations of weak learners.
- This task is achieved by sequentially constructing linear combinations.

Discussion

- Empirical setting: both algorithms track the infimum of

$$C_n(F) = \frac{1}{n}\sum_{i=1}^{n}\psi(F(X_i), Y_i)$$

  over the linear combinations of weak learners.
- This task is achieved by sequentially constructing linear combinations.
- F_t and $\bar{F}_n$ are functions of the data set $\mathscr{D}_n$.

Discussion

- Empirical setting: both algorithms track the infimum of

$$C_n(F) = \frac{1}{n} \sum_{i=1}^{n} \psi(F(X_i), Y_i)$$

  over the linear combinations of weak learners.
- This task is achieved by sequentially constructing linear combinations.
- F_t and $\bar{F}_n$ are functions of the data set $\mathscr{D}_n$.
- So far: no information on the statistical behavior of $\bar{F}_n$.

Discussion

- Empirical setting: both algorithms track the infimum of

$$C_n(F) = \frac{1}{n}\sum_{i=1}^{n}\psi(F(X_i), Y_i)$$

  over the linear combinations of weak learners.
- This task is achieved by sequentially constructing linear combinations.
- F_t and $\bar{F}_n$ are functions of the data set $\mathscr{D}_n$.
- So far: no information on the statistical behavior of $\bar{F}_n$.
- Question: probabilistic properties of $\bar{F}_n$ as $n \to \infty$?

Discussion

- Catastrophic situations can happen $\rightarrow$ "size" of $\mathrm{lin}(\mathscr{F})$ or $\mathrm{lin}(\mathscr{P})$.

Discussion

- Catastrophic situations can happen $\to$ "size" of $\mathrm{lin}(\mathscr{F})$ or $\mathrm{lin}(\mathscr{P})$.
- Example: $\psi(x, y) = (y - x)^2$ and $\mathscr{F}$ = all trees with $d + 1$ leaves. Then

$$\lim_{t \to \infty} C_n(F_t) = C_n(\bar{F}_n),$$

where

$$\bar{F}_n = \arg\min_{F \in L^2(P_n)} C_n(F).$$

Discussion

- Catastrophic situations can happen $\rightarrow$ "size" of $\mathrm{lin}(\mathscr{F})$ or $\mathrm{lin}(\mathscr{P})$.
- Example: $\psi(x, y) = (y - x)^2$ and $\mathscr{F}$ = all trees with $d + 1$ leaves. Then
$$\lim_{t \to \infty} C_n(F_t) = C_n(\bar{F}_n),$$
where
$$\bar{F}_n = \arg\min_{F \in L^2(P_n)} C_n(F).$$
- Overfitting: $\bar{F}_n$ reproduces the data.

Discussion

- Catastrophic situations can happen $\rightarrow$ "size" of $\text{lin}(\mathscr{F})$ or $\text{lin}(\mathscr{P})$.
- Example: $\psi(x, y) = (y - x)^2$ and $\mathscr{F}$ = all trees with $d + 1$ leaves. Then
$$\lim_{t \to \infty} C_n(F_t) = C_n(\bar{F}_n),$$
where
$$\bar{F}_n = \arg\min_{F \in L^2(P_n)} C_n(F).$$
- Overfitting: $\bar{F}_n$ reproduces the data.
- No chance that $\bar{F}_n$ converges to $F^\star(x) = \mathbb{E}(Y|X = x)$.

Discussion

- Catastrophic situations can happen $\to$ "size" of $\text{lin}(\mathscr{F})$ or $\text{lin}(\mathscr{P})$.
- Example: $\psi(x, y) = (y - x)^2$ and $\mathscr{F}$ = all trees with $d + 1$ leaves. Then

$$\lim_{t\to\infty} C_n(F_t) = C_n(\bar{F}_n),$$

  where

$$\bar{F}_n = \arg\min_{F\in L^2(P_n)} C_n(F).$$

- Overfitting: $\bar{F}_n$ reproduces the data.
- No chance that $\bar{F}_n$ converges to $F^\star(x) = \mathbb{E}(Y|X = x)$.
- Classical solution: early stopping.

Can we avoid early stopping?

- Yes, under appropriate conditions.

Can we avoid early stopping?

- Yes, under appropriate conditions.
- Problem: the minimizations are performed over vector spaces.

Can we avoid early stopping?

- Yes, under appropriate conditions.
- Problem: the minimizations are performed over vector spaces.

✖ No question of imposing constraints on the coefficients.

Can we avoid early stopping?

- Yes, under appropriate conditions.
- Problem: the minimizations are performed over vector spaces.

✖ No question of imposing constraints on the coefficients.

✔ Solution: carefully constraint the "complexity" of the vector spaces.

Can we avoid early stopping?

- Yes, under appropriate conditions.
- Problem: the minimizations are performed over vector spaces.

✖ No question of imposing constraints on the coefficients.

✔ Solution: carefully constraint the "complexity" of the vector spaces.

- Importance of having a strongly convex risk functional to minimize.

Can we avoid early stopping?

- Yes, under appropriate conditions.
- Problem: the minimizations are performed over vector spaces.

✖ No question of imposing constraints on the coefficients.

✔ Solution: carefully constraint the "complexity" of the vector spaces.

- Importance of having a strongly convex risk functional to minimize.

✔ Solution: possible regularization with an L^2-type penalty.

Large sample properties

Context

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.

Context

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.
- $\mathscr{X}$ is a compact subset of $\mathbb{R}^d$.

Context

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.
- $\mathscr{X}$ is a compact subset of $\mathbb{R}^d$.
- Each X_i has a density g on $\mathscr{X}$, with

$$0 < \inf_{\mathscr{X}} g \leq \sup_{\mathscr{X}} g < \infty.$$

Context

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.
- $\mathscr{X}$ is a compact subset of $\mathbb{R}^d$.
- Each X_i has a density g on $\mathscr{X}$, with

$$0 < \inf_{\mathscr{X}} g \leq \sup_{\mathscr{X}} g < \infty.$$

- We concentrate on Algorithm 1.

Context

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.
- $\mathscr{X}$ is a compact subset of $\mathbb{R}^d$.
- Each X_i has a density g on $\mathscr{X}$, with

$$0 < \inf_{\mathscr{X}} g \leq \sup_{\mathscr{X}} g < \infty.$$

- We concentrate on Algorithm 1.
- Weak learners: a finite class $\mathscr{F}_n$ of ± 1-values simple functions on $\mathscr{X}$.

Context

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.
- $\mathscr{X}$ is a compact subset of $\mathbb{R}^d$.
- Each X_i has a density g on $\mathscr{X}$, with

$$0 < \inf_{\mathscr{X}} g \leq \sup_{\mathscr{X}} g < \infty.$$

- We concentrate on Algorithm 1.
- Weak learners: a finite class $\mathscr{F}_n$ of ± 1-values simple functions on $\mathscr{X}$.
- Example: a finite class of trees with k leaves.

Context

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.
- $\mathscr{X}$ is a compact subset of $\mathbb{R}^d$.
- Each X_i has a density g on $\mathscr{X}$, with

$$0 < \inf_{\mathscr{X}} g \leq \sup_{\mathscr{X}} g < \infty.$$

- We concentrate on Algorithm 1.
- Weak learners: a finite class $\mathscr{F}_n$ of ± 1-values simple functions on $\mathscr{X}$.
- Example: a finite class of trees with k leaves.
- Consequence: any $F \in \text{lin}(\mathscr{F}_n)$ takes the form $F = \sum_{j=1}^{N} \alpha_j \mathbb{1}_{A_j^n}$.

Context

- Observations: $\mathscr{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ in $\mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^d \times \mathbb{R}$.
- $\mathscr{X}$ is a compact subset of $\mathbb{R}^d$.
- Each X_i has a density g on $\mathscr{X}$, with

$$0 < \inf_{\mathscr{X}} g \leq \sup_{\mathscr{X}} g < \infty.$$

- We concentrate on Algorithm 1.
- Weak learners: a finite class $\mathscr{F}_n$ of ± 1-values simple functions on $\mathscr{X}$.
- Example: a finite class of trees with k leaves.
- Consequence: any $F \in \mathrm{lin}(\mathscr{F}_n)$ takes the form $F = \sum_{j=1}^N \alpha_j \mathbb{1}_{A_j^n}$.
- Assumption: there exists $(v_n)_n$ such that $\min_{1 \leq j \leq N} \lambda(A_j^n) \geq v_n$.

Objective

- Objective: minimize over $\mathrm{lin}(\mathscr{F}_n)$ the empirical risk functional

$$C_n(F) = \frac{1}{n}\sum_{i=1}^{n} \psi(F(X_i), Y_i),$$

where $\psi(x, y) = \phi(x, y) + \gamma_n x^2$.

Objective

- Objective: minimize over $\mathrm{lin}(\mathscr{F}_n)$ the empirical risk functional

$$C_n(F) = \frac{1}{n}\sum_{i=1}^{n} \psi(F(X_i), Y_i),$$

  where $\psi(x, y) = \phi(x, y) + \gamma_n x^2$.
- Differently:

$$C_n(F) = A_n(F) + \gamma_n \|F\|_{P_n}^2,$$

  where

$$A_n(F) = \frac{1}{n}\sum_{i=1}^{n} \phi(F(X_i), Y_i).$$

Objective

- Objective: minimize over $\text{lin}(\mathscr{F}_n)$ the empirical risk functional

$$C_n(F) = \frac{1}{n}\sum_{i=1}^{n}\psi(F(X_i), Y_i),$$

  where $\psi(x, y) = \phi(x, y) + \gamma_n x^2$.
- Differently:

$$C_n(F) = A_n(F) + \gamma_n \|F\|_{P_n}^2,$$

  where

$$A_n(F) = \frac{1}{n}\sum_{i=1}^{n}\phi(F(X_i), Y_i).$$

- The strong convexity Assumption $\mathbf{A_2}$ is satisfied.

Consistency of the boosting predictor

- Boosting predictor: $\bar{F}_n = \arg\min_{F \in \mathrm{lin}(\mathscr{F}_n)} C_n(F)$.

Consistency of the boosting predictor

- Boosting predictor: $\bar{F}_n = \arg\min_{F \in \text{lin}(\mathscr{F}_n)} C_n(F)$.
- Whenever Assumption $\mathbf{A_3}$ is satisfied, $\lim_{t\to\infty} C_n(F_t) = C_n(\bar{F}_n)$.

Consistency of the boosting predictor

- Boosting predictor: $\bar{F}_n = \arg\min_{F \in \text{lin}(\mathscr{F}_n)} C_n(F)$.
- Whenever Assumption $\mathbf{A_3}$ is satisfied, $\lim_{t\to\infty} C_n(F_t) = C_n(\bar{F}_n)$.
- Objective: prove that $\lim_{n\to\infty} A(\bar{F}_n) = A(F^\star)$, where

$$A(F) = \mathbb{E}\phi(F(X_1), Y_1) \quad \text{and} \quad F^\star \in \arg\min_{F \in L^2(P)} A(F).$$

Consistency of the boosting predictor

- Boosting predictor: $\bar{F}_n = \arg\min_{F \in \mathrm{lin}(\mathscr{F}_n)} C_n(F)$.
- Whenever Assumption $\mathbf{A_3}$ is satisfied, $\lim_{t\to\infty} C_n(F_t) = C_n(\bar{F}_n)$.
- Objective: prove that $\lim_{n\to\infty} A(\bar{F}_n) = A(F^\star)$, where

$$A(F) = \mathbb{E}\phi(F(X_1), Y_1) \quad \text{and} \quad F^\star \in \arg\min_{F \in L^2(P)} A(F).$$

- Example 1: $F^\star(x) = \mathbb{E}(Y|X = x)$ with $\phi(x, y) = (y - x)^2$.

Consistency of the boosting predictor

- Boosting predictor: $\bar{F}_n = \arg\min_{F \in \text{lin}(\mathscr{F}_n)} C_n(F)$.
- Whenever Assumption $\mathbf{A_3}$ is satisfied, $\lim_{t\to\infty} C_n(F_t) = C_n(\bar{F}_n)$.
- Objective: prove that $\lim_{n\to\infty} A(\bar{F}_n) = A(F^\star)$, where

$$A(F) = \mathbb{E}\phi(F(X_1), Y_1) \quad \text{and} \quad F^\star \in \arg\min_{F \in L^2(P)} A(F).$$

- Example 1: $F^\star(x) = \mathbb{E}(Y|X=x)$ with $\phi(x,y) = (y-x)^2$.
- Example 2: $F^\star(x) = \log(\frac{\eta(x)}{1-\eta(x)})$ with $\phi(x,y) = \log_2(1+e^{-yx})$.

Consistency of the boosting predictor

- Boosting predictor: $\bar{F}_n = \arg\min_{F \in \text{lin}(\mathscr{F}_n)} C_n(F)$.
- Whenever Assumption $\mathbf{A_3}$ is satisfied, $\lim_{t\to\infty} C_n(F_t) = C_n(\bar{F}_n)$.
- Objective: prove that $\lim_{n\to\infty} A(\bar{F}_n) = A(F^\star)$, where

$$A(F) = \mathbb{E}\phi(F(X_1), Y_1) \quad \text{and} \quad F^\star \in \arg\min_{F \in L^2(P)} A(F).$$

- Example 1: $F^\star(x) = \mathbb{E}(Y|X = x)$ with $\phi(x, y) = (y - x)^2$.
- Example 2: $F^\star(x) = \log(\frac{\eta(x)}{1-\eta(x)})$ with $\phi(x, y) = \log_2(1 + e^{-yx})$.
- What we know so far:

$$A_n(\bar{F}_n) + \gamma_n \|\bar{F}_n\|^2_{P_n} - A(F^\star) = \inf_{F \in \text{lin}(\mathscr{F}_n)} \big(A_n(F) + \gamma_n \|F\|^2_{P_n} - A(F^\star)\big).$$

Main result

Assumption $\mathbf{A_4}$

For all $p \geq 0$, there exists a constant $\zeta(p) > 0$ such that, for all $(x_1, x_2, y) \in \mathbb{R}^2 \times \mathscr{Y}$ with $\max(|x_1|, |x_2|) \leq p$,

$$|\phi(x_1, y) - \phi(x_2, y)| \leq \zeta(p)|x_1 - x_2|.$$

Main result

Assumption $\mathbf{A_4}$

For all $p \geq 0$, there exists a constant $\zeta(p) > 0$ such that, for all $(x_1, x_2, y) \in \mathbb{R}^2 \times \mathscr{Y}$ with $\max(|x_1|, |x_2|) \leq p$,

$$|\phi(x_1, y) - \phi(x_2, y)| \leq \zeta(p)|x_1 - x_2|.$$

Theorem

Assume that Assumptions $\mathbf{A_3}$ and $\mathbf{A_4}$ are satisfied, and that $F^\star$ is bounded. Assume, in addition, that $\mathrm{diam}(A^n(X)) \to 0$ in probability as $n \to \infty$. Then, provided $\gamma_n \to 0$, $N \to \infty$, $\frac{\log N}{n v_n} \to 0$, and

$$\frac{1}{\sqrt{n v_n \gamma_n}} \zeta\left(\sqrt{\frac{2\bar{\phi}}{v_n \gamma_n \inf_{\mathscr{X}} g}}\right) \to 0,$$

we have $\lim_{n\to\infty} \mathbb{E} A(\bar{F}_n) = A(F^\star)$.

Discussion

- Gradient boosting does not always overfit.

Discussion

- Gradient boosting does not always overfit.
- If the function $\phi(\cdot, y)$ is already α-strongly convex: ✔

Discussion

- Gradient boosting does not always overfit.
- If the function $\phi(\cdot, y)$ is already α-strongly convex: ✔
- Example:

Discussion

- Gradient boosting does not always overfit.
- If the function $\phi(\cdot, y)$ is already α-strongly convex: ✔
- Example:
  - ▷ $\mathscr{X} = [0, 1]^d$;

Discussion

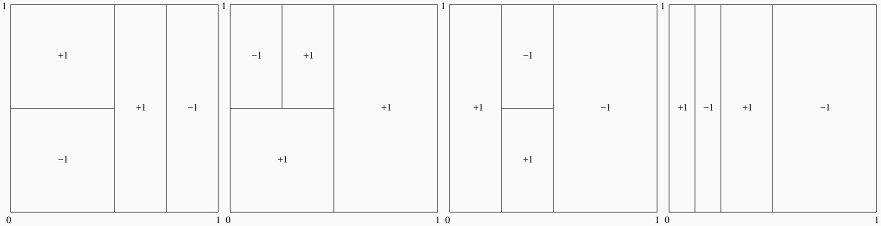
- Gradient boosting does not always overfit.
- If the function $\phi(\cdot, y)$ is already α-strongly convex: ✔
- Example:
  - ▷ $\mathscr{X} = [0,1]^d$;
  - ▷ $\mathscr{F}_n$ = all trees on $[0,1]^d$ with k_n leaves;

Discussion

- Gradient boosting does not always overfit.
- If the function $\phi(\cdot, y)$ is already α-strongly convex: ✔
- Example:
  - ▷ $\mathscr{X} = [0, 1]^d$;
  - ▷ $\mathscr{F}_n$ = all trees on $[0, 1]^d$ with k_n leaves;
  - ▷ Cuts are orthogonal and located at the middle of the cells;

Discussion

- Gradient boosting does not always overfit.
- If the function $\phi(\cdot, y)$ is already α-strongly convex: ✔
- Example:
  - ▷ $\mathscr{X} = [0,1]^d$;
  - ▷ $\mathscr{F}_n =$ all trees on $[0,1]^d$ with k_n leaves;
  - ▷ Cuts are orthogonal and located at the middle of the cells;
  - ▷ Although combinatorially rich, this family of trees is finite.



Example

- Any $F \in \mathrm{lin}(\mathscr{F}_n)$ takes the form $F = \sum_{j=1}^{N} \alpha_j \mathbb{1}_{A_j^n}$.
- $N \leq 2^{dk_n}$ and the A_j^n form a regular grid over $[0,1]^d$.

Example

- Any $F \in \mathrm{lin}(\mathscr{F}_n)$ takes the form $F = \sum_{j=1}^{N} \alpha_j \mathbb{1}_{A_j^n}$.
- $N \leq 2^{dk_n}$ and the A_j^n form a regular grid over $[0, 1]^d$.
- Also, $v_n \geq 2^{-dk_n}$.

Example

- Any $F \in \text{lin}(\mathscr{F}_n)$ takes the form $F = \sum_{j=1}^{N} \alpha_j \mathbb{1}_{A_j^n}$.
- $N \leq 2^{dk_n}$ and the A_j^n form a regular grid over $[0,1]^d$.
- Also, $v_n \geq 2^{-dk_n}$.
- With $\phi(x,y) = (y-x)^2$, the conditions of the theorem read

$$k_n \to \infty, \quad \frac{k_n 2^{dk_n}}{n} \to 0, \quad \text{and} \quad \frac{2^{dk_n}}{\sqrt{n}} \to 0.$$

± 1-classification

- Each F defines a classifier g_F in a natural way:

$$g_F(x) = \begin{cases} +1 & \text{if } F(x) > 0 \\ -1 & \text{otherwise.} \end{cases}$$

± 1-classification

- Each F defines a classifier g_F in a natural way:

$$g_F(x) = \begin{cases} +1 & \text{if } F(x) > 0 \\ -1 & \text{otherwise.} \end{cases}$$

- Proximity between $L(g_F) = \mathbb{P}(g_F(X) \neq Y)$ and the Bayes risk $L^\star$.

± 1-classification

- Each F defines a classifier g_F in a natural way:

$$g_F(x) = \begin{cases} +1 & \text{if } F(x) > 0 \\ -1 & \text{otherwise.} \end{cases}$$

- Proximity between $L(g_F) = \mathbb{P}(g_F(X) \neq Y)$ and the Bayes risk $L^\star$.
- Most often: $L(g_F) - L^\star$ is small as long as $A(F) - A(F^\star)$ is.

± 1-classification

- Each F defines a classifier g_F in a natural way:

$$g_F(x) = \begin{cases} +1 & \text{if } F(x) > 0 \\ -1 & \text{otherwise.} \end{cases}$$

- Proximity between $L(g_F) = \mathbb{P}(g_F(X) \neq Y)$ and the Bayes risk $L^\star$.
- Most often: $L(g_F) - L^\star$ is small as long as $A(F) - A(F^\star)$ is.
- References: Zhang (2004), Bartlett et al. (2006).

± 1-classification

- Each F defines a classifier g_F in a natural way:

$$g_F(x) = \begin{cases} +1 & \text{if } F(x) > 0 \\ -1 & \text{otherwise.} \end{cases}$$

- Proximity between $L(g_F) = \mathbb{P}(g_F(X) \neq Y)$ and the Bayes risk $L^\star$.
- Most often: $L(g_F) - L^\star$ is small as long as $A(F) - A(F^\star)$ is.
- References: Zhang (2004), Bartlett et al. (2006).
- For such well-behaved losses,

$$\lim_{n\to\infty} \mathbb{E} L(g_{\bar{F}_n}) = L^\star.$$

Boosting gradient boosting

Accelerated gradient boosting

- Gradient boosting is a first-order optimization procedure.

Accelerated gradient boosting

- Gradient boosting is a first-order optimization procedure.
- Large-scale machine learning has promoted accelerated first-order schemes.

Accelerated gradient boosting

- Gradient boosting is a first-order optimization procedure.
- Large-scale machine learning has promoted accelerated first-order schemes.
- Nesterov's accelerated gradient descent (1983): $x_0 = y_0$, and

$$\begin{aligned} x_{t+1} &= y_t - w\nabla f(y_t) \\ y_{t+1} &= (1-\gamma_t)x_{t+1} + \gamma_t x_t, \end{aligned}$$

where

$$\lambda_0 = 0, \quad \lambda_t = \frac{1+\sqrt{1+4\lambda_{t-1}^2}}{2}, \quad \text{and} \quad \gamma_t = \frac{1-\lambda_t}{\lambda_{t+1}}.$$

Accelerated gradient boosting

- Gradient boosting is a first-order optimization procedure.
- Large-scale machine learning has promoted accelerated first-order schemes.
- Nesterov's accelerated gradient descent (1983): $x_0 = y_0$, and

$$\begin{aligned} x_{t+1} &= y_t - w\nabla f(y_t) \\ y_{t+1} &= (1-\gamma_t)x_{t+1} + \gamma_t x_t, \end{aligned}$$

where

$$\lambda_0 = 0, \quad \lambda_t = \frac{1+\sqrt{1+4\lambda_{t-1}^2}}{2}, \quad \text{and} \quad \gamma_t = \frac{1-\lambda_t}{\lambda_{t+1}}.$$

- An optimal method for smooth convex optimization: rate $\mathrm{O}(1/t^2)$.

Accelerated gradient boosting

- Gradient boosting is a first-order optimization procedure.
- Large-scale machine learning has promoted accelerated first-order schemes.
- Nesterov's accelerated gradient descent (1983): $x_0 = y_0$, and

$$\begin{array}{rcl} x_{t+1} & = & y_t - w\nabla f(y_t) \\ y_{t+1} & = & (1-\gamma_t)x_{t+1} + \gamma_t x_t, \end{array}$$

  where

$$\lambda_0 = 0, \quad \lambda_t = \frac{1+\sqrt{1+4\lambda_{t-1}^2}}{2}, \quad \text{and} \quad \gamma_t = \frac{1-\lambda_t}{\lambda_{t+1}}.$$

- An optimal method for smooth convex optimization: rate $\mathrm{O}(1/t^2)$.
- Applications in sparse linear regression, compressed sensing, distributed gradient descent, deep and recurrent neural networks, etc.

Accelerated gradient boosting

- Gradient boosting is a first-order optimization procedure.
- Large-scale machine learning has promoted accelerated first-order schemes.
- Nesterov's accelerated gradient descent (1983): $x_0 = y_0$, and

$$\begin{aligned} x_{t+1} &= y_t - w\nabla f(y_t) \\ y_{t+1} &= (1-\gamma_t)x_{t+1} + \gamma_t x_t, \end{aligned}$$

where

$$\lambda_0 = 0, \quad \lambda_t = \frac{1+\sqrt{1+4\lambda_{t-1}^2}}{2}, \quad \text{and} \quad \gamma_t = \frac{1-\lambda_t}{\lambda_{t+1}}.$$

- An optimal method for smooth convex optimization: rate $\mathrm{O}(1/t^2)$.
- Applications in sparse linear regression, compressed sensing, distributed gradient descent, deep and recurrent neural networks, etc.

☞ Idea: combine gradient tree boosting and Nesterov's mechanism.

AGB algorithm

for $t = 0$ to $(T-1)$ **do**

For $i = 1, \ldots, n$, **compute** the negative gradient instances

$$Z_{i,t+1} = -\nabla C_n(G_t)(X_i).$$

Fit a regression tree to the pairs $(X_i, Z_{i,t+1})$, giving terminal nodes $R_{j,t+1}$, $1 \leq j \leq k$.

For $j = 1, \ldots, k$, **compute**

$$w_{j,t+1} \in \arg\min_{w>0} \sum_{X_i \in R_{j,t+1}} \psi(G_t(X_i) + w, Y_i).$$

Update

(a) $F_{t+1} = G_t + \nu \sum_{j=1}^{k} w_{j,t+1} \mathbb{1}_{R_{j,t+1}}$.

(b) $G_{t+1} = (1-\gamma_t)F_{t+1} + \gamma_t F_t$.

end for

Output F_T.

`AGB` algorithm

for $t = 0$ to $(T-1)$ **do**

For $i = 1, \ldots, n$, **compute** the negative gradient instances

$$Z_{i,t+1} = -\nabla C_n(G_t)(X_i).$$

Fit a regression tree to the pairs $(X_i, Z_{i,t+1})$, giving terminal nodes $R_{j,t+1}$, $1 \le j \le k$.

For $j = 1, \ldots, k$, **compute**

$$w_{j,t+1} \in \arg\min_{w>0} \sum_{X_i \in R_{j,t+1}} \psi(G_t(X_i) + w, Y_i).$$

Update

(a) $\quad F_{t+1} = G_t + \nu \sum_{j=1}^{k} w_{j,t+1} \mathbb{1}_{R_{j,t+1}}.$

(b) $\quad G_{t+1} = (1-\gamma_t) F_{t+1} + \gamma_t F_t.$

end for

Output F_T.

AGB algorithm

1: **for** $t = 0$ to $(T-1)$ **do**

2: For $i = 1, \ldots, n$, **compute** the negative gradient instances

$$Z_{i,t+1} = -\nabla C_n(G_t)(X_i).$$

3: **Fit** a regression tree to the pairs $(X_i, Z_{i,t+1})$, giving terminal nodes $R_{j,t+1}$, $1 \leq j \leq k$.

4: For $j = 1, \ldots, k$, **compute**

$$w_{j,t+1} \in \arg\min_{w>0} \sum_{X_i \in R_{j,t+1}} \psi(G_t(X_i) + w, Y_i).$$

5: **Update**

(a) $F_{t+1} = G_t + \nu \sum_{j=1}^{k} w_{j,t+1} \mathbb{1}_{R_{j,t+1}}$.

(b) $G_{t+1} = (1-\gamma_t)F_{t+1} + \gamma_t F_t$.

6: **end for**

7: **Output** F_T.

`AGB` algorithm

for $t = 0$ to $(T-1)$ **do**

For $i = 1, \ldots, n$, **compute** the negative gradient instances

$$Z_{i,t+1} = -\nabla C_n(G_t)(X_i).$$

Fit a regression tree to the pairs $(X_i, Z_{i,t+1})$, giving terminal nodes $R_{j,t+1}$, $1 \leq j \leq k$.

For $j = 1, \ldots, k$, **compute**

$$w_{j,t+1} \in \arg\min_{w>0} \sum_{X_i \in R_{j,t+1}} \psi(G_t(X_i) + w, Y_i).$$

Update

(a) $F_{t+1} = G_t + \nu \sum_{j=1}^{k} w_{j,t+1} \mathbb{1}_{R_{j,t+1}}$.

(b) $G_{t+1} = (1-\gamma_t)F_{t+1} + \gamma_t F_t$.

end for

Output F_T.

`AGB` algorithm

for $t = 0$ to $(T-1)$ **do**

For $i = 1, \dots, n$, **compute** the negative gradient instances

$$Z_{i,t+1} = -\nabla C_n(G_t)(X_i).$$

Fit a regression tree to the pairs $(X_i, Z_{i,t+1})$, giving terminal nodes $R_{j,t+1}$, $1 \le j \le k$.

For $j = 1, \dots, k$, **compute**

$$w_{j,t+1} \in \arg\min_{w>0} \sum_{X_i \in R_{j,t+1}} \psi(G_t(X_i) + w, Y_i).$$

Update

(a) $F_{t+1} = G_t + \nu \sum_{j=1}^{k} w_{j,t+1} \mathbb{1}_{R_{j,t+1}}$.

(b) $G_{t+1} = (1-\gamma_t)F_{t+1} + \gamma_t F_t$.

end for

Output F_T.

`AGB` algorithm

for $t = 0$ to $(T-1)$ **do**

For $i = 1, \dots, n$, **compute** the negative gradient instances

$$Z_{i,t+1} = -\nabla C_n(G_t)(X_i).$$

Fit a regression tree to the pairs $(X_i, Z_{i,t+1})$, giving terminal nodes $R_{j,t+1}$, $1 \leq j \leq k$.

For $j = 1, \dots, k$, **compute**

$$w_{j,t+1} \in \arg\min_{w>0} \sum_{X_i \in R_{j,t+1}} \psi(G_t(X_i) + w, Y_i).$$

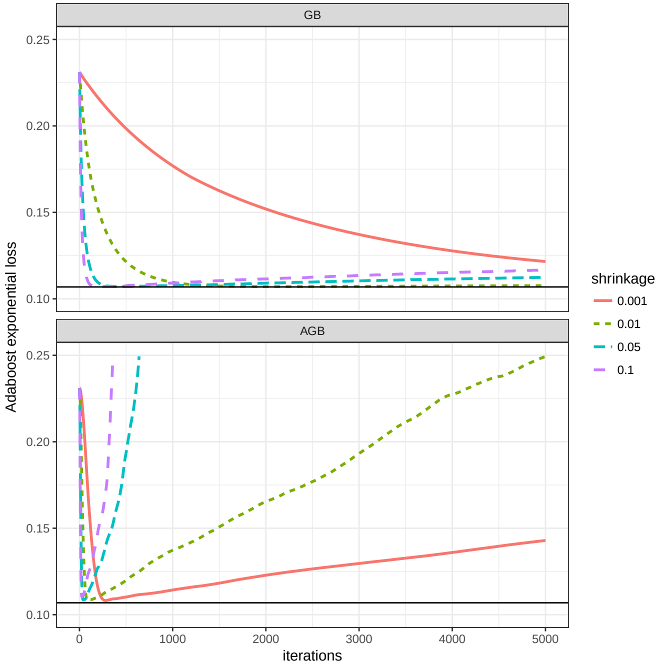
Update

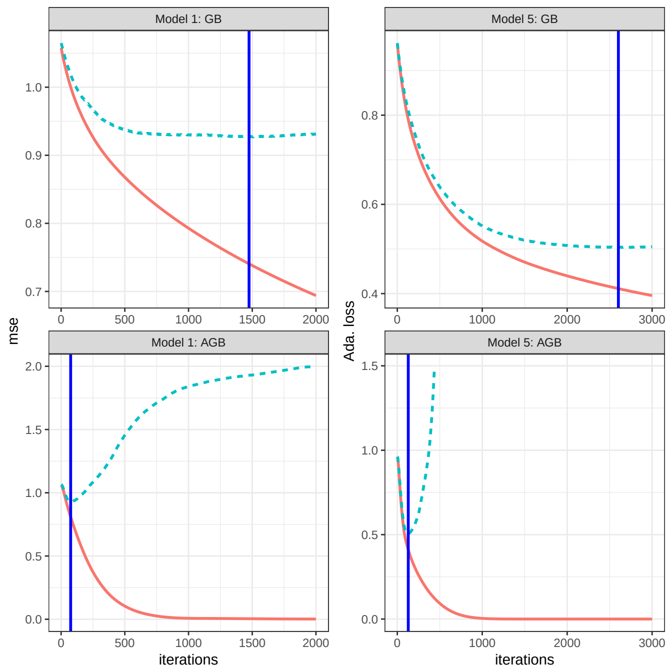
(*a*) $F_{t+1} = G_t + \nu \sum_{j=1}^{k} w_{j,t+1} \mathbb{1}_{R_{j,t+1}}$.

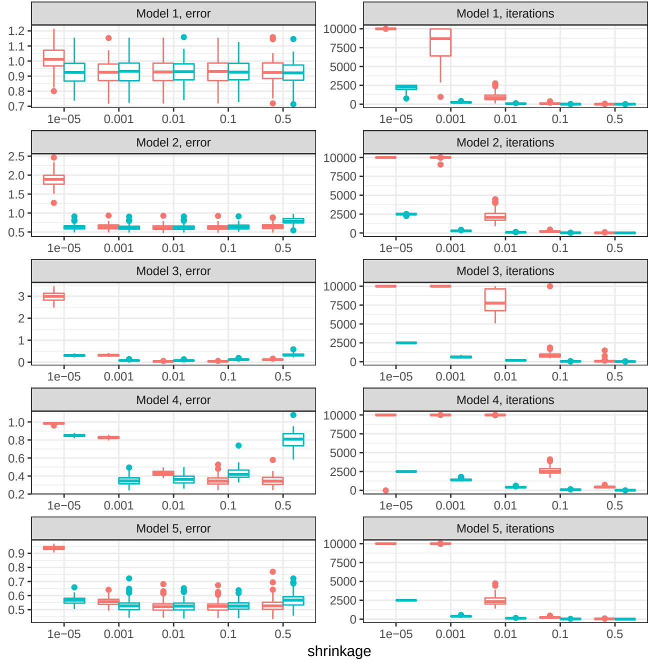
(*b*) $G_{t+1} = (1-\gamma_t) F_{t+1} + \gamma_t F_t$.

end for

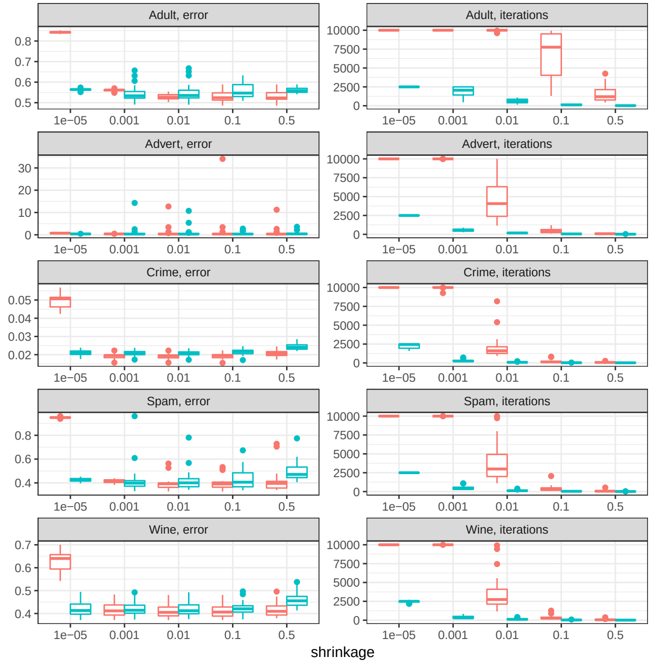
Output F_T.


GB
AGB
Adaboost exponential loss
0.25
0.20
0.15
0.10
0
1000
2000
3000
4000
5000
iterations
shrinkage
0.001
0.01
0.05
0.1


Model 1: GB
Model 5: GB
Model 1: AGB
Model 5: AGB
mse
Ada. loss
iterations
iterations
0
500
1000
1500
2000
3000
0.7
0.8
0.9
1.0
0.4
0.6
0.0
0.5
1.5
2.0



Model 1, error
Model 1, iterations
Model 2, error
Model 2, iterations
Model 3, error
Model 3, iterations
Model 4, error
Model 4, iterations
Model 5, error
Model 5, iterations
1e−05
0.001
0.01
0.1
0.5
shrinkage



Adult, error
Adult, iterations
Advert, error
Advert, iterations
Crime, error
Crime, iterations
Spam, error
Spam, iterations
Wine, error
Wine, iterations
1e−05
0.001
0.01
0.1
0.5
shrinkage

Take-home message

- AGB retains the excellent performance of gradient boosting.
- It is less sensitive to the shrinkage parameter.
- It outputs sparse predictors.
- A decisive advantage in large-scale learning.
- More at github.com/lrouviere/AGB.