Recitation 9

Gradient Boosting

Brett Bernstein

CDS at NYU

March 29, 2017

Intro Question

Question

Suppose 10 different meteorologists have produced functions $f_1, \ldots, f_{10} : \mathbb{R}^d \to \mathbb{R}$ that forecast tomorrow's noon-time temperature using the same d features. Given a validation set containing 1000 data points $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$ of similar forecast situations, describe a method to forecast tomorrow's noon-time temperature. Would you use boosting, bagging or neither?

Intro Solution

Solution

Let $\hat{x}_i = (x_i, f_1(x_i), \ldots, f_{10}(x_i)) \in \mathbb{R}^{d+10}$. Then use any fitting method you like to produce an aggregate decision function $f : \mathbb{R}^{d+10} \to \mathbb{R}$. This method is sometimes called stacking.

1. This isn't bagging - we didn't generate bootstrap samples and learn a decision function on each of them.
2. This isn't boosting - boosting learns decision functions on varying datasets to produce an aggregate classifier.

Different Ensembles

1. Parallel ensemble: each base model is fit independently of the other models. Examples are bagging and stacking.
2. Sequential ensemble: each base model is fit in stages depending on the previous fits. Examples are AdaBoost and Gradient Boosting.

AdaBoost Review

1. Recall that a learner, or learning algorithm take a dataset as input and produces a decision function in some hypothesis space.

Question

Suppose we had a learner that given a dataset, and a weighting (importance) scheme on that dataset, would produce a classifier h that has lower than .5 loss using the weighted $0-1$ loss:

$$\frac{1}{n}\sum_{i=1}^{n} w_i\,\mathbf{1}(y_i \neq h(x_i)) \leq .5.$$

Can we use this learner to create an ensemble that makes accurate predictions?

AdaBoost Review

1. Recall that a learner, or learning algorithm take a dataset as input and produces a decision function in some hypothesis space.

Question

Suppose we had a learner that given a dataset, and a weighting (importance) scheme on that dataset, would produce a classifier h that has lower than .5 loss using the weighted $0-1$ loss:

$$\frac{1}{n}\sum_{i=1}^{n} w_i \mathbf{1}(y_i \neq h(x_i)) \leq .5.$$

Can we use this learner to create an ensemble that makes accurate predictions?

- We saw that AdaBoost solves this problem.
- Can get around weighted loss functions using sampling trick.

Additive Models

1. *Additive models* over a *base hypothesis space* $\mathcal{H}$ take the form

$$\mathcal{F} = \left\{ f(x) = \sum_{m=1}^{M} \nu_m h_m(x) \mid h_m \in \mathcal{H}, \nu_m \in \mathbb{R} \right\}.$$

2. Since we are taking linear combinations, we assume the h_m functions take values in $\mathbb{R}$ or some other vector space.
3. Empirical risk minimization over $\mathcal{F}$ tries to find

$$\operatorname*{arg\,min}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)).$$

4. This in general is a difficult task, as the number of base hypotheses M is unknown, and each base hypothesis h_m ranges over all of $\mathcal{H}$.

Forward Stagewise Additive Modeling (FSAM)

The FSAM method fits additive models using the following (greedy) algorithmic structure:

1. Initialize $f_0 \equiv 0$.
2. For stage $m = 1, \ldots, M$:
   1. Choose $h_m \in \mathcal{H}$ and $\nu_m \in \mathbb{R}$ so that
      $$f_m = f_{m-1} + \nu_m h_m$$
      has the minimum empirical risk.
   2. The function f_m has the form
      $$f_m = \nu_1 h_1 + \cdots + \nu_m h_m.$$

Forward Stagewise Additive Modeling (FSAM)

The FSAM method fits additive models using the following (greedy) algorithmic structure:

1. Initialize $f_0 \equiv 0$.
2. For stage $m = 1, \ldots, M$:
    1. Choose $h_m \in \mathcal{H}$ and $\nu_m \in \mathbb{R}$ so that
    $$f_m = f_{m-1} + \nu_m h_m$$
    has the minimum empirical risk.
    2. The function f_m has the form
    $$f_m = \nu_1 h_1 + \cdots + \nu_m h_m.$$

- When choosing h_m, ν_m during stage m, we must solve the minimization

$$(\nu_m, h_m) = \operatorname*{arg\,min}_{\nu \in \mathbb{R}, h \in \mathcal{H}} \sum_{i=1}^{n} \ell(y_i, f_{m-1}(x_i) + \nu h(x_i)).$$

Gradient Boosting

1. Can we simplify the following minimization problem:

$$(\nu_m, h_m) = \underset{\nu \in \mathbb{R}, h \in \mathcal{H}}{\arg\min} \sum_{i=1}^{n} \ell(y_i, f_{m-1}(x_i) + \nu h(x_i)).$$

2. What if we linearize the problem and take a step along the steepest descent direction?
3. Good idea, but how do we handle the constraint that h is a function that lies in $\mathcal{H}$, the base hypothesis space?

Gradient Boosting

1. Can we simplify the following minimization problem:

$$(\nu_m, h_m) = \underset{\nu\in\mathbb{R}, h\in\mathcal{H}}{\arg\min} \sum_{i=1}^{n} \ell(y_i, f_{m-1}(x_i) + \nu h(x_i)).$$

2. What if we linearize the problem and take a step along the steepest descent direction?
3. Good idea, but how do we handle the constraint that h is a function that lies in $\mathcal{H}$, the base hypothesis space?
4. First idea: since we are doing empirical risk minimization, we only care about the values h takes on the training set. Thus we can think of h as a vector $(h(x_1), \ldots, h(x_n))$.
5. Second idea: first compute unconstrained steepest descent direction, and then constrain (project) onto possible choices from $\mathcal{H}$.

Gradient Boosting Machine

1. Initialize $f_0 \equiv 0$.
2. For stage $m = 1, \ldots, M$:
    1. Compute the steepest descent direction (also called *pseudoresiduals*):
    $$r_m = -\left(\partial_2 \ell(y_1, f_{m-1}(x_1)), \ldots, \partial_2 \ell(y_n, f_{m-1}(x_n))\right).$$
    2. Find the closest base hypothesis (using Euclidean distance):
    $$h_m = \operatorname*{arg\,min}_{h \in \mathcal{H}} \sum_{i=1}^{n} ((r_m)_i - h(x_i))^2.$$
    3. Choose fixed step size $\nu_m \in (0, 1]$ or line search:
    $$\nu_m = \operatorname*{arg\,min}_{\nu \geq 0} \sum_{i=1}^{n} \ell(y_i, f_{m-1}(x_i) + \nu h_m(x_i)).$$
    4. Take the step:
    $$f_m(x) = f_{m-1}(x) + \nu_m h_m(x).$$

Gradient Boosting Machine

1. Each stage we need to solve the following step:

$$h_m = \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{n} ((r_m)_i - h(x_i))^2.$$

How do we do this?

Gradient Boosting Machine

1. Each stage we need to solve the following step:

$$h_m = \operatorname*{arg\,min}_{h \in \mathcal{H}} \sum_{i=1}^{n} ((r_m)_i - h(x_i))^2.$$

How do we do this?

2. This is a standard least squares regression task on the "mock" dataset

$$\mathcal{D}^{(m)} = \{(x_1, (r_m)_1), \ldots, (x_n, (r_m)_n)\}.$$

3. We assume that we have a learner that (approximately) solves least squares regression over $\mathcal{H}$.

Gradient Boosting Comments

1. The algorithm above is sometimes called AnyBoost or Functional Gradient Descent.
2. The most commonly used base hypothesis space is small regression trees (HTF recommends between 4 and 8 leaves).

Practice With Different Loss Functions

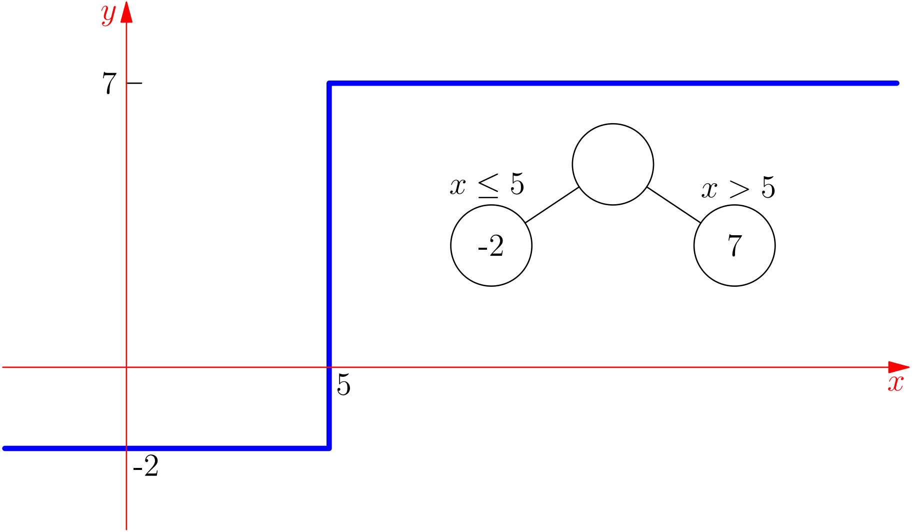
Question

Explain how to perform gradient boosting with the following loss functions:

1. Square loss: $\ell(y, a) = (y - a)^2/2$.
2. Absolute loss: $\ell(y, a) = |y - a|$.
3. Exponential margin loss: $\ell(y, a) = e^{-ya}$.

Demonstration Using Decision Stumps

Below is an example of a decision stump for functions $h : \mathbb{R} \to \mathbb{R}$.



Demonstration Using Decision Stumps

Below is the dataset we will use.