Ensemble Methods

Today

Decision tree summary
Random forest review
Boosting

- Adaboost
- Gradient boosting

Decision Trees: Summary

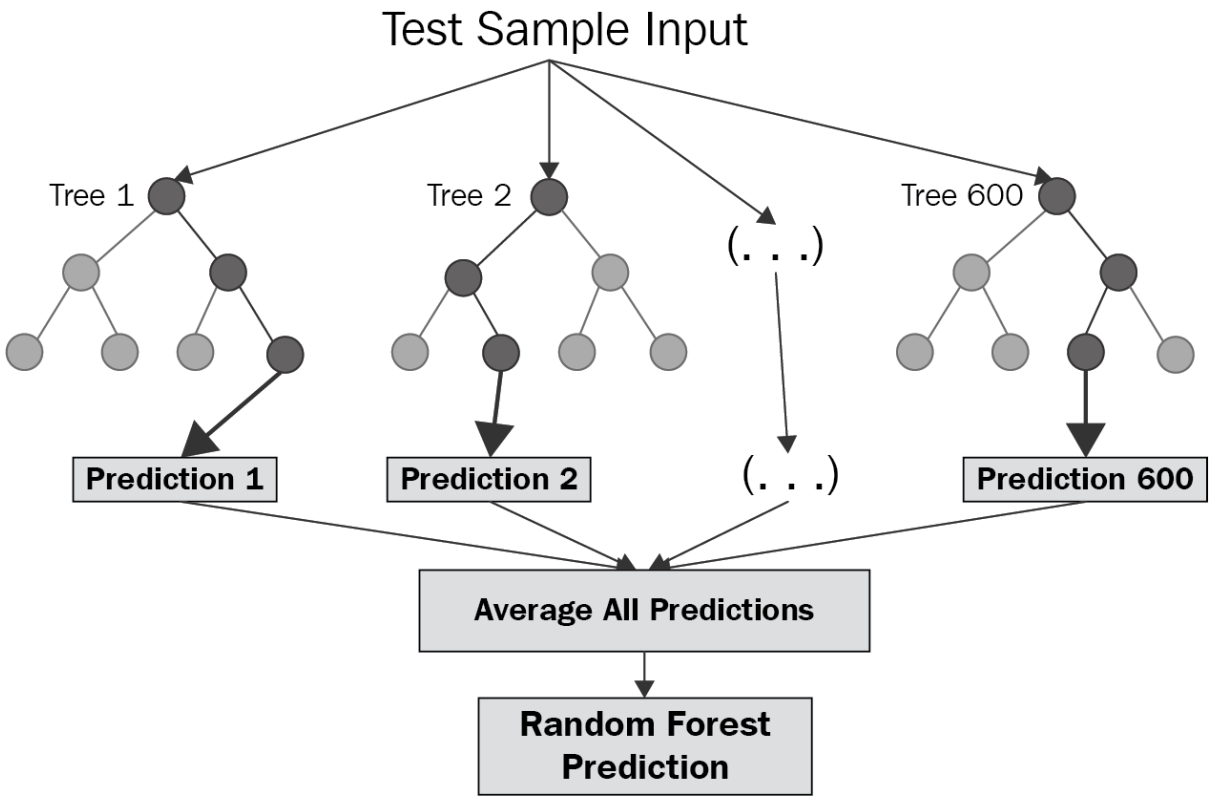
Strengths:

- Interpretable, expressive
- Fast (learning + prediction)
- Categorical features

Weaknesses:

- Training is suboptimal
- Sensitivity to perturbations of training data

Random Forests



Random forests achieve state-of-the-art prediction for many datasets. They overcome the stability issue, at the expense of interpretability.

Adaboost

- Adaboost is an ensemble method for classification in which
  - The final vote/average is weighted
  - The elements of the ensemble are determined sequentially
- We'll focus on binary classification (labels -1 and 1)
- Final classifier has the form

$$h_T(x) = \text{sign}\left\{ \sum_{t=1}^{T} \alpha_t f_t(x) \right\} \qquad f_t(x) \in \{-1, 1\}$$

- $f_1, \ldots, f_T$ are called *base classifiers* (not necessarily trees)
- $\alpha_1, \ldots, \alpha_T > 0$ reflect the confidence in the various base classifiers

Base Learners

- Let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ be the training data
- Let $\mathcal{F}$ be a fixed set of classifiers called the *base class*
- A *base learner* for $\mathcal{F}$ is an algorithm that takes as input a set of weights $\boldsymbol{w} = (w_1, \ldots, w_n)$ such that $w_i \geq 0$, $\sum w_i = 1$, and outputs a classifier $f \in \mathcal{F}$ such that the weighted empirical risk

$$\sum_{i=1}^{n} w_i 1_{\{f(x_i) \neq y_i\}}$$

  is (approximately) minimized.
- **Example:** $\mathcal{F}$ = set of all decision trees of depth J, for some fixed J.
- How should we implement the base learner for decision trees of a certain depth?

Resample training data $\sim (w_1, \ldots, w_n)$, then run a conventional DT learning algorithm

The Boosting Principle

- The basic idea behind boosting is to learn $f_1, \ldots, f_T$ sequentially, where f_t is produced by the base learner given a weight vector $\boldsymbol{w}^t = (w_1^t, \ldots, w_n^t)$ as input.
- The weights are updated to place more emphasis on training examples that are harder to classify
- Conceptually, we want
  - If $f_t(\boldsymbol{x}_i) = y_i$, then $w_i^{t+1} < w_i^t$
  - If $f_t(\boldsymbol{x}_i) \neq y_i$, then $w_i^{t+1} > w_i^t$

Adaboost

- Short for "adaptive boosting"

Input: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n, T, \mathcal{F}$, base learner
Initialize: $\boldsymbol{w}^1 = (\frac{1}{n}, \ldots, \frac{1}{n})$
For $t = 1, \ldots, T$

$$e_w(f) = \sum w_i 1_{\{f(x_i) \neq y_i\}}$$

$$\boldsymbol{w}^t \rightarrow \boxed{\text{base learner}} \rightarrow f_t$$

$$r_t = \sum_{i=1}^n w_i^t \mathbf{1}_{\{f_t(\boldsymbol{x}_i) \neq y_i\}} = e_{\boldsymbol{w}^t}(f_t)$$

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - r_t}{r_t}\right)$$

$$w_i^{t+1} = \frac{w_i^t \exp(-\alpha_t y_i f_t(x_i))}{\sum_{j=1}^n w_j^t \exp(-\alpha_t y_j f_t(x_j))}$$

End
Output: $h_T(\boldsymbol{x}) = \text{sign}\left\{\sum_{t=1}^T \alpha_t f_t(\boldsymbol{x})\right\}$

Demo

Poll

Recall that

$$r_t = \sum_{i=1}^{n} w_i^t \mathbf{1}_{\{f_t(\boldsymbol{x}_i) \neq y_i\}} = e_{\boldsymbol{w}}(f_t)$$

$\boldsymbol{w} = w^t$

Let $-f_t$ be the classifier that always makes the opposite prediction from f_t, i.e., $(-f_t)(\boldsymbol{x}) = -(f_t(\boldsymbol{x}))$.

What is $e_{\boldsymbol{w}}(-f_t)$? ($\boldsymbol{w} = w^t$)

(A) $-r_t$

(B) $1 - r_t$

(C) $2 - r_t$

(D) $2r_t$

$$1 = \sum_{i=1}^{n} w_i^t$$

$$= \sum_{i=1}^{n} w_i^t \mathbf{1}_{\{f_t(x_i) \neq y_i\}} + \sum_{i=1}^{n} w_i^t \mathbf{1}_{\{f_t(x_i) = y_i\}}$$

$$= r_t + (1 - r_t) = e_{w^t}(-f_t)$$

Exponential Convergence

- Denote $\gamma_t = \frac{1}{2} - r_t$. Note that we may assume $\gamma_t \geq 0 \Leftrightarrow r_t \leq \frac{1}{2}$. If not, just replace f_t with $-f_t$ and note that for any f and $\boldsymbol{w}$,

$$e_w(f) + e_w(-f) = 1$$

- **Theorem:** The training error of Adaboost satisfies

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\{h_T(\boldsymbol{x}_i) \neq y_i\}} \leq \exp\left(-2\sum_{t=1}^{T} \gamma_t^2\right)$$

  In particular, if $\gamma_t \geq \gamma > 0$ for all t, then

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\{h_T(\boldsymbol{x}_i) \neq y_i\}} \leq \exp\left(-2T\gamma^2\right)$$

$r_t \leq \frac{1}{2} - \gamma$

- We may interpret $r_t = \frac{1}{2}$ as corresponding to a base classifier f_t that randomly guesses. Thus $\gamma_t \geq \gamma > 0$ means f_t is at least slightly better than random guessing (the so-called *weak learning hypothesis*)

Poll

True or False: It is usually unwise to continue iterating Adaboost after the training error reaches zero. *Hint: Disregard your intuition based on this course so far.*

(A) True

(B) False

"Boosting the Margin"

Summary of Adaboost

- Adaboost was the backbone of the first real-time, high accuracy face detection system (Viola-Jones)
- Multiclass extension of Adaboost is possible but nontrivial
- See the book "Boosting" by Schapire and Freund

Boosting as ERM

- It turns out that Adaboost can be viewed as an iterative algorithm for minimizing the empirical risk corresponding to the exponential loss. By generalizing the loss, we obtain different boosting algorithms with different properties.

- For a fixed base class $\mathcal{F}$, define

$$\mathcal{F}_T = \left\{ \sum_{t=1}^{T} \alpha_t f_t \;\middle|\; f_1, \ldots, f_T \in \mathcal{F},\ \alpha_1, \ldots, \alpha_T > 0 \right\}$$

- Now consider the problem

$$\min_{F \in \mathcal{F}_T} \frac{1}{n} \sum_{i=1}^{n} 1_{\{y_i F(x_i) < 0\}}$$

$\phi(s) = \log(1 + e^{-s})$

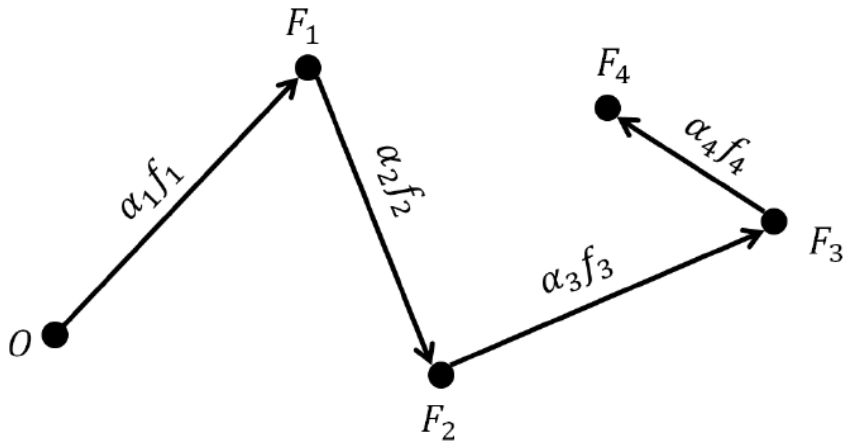
$\phi(s) = \max(0, 1-s)$

$\phi(s) = e^{-s}$

- Let L be a surrogate loss. Assume $L(y,t) = \phi(yt)$ for some ϕ. A more tractable problem is

$$\min_{F \in \mathcal{F}_T} \frac{1}{n} \sum_{i=1}^{n} \phi(y_i F(x_i))$$

Gradient Boosting 1: Functional Gradient Descent



- Denote $F_t = \alpha_1 f_1 + \cdots + \alpha_t f_t$
- Write $F_t = F_{t-1} + \alpha_t f_t$
- The idea is to optimize over $F \in \text{span}(\mathcal{F})$ sequentially
- View $\alpha_1, f_1, \ldots, \alpha_{t-1}, f_{t-1}$ as fixed, and set

$$B_t(\alpha, f) = \frac{1}{n}\sum_{i=1}^{n} \phi\left(y_i F_{t-1}(x_i) + y_i \cdot \alpha f(x_i)\right)$$

- We can choose α_t, f_t by

1. f_t = function $f \in \mathcal{F}$ for which the directional derivative of B_t in the direction f is minimized (given by weak learner with certain weights depending on the loss and F_{t-1})
2. α_t = stepsize $\alpha > 0$ in the direction f_t for which $B_t(\alpha, f_t)$ is minimized

Assume ϕ is differentiable and $\phi' < 0$ everywhere

$$\left.\frac{\partial B_t(\alpha, f)}{\partial \alpha}\right|_{\alpha=0} = \frac{1}{n}\sum_{i=1}^{n} y_i f(x_i)\, \phi'\Big(y_i F_{t-1}(x_i) + \underbrace{y_i \alpha f(x_i)}_{=0}\Big)$$

$$\alpha - \sum_{i=1}^{n} y_i f(x_i) \cdot \underbrace{\frac{\phi'(y_i F_{t-1}(x_i))}{\sum_{j=1}^{n} \phi'(y_j F_{t-1}(x_i))}}_{w_i^t}$$

$$= \sum_{i=1}^{n} w_i^t 1_{\{f(x_i) \neq y_i\}} - \sum w_i^t 1_{\{f(x_i) = y_i\}}$$

$$= 2\Big(\underbrace{\sum w_i^t 1_{\{f(x_i) \neq y_i\}}}_{\text{minimize w/ base learner}}\Big) - 1$$

Functional Gradient Descent

2)

$$\alpha_t = \arg\min_{\alpha} \frac{1}{n} \sum_i \phi\left(y_i f_{t-1}(x_i) + y_i \alpha f_t(x_i)\right)$$

This is a one-dimensional optimization problem, so even if no closed form solution exists, it can be solved very quickly using an iterative solver such as Newton's method.

Generalization of Adaboost

- Let ϕ be a surrogate loss such that $\phi' < 0$ everywhere

Input: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n, T, \phi, \mathcal{F}$, base learner
Initialize: $\boldsymbol{w}^1 = (\frac{1}{n}, \ldots, \frac{1}{n})$, $F_0 = 0$
For $t = 1, \ldots, T$

$$\boldsymbol{w}^t \rightarrow \boxed{\text{base learner}} \rightarrow f_t$$

$$\alpha_t = \arg\min_{\alpha} \frac{1}{n} \sum_{i=1}^{n} \phi(y_i F_{t-1}(\boldsymbol{x}_i) + y_i \alpha f_t(\boldsymbol{x}_i))$$

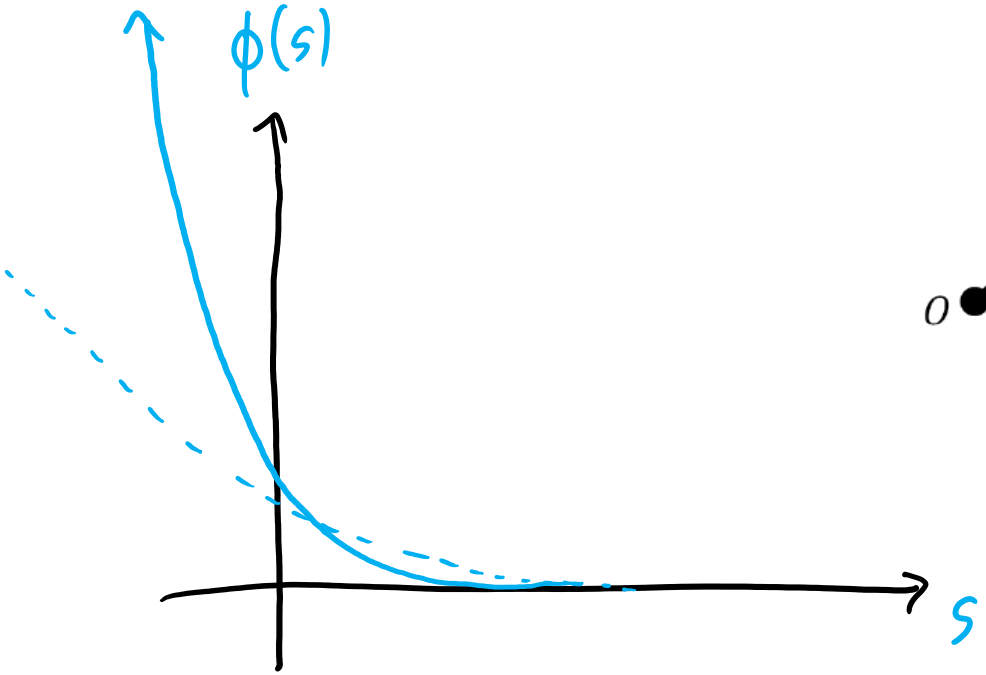
$$F_t = F_{t-1} + \alpha_t f_t$$

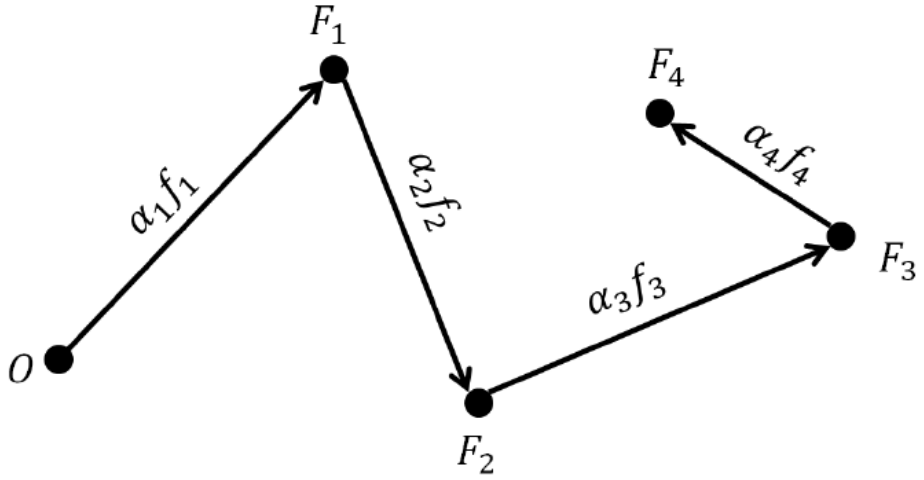
$$w_i^{t+1} = \frac{\phi(y_i F_t(\boldsymbol{x}_i))}{\sum_j \phi(y_j F_t(\boldsymbol{x}_j))}$$

End
Output: $h_T(\boldsymbol{x}) = \text{sign}\left\{ \sum_{t=1}^{T} \alpha_t f_t(\boldsymbol{x}) \right\}$

Functional Gradient Descent

- If $L(y,t) = \exp(-yt)$, the previous algorithm reduces to Adaboost.
- Exponential loss leads to nice formulas for the iterative updates, but other losses are still tractable and are less sensitive to outliers.





Gradient Boosting: Regression

- Fix a base class $\mathcal{F}$ of regression trees, e.g., all regression trees with depth ≤ 6.
- For a fixed $\mathcal{F}$, define

$$\mathcal{F}_T^+ = \left\{ \sum_{t=1}^{T} f_t \,\middle|\, T \geq 1, f_t \in \mathcal{F} \right\}$$

- Now consider the problem

$$\min_{F \in \mathcal{F}_T^+} \frac{1}{n} \sum_{i=1}^{n} (y_i - F(\boldsymbol{x}_i))^2$$

- Global optimization is intractable because of complexity of search space
- Idea: Learn F sequentially (i.e., greedily)

view as response $\tilde{y}_i = y_i - F_{t-1}(x_i)$

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - F_{t-1}(x_i) - f(x) \right)^2$$

Gradient Boosting 2: Classification

- Fix a base class $\mathcal{F}$ of regression trees, e.g., all regression trees with depth ≤ 6.
- For a fixed $\mathcal{F}$, define

$$\mathcal{F}_+ = \left\{ \sum_{t=1}^{T} f_t \,\middle|\, T \geq 1, f_t \in \mathcal{F} \right\}$$

- Now consider the problem

$$\min_{F \in \mathcal{F}_+} \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{\text{sign}(F(\boldsymbol{x}_i)) \neq y_i\}}$$

- Let L be a surrogate loss. A more tractable problem is

$$\min_{F \in \mathcal{F}_+} \frac{1}{n} \sum_{i=1}^{n} L(y_i, F(\boldsymbol{x}_i))$$

- Idea: Learn F sequentially

Gradient Boosting 2: General Strategy

- Denote $F_t = f_1 + \cdots + f_t$
- Write $F_t = F_{t-1} + f_t$
- Want to solve

$$\min_{F \in \mathcal{F}_t} \frac{1}{n} \sum_{i=1}^{n} L(y_i, F(\boldsymbol{x}_i))$$

- The idea is to optimize over $F \in \mathcal{F}_t$ sequentially
- View $f_1, \ldots, f_{t-1}$ as fixed, and solve

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(y_i, F_{t-1}(\boldsymbol{x}_i) + f(\boldsymbol{x}_i))$$

Second Order Approximation

- Denote

$$g_i(s) = \frac{\partial L(y_i, s)}{\partial s}$$

and

$$h_i(s) = \frac{\partial^2 L(y_i, s)}{\partial s^2}$$

- By Taylor's theorem:

$$L(y_i, F_{t-1}(\boldsymbol{x}_i) + f(\boldsymbol{x}_i)) \approx$$
$$L(y, F_{t-1}(\boldsymbol{x}_i)) + g_i(F_{t-1}(\boldsymbol{x}_i))f(\boldsymbol{x}_i) + \frac{1}{2}h_i(F_{t-1}(\boldsymbol{x}_i))f^2(\boldsymbol{x}_i)$$

- The approximate empirical risk can be minimized efficiently used tree-like growing algorithms

Summary of Gradient Boosting

- There are multiple approaches to gradient boosting. We looked at the two major ones.
- Highly optimized implementation of gradient boosting based on second approach: XGBoost.
- T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," KDD 2016.
- XGBoost has been used in an impressive number of winning Kaggle submissions
- Other highly optimized implementations: CatBoost, LightGBM and Scikit-learn.

GBM