A Gentle Introduction to Gradient Boosting

Cheng Li
chengli@ccs.neu.edu
College of Computer and Information Science
Northeastern University

Gradient Boosting

- a powerful machine learning algorithm
- it can do
  - regression
  - classification
  - ranking
- won Track 1 of the Yahoo Learning to Rank Challenge

Our implementation of Gradient Boosting is available at
`https://github.com/cheng-li/pyramid`

Outline of the Tutorial

1. What is Gradient Boosting
2. A brief history
3. Gradient Boosting for regression
4. Gradient Boosting for classification
5. A demo of Gradient Boosting
6. Relationship between Adaboost and Gradient Boosting
7. Why it works

Note: This tutorial focuses on the intuition. For a formal treatment, see [Friedman, 2001]

What is Gradient Boosting

Gradient Boosting = Gradient Descent + Boosting

Adaboost

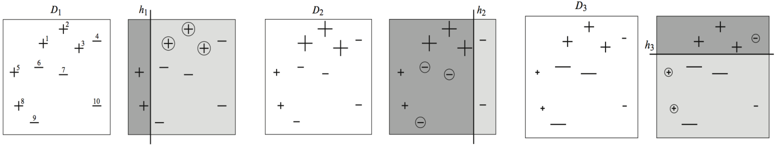


Figure: AdaBoost. Source: Figure 1.1 of [Schapire and Freund, 2012]

What is Gradient Boosting

Gradient Boosting = Gradient Descent + Boosting

Adaboost

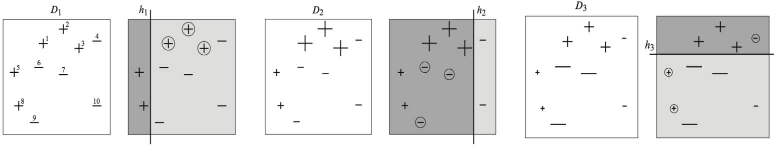


Figure: AdaBoost. Source: Figure 1.1 of [Schapire and Freund, 2012]

- Fit an additive model (ensemble) $\sum_t \rho_t h_t(x)$ in a forward stage-wise manner.
- In each stage, introduce a weak learner to compensate the shortcomings of existing weak learners.
- In Adaboost, "shortcomings" are identified by high-weight data points.

Gradient Boosting = Gradient Descent + Boosting

Adaboost

$$H(x) = \sum_t \rho_t h_t(x)$$

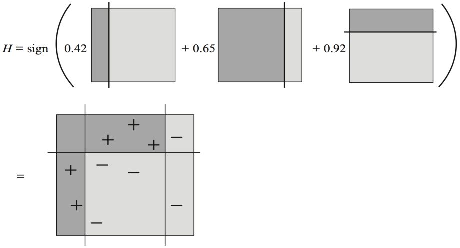


Figure: AdaBoost. Source: Figure 1.2 of [Schapire and Freund, 2012]

What is Gradient Boosting

Gradient Boosting = Gradient Descent + Boosting

Gradient Boosting

- Fit an additive model (ensemble) $\sum_t \rho_t h_t(x)$ in a forward stage-wise manner.
- In each stage, introduce a weak learner to compensate the shortcomings of existing weak learners.
- In Gradient Boosting, "shortcomings" are identified by gradients.
- Recall that, in Adaboost, "shortcomings" are identified by high-weight data points.
- Both high-weight data points and gradients tell us how to improve our model.

What is Gradient Boosting

Why and how did researchers invent Gradient Boosting?

A Brief History of Gradient Boosting

- Invent Adaboost, the first successful boosting algorithm [Freund et al., 1996, Freund and Schapire, 1997]
- Formulate Adaboost as gradient descent with a special loss function[Breiman et al., 1998, Breiman, 1999]
- Generalize Adaboost to Gradient Boosting in order to handle a variety of loss functions
  [Friedman et al., 2000, Friedman, 2001]

Gradient Boosting for Different Problems

Difficulty:

regression $===>$ classification $===>$ ranking

Gradient Boosting for Regression

Let's play a game...

You are given $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, and the task is to fit a model $F(x)$ to minimize square loss.

Suppose your friend wants to help you and gives you a model F.

You check his model and find the model is good but not perfect.

There are some mistakes: $F(x_1) = 0.8$, while $y_1 = 0.9$, and $F(x_2) = 1.4$ while $y_2 = 1.3$... How can you improve this model?

Gradient Boosting for Regression

Let's play a game...

You are given $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, and the task is to fit a model $F(x)$ to minimize square loss.

Suppose your friend wants to help you and gives you a model F.

You check his model and find the model is good but not perfect.

There are some mistakes: $F(x_1) = 0.8$, while $y_1 = 0.9$, and $F(x_2) = 1.4$ while $y_2 = 1.3$... How can you improve this model?

Rule of the game:

- You are not allowed to remove anything from F or change any parameter in F. 不修改F

Gradient Boosting for Regression

Let's play a game...
You are given $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, and the task is to fit a model $F(x)$ to minimize square loss.
Suppose your friend wants to help you and gives you a model F.
You check his model and find the model is good but not perfect.
There are some mistakes: $F(x_1) = 0.8$, while $y_1 = 0.9$, and $F(x_2) = 1.4$ while $y_2 = 1.3$... How can you improve this model?
Rule of the game:

- You are not allowed to remove anything from F or change any parameter in F.
- You can add an additional model (regression tree) h to F, so the new prediction will be $F(x) + h(x)$.

不修改F，但是可以通过添加子模型。也就是不修改效果不完美的模型，通过加上新的模型来改进模型的整体效果。

Gradient Boosting for Regression

Simple solution:

You wish to improve the model such that

$$
\begin{aligned}
&F(x_1) + h(x_1) = y_1 \\
&F(x_2) + h(x_2) = y_2 \\
&... \\
&F(x_n) + h(x_n) = y_n
\end{aligned}
$$

Gradient Boosting for Regression

Simple solution:

Or, equivalently, you wish

$$
\begin{aligned}
h(x_1) &= y_1 - F(x_1) \\
h(x_2) &= y_2 - F(x_2) \\
&\dots \\
h(x_n) &= y_n - F(x_n)
\end{aligned}
$$

Gradient Boosting for Regression

Simple solution:

Or, equivalently, you wish

$$
\begin{aligned}
h(x_1) &= y_1 - F(x_1) \\
h(x_2) &= y_2 - F(x_2) \\
&\ldots \\
h(x_n) &= y_n - F(x_n)
\end{aligned}
$$

Can any regression tree h achieve this goal perfectly?

Gradient Boosting for Regression

Simple solution:

Or, equivalently, you wish

$$
\begin{aligned}
h(x_1) &= y_1 - F(x_1) \\
h(x_2) &= y_2 - F(x_2) \\
&\ldots \\
h(x_n) &= y_n - F(x_n)
\end{aligned}
$$

Can any regression tree h achieve this goal perfectly?
Maybe not....

Gradient Boosting for Regression

Simple solution:

Or, equivalently, you wish

$$
\begin{aligned}
h(x_1) &= y_1 - F(x_1) \\
h(x_2) &= y_2 - F(x_2) \\
&\ldots \\
h(x_n) &= y_n - F(x_n)
\end{aligned}
$$

Can any regression tree h achieve this goal perfectly?
Maybe not....
But some regression tree might be able to do this approximately.

Gradient Boosting for Regression

Simple solution:

Or, equivalently, you wish

$$
\begin{aligned}
h(x_1) &= y_1 - F(x_1) \\
h(x_2) &= y_2 - F(x_2) \\
&\ldots \\
h(x_n) &= y_n - F(x_n)
\end{aligned}
$$

Can any regression tree h achieve this goal perfectly?
Maybe not....
But some regression tree might be able to do this approximately.
How?

Gradient Boosting for Regression

Simple solution:

Or, equivalently, you wish

$$
\begin{aligned}
h(x_1) &= y_1 - F(x_1) \\
h(x_2) &= y_2 - F(x_2) \\
&\dots \\
h(x_n) &= y_n - F(x_n)
\end{aligned}
$$

Can any regression tree h achieve this goal perfectly?
Maybe not....
But some regression tree might be able to do this approximately.
How?
Just fit a regression tree h to data
$(x_1, y_1 - F(x_1)), (x_2, y_2 - F(x_2)), \dots, (x_n, y_n - F(x_n))$
新的模型对应的新的样本数据。y变成了前一个模型的残差。
前一个模型残差大的y在新的模型中会“被重视”。

Gradient Boosting for Regression

Simple solution:

Or, equivalently, you wish

$$
\begin{aligned}
h(x_1) &= y_1 - F(x_1) \\
h(x_2) &= y_2 - F(x_2) \\
&\dots \\
h(x_n) &= y_n - F(x_n)
\end{aligned}
$$

Can any regression tree h achieve this goal perfectly?
Maybe not....
But some regression tree might be able to do this approximately.
How?
Just fit a regression tree h to data
$(x_1, y_1 - F(x_1)), (x_2, y_2 - F(x_2)), ..., (x_n, y_n - F(x_n))$
Congratulations, you get a better model!

Gradient Boosting for Regression

Simple solution: 残差表示已有模型的不足

$y_i - F(x_i)$ are called **residuals**. These are the parts that existing model F cannot do well.

The role of h is to compensate the shortcoming of existing model F.

Gradient Boosting for Regression

Simple solution:

$y_i - F(x_i)$ are called **residuals**. These are the parts that existing model F cannot do well.

The role of h is to compensate the shortcoming of existing model F.

If the new model $F + h$ is still not satisfactory,

Gradient Boosting for Regression

Simple solution:

$y_i - F(x_i)$ are called **residuals**. These are the parts that existing model F cannot do well.

The role of h is to compensate the shortcoming of existing model F.

If the new model $F + h$ is still not satisfactory, we can add another regression tree...

Gradient Boosting for Regression

Simple solution:

$y_i - F(x_i)$ are called **residuals**. These are the parts that existing model F cannot do well.

The role of h is to compensate the shortcoming of existing model F.

If the new model $F + h$ is still not satisfactory, we can add another regression tree...

We are improving the predictions of training data, is the procedure also useful for test data?

Gradient Boosting for Regression

Simple solution:

$y_i - F(x_i)$ are called **residuals**. These are the parts that existing model F cannot do well.

The role of h is to compensate the shortcoming of existing model F.

If the new model $F + h$ is still not satisfactory, we can add another regression tree...

We are improving the predictions of training data, is the procedure also useful for test data?

Yes! Because we are building a model, and the model can be applied to test data as well.

Gradient Boosting for Regression

Simple solution:

$y_i - F(x_i)$ are called **residuals**. These are the parts that existing model F cannot do well.

The role of h is to compensate the shortcoming of existing model F.

If the new model $F + h$ is still not satisfactory, we can add another regression tree...

We are improving the predictions of training data, is the procedure also useful for test data?

Yes! Because we are building a model, and the model can be applied to test data as well.

How is this related to gradient descent?

Gradient Boosting for Regression

Gradient Descent

Minimize a function by moving in the opposite direction of the gradient.

$$\theta_i := \theta_i - \rho \frac{\partial J}{\partial \theta_i}$$

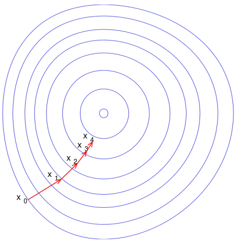


Figure: Gradient Descent. Source: http://en.wikipedia.org/wiki/Gradient_descent

Gradient Boosting for Regression

How is this related to gradient descent?

Loss function $L(y, F(x)) = (y - F(x))^2/2$
We want to minimize $J = \sum_i L(y_i, F(x_i))$ by adjusting $F(x_1), F(x_2), ..., F(x_n)$.
Notice that $F(x_1), F(x_2), ..., F(x_n)$ are just some numbers. We can treat $F(x_i)$ as parameters and take derivatives

$$\frac{\partial J}{\partial F(x_i)} = \frac{\partial \sum_i L(y_i, F(x_i))}{\partial F(x_i)} = \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} = F(x_i) - y_i$$

So we can interpret residuals as negative gradients.

$$y_i - F(x_i) = -\frac{\partial J}{\partial F(x_i)}$$

Gradient Boosting for Regression

How is this related to gradient descent?

$$
\begin{aligned}
F(x_i) &:= F(x_i) + h(x_i) \\
F(x_i) &:= F(x_i) + y_i - F(x_i) \\
F(x_i) &:= F(x_i) - 1\frac{\partial J}{\partial F(x_i)} \\
\theta_i &:= \theta_i - \rho\frac{\partial J}{\partial \theta_i}
\end{aligned}
$$

Gradient Boosting for Regression

How is this related to gradient descent?

For regression with **square loss**,

$$\textit{residual} \Leftrightarrow \textit{negative gradient}$$

$$\textit{fit } h \textit{ to residual} \Leftrightarrow \textit{fit } h \textit{ to negative gradient}$$

$$\textit{update } F \textit{ based on residual} \Leftrightarrow \textit{update } F \textit{ based on negative gradient}$$

Gradient Boosting for Regression

How is this related to gradient descent?

For regression with **square loss**,

$$\textit{residual} \Leftrightarrow \textit{negative gradient}$$

$$\textit{fit } h \textit{ to residual} \Leftrightarrow \textit{fit } h \textit{ to negative gradient}$$

$$\textit{update } F \textit{ based on residual} \Leftrightarrow \textit{update } F \textit{ based on negative gradient}$$

So we are actually updating our model using **gradient descent**!

Gradient Boosting for Regression

How is this related to gradient descent?

For regression with **square loss**,

$$\textit{residual} \Leftrightarrow \textit{negative gradient}$$

$$\textit{fit } h \textit{ to residual} \Leftrightarrow \textit{fit } h \textit{ to negative gradient}$$

$$\textit{update } F \textit{ based on residual} \Leftrightarrow \textit{update } F \textit{ based on negative gradient}$$

So we are actually updating our model using **gradient descent**!
It turns out that the concept of **gradients** is more general and useful than the concept of **residuals**. So from now on, let's stick with gradients. The reason will be explained later.

Gradient Boosting for Regression

Regression with square Loss

Let us summarize the algorithm we just derived using the concept of gradients. Negative gradient:

$$-g(x_i) = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} = y_i - F(x_i)$$

start with an initial model, say, $F(x) = \frac{\sum_{i=1}^{n} y_i}{n}$
iterate until converge:
  calculate negative gradients $-g(x_i)$
  fit a regression tree h to negative gradients $-g(x_i)$
  $F := F + \rho h$, where $\rho = 1$

Gradient Boosting for Regression

Regression with square Loss

Let us summarize the algorithm we just derived using the concept of gradients. Negative gradient:

$$-g(x_i) = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} = y_i - F(x_i)$$

start with an initial model, say, $F(x) = \frac{\sum_{i=1}^{n} y_i}{n}$
iterate until converge:
    calculate negative gradients $-g(x_i)$
    fit a regression tree h to negative gradients $-g(x_i)$
    $F := F + \rho h$, where $\rho = 1$

The benefit of formulating this algorithm using gradients is that it allows us to consider other loss functions and derive the corresponding algorithms in the same way.

Gradient Boosting for Regression

Loss Functions for Regression Problem

Why do we need to consider other loss functions? Isn't square loss good enough?

Loss Functions for Regression Problem

Square loss is:

- ✓ Easy to deal with mathematically
- × Not robust to outliers

  Outliers are heavily punished because the error is squared.

  Example:

y_i	0.5	1.2	2	5*
$F(x_i)$	0.6	1.4	1.5	1.7
$L = (y - F)^2/2$	0.005	0.02	0.125	5.445

  Consequence?

Loss Functions for Regression Problem

Square loss is:

- ✓ Easy to deal with mathematically
- × Not robust to outliers 对于异常值不够健壮

  Outliers are heavily punished because the error is squared.

  Example:

y_i	0.5	1.2	2	5*
$F(x_i)$	0.6	1.4	1.5	1.7
$L = (y - F)^2/2$	0.005	0.02	0.125	5.445

  Consequence?

  Pay too much attention to outliers. Try hard to incorporate outliers into the model. Degrade the overall performance.

Gradient Boosting for Regression

Loss Functions for Regression Problem

- Absolute loss (more robust to outliers)

$$L(y, F) = |y - F|$$

Gradient Boosting for Regression

Loss Functions for Regression Problem

- Absolute loss (more robust to outliers)

$$L(y, F) = |y - F|$$

- Huber loss (more robust to outliers)

$$L(y, F) = \begin{cases} \frac{1}{2}(y - F)^2 & |y - F| \leq \delta \\ \delta(|y - F| - \delta/2) & |y - F| > \delta \end{cases}$$

Loss Functions for Regression Problem

- Absolute loss (more robust to outliers)

$$L(y, F) = |y - F|$$

- Huber loss (more robust to outliers)

$$L(y, F) = \begin{cases} \frac{1}{2}(y - F)^2 & |y - F| \leq \delta \\ \delta(|y - F| - \delta/2) & |y - F| > \delta \end{cases}$$

y_i	0.5	1.2	2	5*
$F(x_i)$	0.6	1.4	1.5	1.7
Square loss	0.005	0.02	0.125	5.445
Absolute loss	0.1	0.2	0.5	3.3
Huber loss($\delta = 0.5$)	0.005	0.02	0.125	1.525

Gradient Boosting for Regression

Regression with Absolute Loss

Negative gradient:

$$-g(x_i) = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} = sign(y_i - F(x_i))$$

start with an initial model, say, $F(x) = \frac{\sum_{i=1}^{n} y_i}{n}$
iterate until converge:
    calculate gradients $-g(x_i)$
    fit a regression tree h to negative gradients $-g(x_i)$
    $F := F + \rho h$

Gradient Boosting for Regression

Regression with Huber Loss

Negative gradient:

$$
\begin{aligned}
-g(x_i) &= -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \\
&= \begin{cases} y_i - F(x_i) & |y_i - F(x_i)| \leq \delta \\ \delta sign(y_i - F(x_i)) & |y_i - F(x_i)| > \delta \end{cases}
\end{aligned}
$$

start with an initial model, say, $F(x) = \frac{\sum_{i=1}^{n} y_i}{n}$

iterate until converge:

calculate negative gradients $-g(x_i)$

fit a regression tree h to negative gradients $-g(x_i)$

$F := F + \rho h$

Gradient Boosting for Regression

Regression with loss function L: general procedure

Give any differentiable loss function L

start with an initial model, say $F(x) = \frac{\sum_{i=1}^{n} y_i}{n}$
iterate until converge:
  calculate negative gradients $-g(x_i) = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}$
  fit a regression tree h to negative gradients $-g(x_i)$
  $F := F + \rho h$

Gradient Boosting for Regression

Regression with loss function L: general procedure

Give any differentiable loss function L

start with an initial model, say $F(x) = \frac{\sum_{i=1}^{n} y_i}{n}$
iterate until converge:
calculate negative gradients $-g(x_i) = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}$
fit a regression tree h to negative gradients $-g(x_i)$
$F := F + \rho h$

In general,

$$\textit{negative gradients} \not\Leftrightarrow \textit{residuals}$$

We should follow negative gradients rather than residuals. Why?

Gradient Boosting for Regression

Negative Gradient vs Residual: An Example

Huber loss

$$L(y, F) = \begin{cases} \frac{1}{2}(y - F)^2 & |y - F| \leq \delta \\ \delta(|y - F| - \delta/2) & |y - F| > \delta \end{cases}$$

Update by Negative Gradient:

$$h(x_i) = -g(x_i) = \begin{cases} y_i - F(x_i) & |y_i - F(x_i)| \leq \delta \\ \delta sign(y_i - F(x_i)) & |y_i - F(x_i)| > \delta \end{cases}$$

Update by Residual:

$$h(x_i) = y_i - F(x_i)$$

残差是用square Loss损失函数时的负梯度，故有square Loss的缺陷。

Difference: negative gradient pays less attention to outliers.

Gradient Boosting for Regression

Summary of the Section

- Fit an additive model $F = \sum_t \rho_t h_t$ in a forward stage-wise manner.
- In each stage, introduce a new regression tree h to compensate the shortcomings of existing model.
- The "shortcomings" are identified by negative gradients.
- For any loss function, we can derive a gradient boosting algorithm.
- Absolute loss and Huber loss are more robust to outliers than square loss.

Things not covered

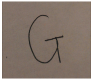
How to choose a proper learning rate for each gradient boosting algorithm. See [Friedman, 2001]

Gradient Boosting for Classification

Problem

Recognize the given hand written capital letter.

- Multi-class classification
- 26 classes. A,B,C,...,Z

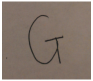


Data Set

- `http://archive.ics.uci.edu/ml/datasets/Letter+Recognition`
- 20000 data points, 16 features

Gradient Boosting for Classification

Feature Extraction



1	horizontal position of box	9	mean y variance
2	vertical position of box	10	mean x y correlation
3	width of box	11	mean of x * x * y
4	height of box	12	mean of x * y * y
5	total number on pixels	13	mean edge count left to right
6	mean x of on pixels in box	14	correlation of x-ege with y
7	mean y of on pixels in box	15	mean edge count bottom to top
8	mean x variance	16	correlation of y-ege with x

Feature Vector$= (2, 1, 3, 1, 1, 8, 6, 6, 6, 6, 5, 9, 1, 7, 5, 10)$
Label $=$ G

Gradient Boosting for Classification

Model

- 26 score functions (our models): $F_A, F_B, F_C, ..., F_Z$.
- $F_A(x)$ assigns a score for class A
- scores are used to calculate probabilities

$$P_A(x) = \frac{e^{F_A(x)}}{\sum_{c=A}^{Z} e^{F_c(x)}}$$

$$P_B(x) = \frac{e^{F_B(x)}}{\sum_{c=A}^{Z} e^{F_c(x)}}$$

$$...$$

$$P_Z(x) = \frac{e^{F_Z(x)}}{\sum_{c=A}^{Z} e^{F_c(x)}}$$

- predicted label = class that has the highest probability

Loss Function for each data point

Step 1 turn the label y_i into a (true) probability distribution $Y_c(x_i)$

For example: y_5=G,

$Y_A(x_5) = 0, Y_B(x_5) = 0, ..., Y_G(x_5) = 1, ..., Y_Z(x_5) = 0$

Gradient Boosting for Classification

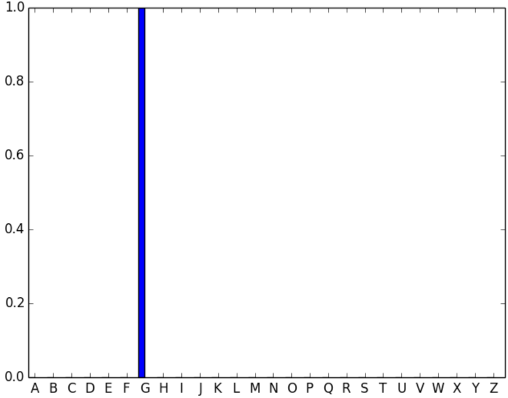


Figure: true probability distribution

Loss Function for each data point

Step 1 turn the label y_i into a (true) probability distribution $Y_c(x_i)$
For example: y_5=G,
$Y_A(x_5) = 0$, $Y_B(x_5) = 0, ..., Y_G(x_5) = 1, ..., Y_Z(x_5) = 0$

Step 2 calculate the predicted probability distribution $P_c(x_i)$ based on the current model $F_A, F_B, ..., F_Z$.
$P_A(x_5) = 0.03, P_B(x_5) = 0.05, ..., P_G(x_5) = 0.3, ..., P_Z(x_5) = 0.05$

Gradient Boosting for Classification

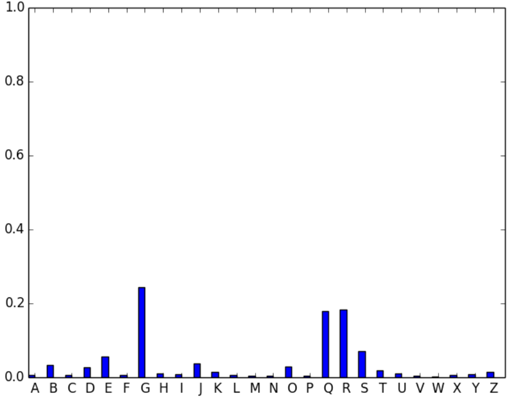


Figure: predicted probability distribution based on current model

Loss Function for each data point

Step 1 turn the label y_i into a (true) probability distribution $Y_c(x_i)$
For example: y_5=G,
$Y_A(x_5) = 0, Y_B(x_5) = 0, ..., Y_G(x_5) = 1, ..., Y_Z(x_5) = 0$

Step 2 calculate the predicted probability distribution $P_c(x_i)$ based on the current model $F_A, F_B, ..., F_Z$.
$P_A(x_5) = 0.03, P_B(x_5) = 0.05, ..., P_G(x_5) = 0.3, ..., P_Z(x_5) = 0.05$

Step 3 calculate the difference between the true probability distribution and the predicted probability distribution.
Here we use KL-divergence

Gradient Boosting for Classification

Goal

- minimize the total loss (KL-divergence)
- for each data point, we wish the predicted probability distribution to match the true probability distribution as closely as possible

Gradient Boosting for Classification

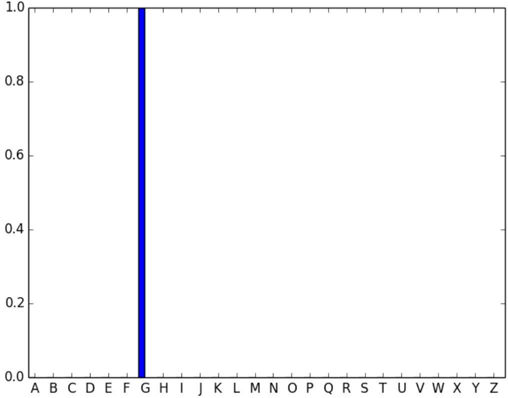


Figure: true probability distribution

Gradient Boosting for Classification

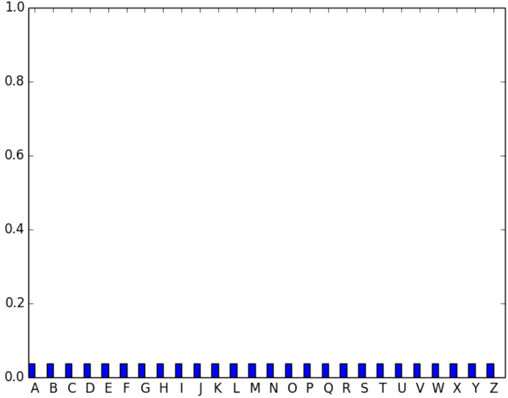


Figure: predicted probability distribution at round 0

Gradient Boosting for Classification

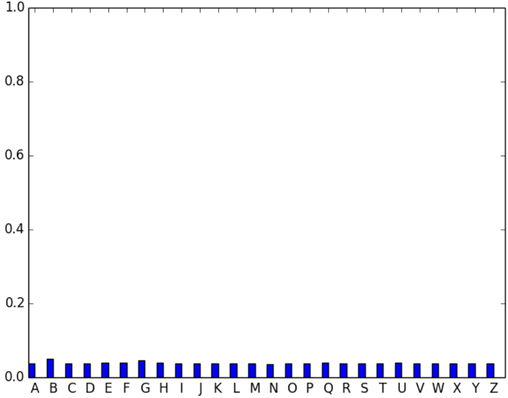


Figure: predicted probability distribution at round 1

Gradient Boosting for Classification

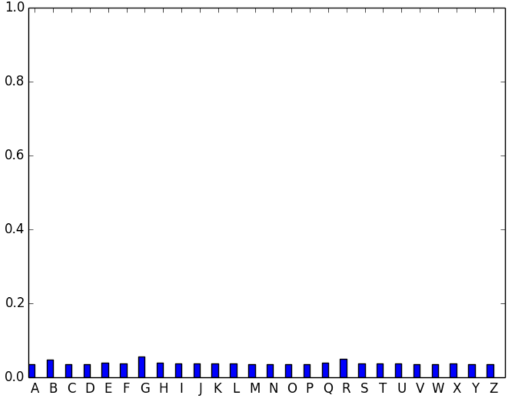


Figure: predicted probability distribution at round 2

Gradient Boosting for Classification

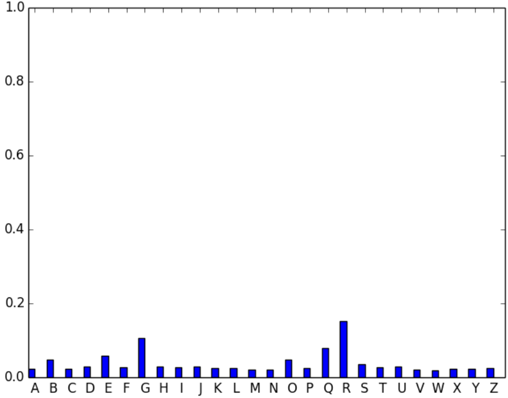


Figure: predicted probability distribution at round 10

Gradient Boosting for Classification

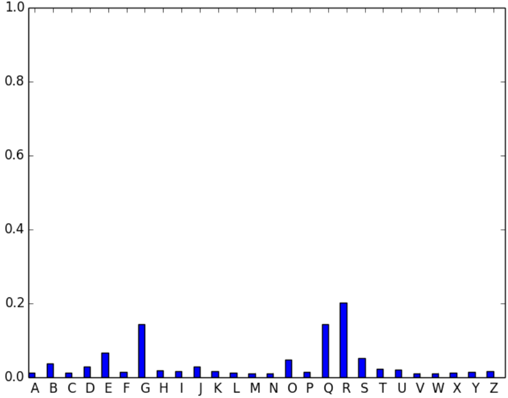


Figure: predicted probability distribution at round 20

Gradient Boosting for Classification

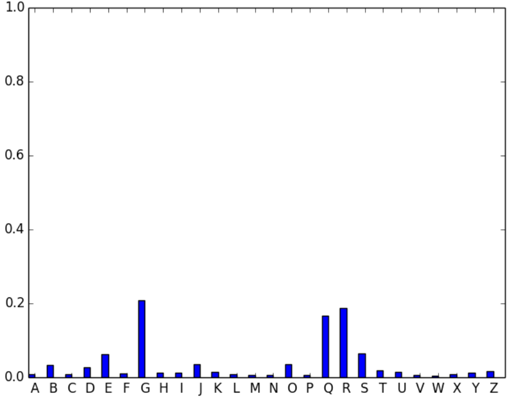


Figure: predicted probability distribution at round 30

Gradient Boosting for Classification

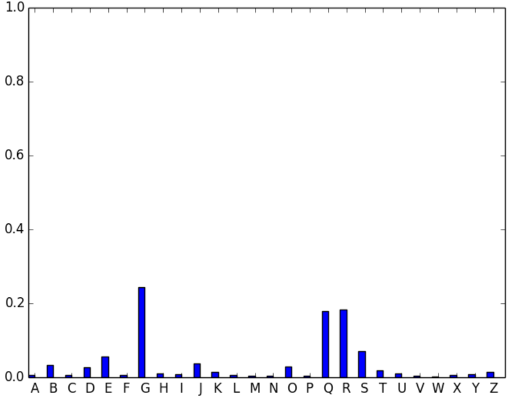


Figure: predicted probability distribution at round 40

Gradient Boosting for Classification

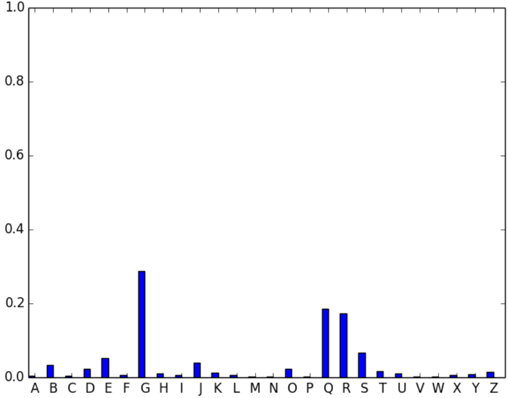


Figure: predicted probability distribution at round 50

Gradient Boosting for Classification

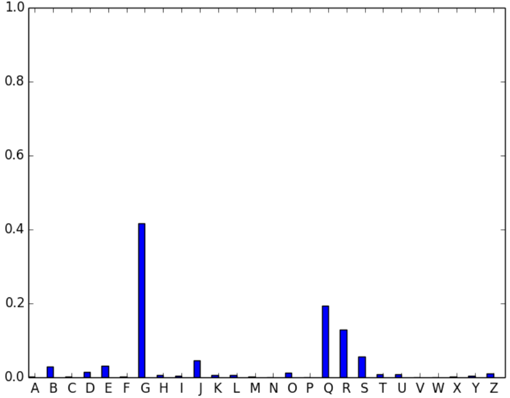


Figure: predicted probability distribution at round 100

Gradient Boosting for Classification

Goal

- minimize the total loss (KL-divergence)
- for each data point, we wish the predicted probability distribution to match the true probability distribution as closely as possible
- we achieve this goal by adjusting our models $F_A, F_B, ..., F_Z$.

Gradient Boosting for Regression: Review

Regression with loss function L: general procedure

Give any differentiable loss function L

start with an initial model F
iterate until converge:
    calculate negative gradients $-g(x_i) = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}$
    fit a regression tree h to negative gradients $-g(x_i)$
    $F := F + \rho h$

Gradient Boosting for Classification

Differences

- $F_A, F_B, ..., F_Z$ vs F
- a matrix of parameters to optimize vs a column of parameters to optimize

$F_A(x_1)$	$F_B(x_1)$	...	$F_Z(x_1)$
$F_A(x_2)$	$F_B(x_2)$	...	$F_Z(x_2)$
...	...	...	...
$F_A(x_n)$	$F_B(x_n)$	...	$F_Z(x_n)$

- a matrix of gradients vs a column of gradients

$\frac{\partial L}{F_A(x_1)}$	$\frac{\partial L}{F_B(x_1)}$	...	$\frac{\partial L}{F_Z(x_1)}$
$\frac{\partial L}{F_A(x_2)}$	$\frac{\partial L}{F_B(x_2)}$	...	$\frac{\partial L}{F_Z(x_2)}$
...	...	...	...
$\frac{\partial L}{F_A(x_n)}$	$\frac{\partial L}{F_B(x_n)}$	...	$\frac{\partial L}{F_Z(x_n)}$

Gradient Boosting for Classification

start with initial models $F_A, F_B, F_C, ..., F_Z$
iterate until converge:
    calculate negative gradients for class A: $-g_A(x_i) = -\frac{\partial L}{\partial F_A(x_i)}$
    calculate negative gradients for class B: $-g_B(x_i) = -\frac{\partial L}{\partial F_B(x_i)}$
    ...
    calculate negative gradients for class Z:$-g_Z(x_i) = -\frac{\partial L}{\partial F_Z(x_i)}$
    fit a regression tree h_A to negative gradients $-g_A(x_i)$
    fit a regression tree h_B to negative gradients $-g_B(x_i)$
    ...
    fit a regression tree h_Z to negative gradients $-g_Z(x_i)$
    $F_A := F_A + \rho_A h_A$
    $F_B := F_B + \rho_B h_B$
    ...
    $F_Z := F_Z + \rho_Z h_Z$

Gradient Boosting for Classification

start with initial models $F_A, F_B, F_C, ..., F_Z$
iterate until converge:
    calculate negative gradients for class A: $-g_A(x_i) = Y_A(x_i) - P_A(x_i)$
    calculate negative gradients for class B: $-g_B(x_i) = Y_B(x_i) - P_B(x_i)$
    ...
    calculate negative gradients for class Z:$-g_Z(x_i) = Y_Z(x_i) - P_Z(x_i)$
    fit a regression tree h_A to negative gradients $-g_A(x_i)$
    fit a regression tree h_B to negative gradients $-g_B(x_i)$
    ...
    fit a regression tree h_Z to negative gradients $-g_Z(x_i)$
    $F_A := F_A + \rho_A h_A$
    $F_B := F_B + \rho_B h_B$
    ...
    $F_Z := F_Z + \rho_Z h_Z$

Gradient Boosting for Classification

round 0

i	y	Y_A	Y_B	Y_C	Y_D	Y_E	Y_F	Y_G	Y_H	Y_I	Y_J	Y_K	Y_L	Y_M	Y_N	Y_O	Y_P	Y_Q	Y_R	Y_S	Y_T	Y_U	Y_V	Y_W	Y_X	Y_Y	Y_Z
1	T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
2	I	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	D	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	N	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
5	G	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

i	y	F_A	F_B	F_C	F_D	F_E	F_F	F_G	F_H	F_I	F_J	F_K	F_L	F_M	F_N	F_O	F_P	F_Q	F_R	F_S	F_T	F_U	F_V	F_W	F_X	F_Y	F_Z
1	T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

i	y	P_A	P_B	P_C	P_D	P_E	P_F	P_G	P_H	P_I	P_J	P_K	P_L	P_M	P_N	P_O	P_P	P_Q	P_R	P_S	P_T	P_U	P_V	P_W	P_X	P_Y	P_Z
1	T	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
2	I	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
3	D	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
4	N	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
5	G	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

i	y	$Y_A - P_A$	$Y_B - P_B$	$Y_C - P_C$	$Y_D - P_D$	$Y_E - P_E$	$Y_F - P_F$	$Y_G - P_G$	$Y_H - P_H$	$Y_I - P_I$	$Y_J - P_J$	$Y_K - P_K$	$Y_L - P_L$	$Y_M - P_M$	$Y_N - P_N$	$Y_O - P_O$	$Y_P - P_P$	$Y_Q - P_Q$	$Y_R - P_R$	$Y_S - P_S$	$Y_T - P_T$	$Y_U - P_U$	$Y_V - P_V$	$Y_W - P_W$	$Y_X - P_X$	$Y_Y - P_Y$	$Y_Z - P_Z$
1	T	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	0.96	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
2	I	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	0.96	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
3	D	-0.04	-0.04	-0.04	0.96	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
4	N	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	0.96	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
5	G	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	0.96	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Gradient Boosting for Classification

$$h_A(x) = \begin{cases} 0.98 & \textit{feature } 10 \textit{ of } x \leq 2.0 \\ -0.07 & \textit{feature } 10 \textit{ of } x > 2.0 \end{cases}$$

$$h_B(x) = \begin{cases} -0.07 & \textit{feature } 15 \textit{ of } x \leq 8.0 \\ 0.22 & \textit{feature } 15 \textit{ of } x > 8.0 \end{cases}$$

$$\dots$$

$$h_Z(x) = \begin{cases} -0.07 & \textit{feature } 8 \textit{ of } x \leq 8.0 \\ 0.82 & \textit{feature } 8 \textit{ of } x > 8.0 \end{cases}$$

$$\begin{aligned} F_A &:= F_A + \rho_A h_A \\ F_B &:= F_B + \rho_B h_B \\ &\dots \\ F_Z &:= F_Z + \rho_Z h_Z \end{aligned}$$

Gradient Boosting for Classification

round 1

i	y	Y_A	Y_B	Y_C	Y_D	Y_E	Y_F	Y_G	Y_H	Y_I	Y_J	Y_K	Y_L	Y_M	Y_N	Y_O	Y_P	Y_Q	Y_R	Y_S	Y_T	Y_U	Y_V	Y_W	Y_X	Y_Y	Y_Z
1	T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
2	I	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	D	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	N	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
5	G	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

i	y	F_A	F_B	F_C	F_D	F_E	F_F	F_G	F_H	F_I	F_J	F_K	F_L	F_M	F_N	F_O	F_P	F_Q	F_R	F_S	F_T	F_U	F_V	F_W	F_X	F_Y	F_Z
1	T	-0.08	-0.07	-0.06	-0.07	-0.02	-0.02	-0.08	-0.02	-0.03	-0.03	-0.06	-0.04	-0.08	-0.08	-0.07	-0.07	-0.02	-0.04	-0.04	0.59	-0.01	-0.07	-0.07	-0.05	-0.06	-0.07
2	I	-0.08	0.23	-0.06	-0.07	-0.02	-0.02	0.16	-0.02	-0.03	-0.03	-0.06	-0.04	-0.08	-0.08	-0.07	-0.07	-0.02	-0.04	-0.04	-0.07	-0.01	-0.07	-0.07	-0.05	-0.06	-0.07
3	D	-0.08	0.23	-0.06	-0.07	-0.02	-0.02	-0.08	-0.02	-0.03	-0.03	-0.06	-0.04	-0.08	-0.08	-0.07	-0.07	-0.02	-0.04	-0.04	-0.07	-0.01	-0.07	-0.07	-0.05	-0.06	-0.07
4	N	-0.08	-0.07	-0.06	-0.07	-0.02	-0.02	0.16	-0.02	-0.03	-0.03	0.26	-0.04	-0.08	0.3	-0.07	-0.07	-0.02	-0.04	-0.04	-0.07	-0.01	-0.07	-0.07	-0.05	-0.06	-0.07
5	G	-0.08	0.23	-0.06	-0.07	-0.02	-0.02	0.16	-0.02	-0.03	-0.03	-0.06	-0.04	-0.08	-0.08	-0.07	-0.07	-0.02	-0.04	-0.04	-0.07	-0.01	-0.07	-0.07	-0.05	-0.06	-0.07
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

i	y	P_A	P_B	P_C	P_D	P_E	P_F	P_G	P_H	P_I	P_J	P_K	P_L	P_M	P_N	P_O	P_P	P_Q	P_R	P_S	P_T	P_U	P_V	P_W	P_X	P_Y	P_Z
1	T	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.07	0.04	0.04	0.04	0.04	0.04	0.04
2	I	0.04	0.05	0.04	0.04	0.04	0.04	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
3	D	0.04	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
4	N	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.04	0.04	0.04	0.05	0.04	0.04	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
5	G	0.04	0.05	0.04	0.04	0.04	0.04	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

i	y	$Y_A - P_A$	$Y_B - P_B$	$Y_C - P_C$	$Y_D - P_D$	$Y_E - P_E$	$Y_F - P_F$	$Y_G - P_G$	$Y_H - P_H$	$Y_I - P_I$	$Y_J - P_J$	$Y_K - P_K$	$Y_L - P_L$	$Y_M - P_M$	$Y_N - P_N$	$Y_O - P_O$	$Y_P - P_P$	$Y_Q - P_Q$	$Y_R - P_R$	$Y_S - P_S$	$Y_T - P_T$	$Y_U - P_U$	$Y_V - P_V$	$Y_W - P_W$	$Y_X - P_X$	$Y_Y - P_Y$	$Y_Z - P_Z$
1	T	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	0.93	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
2	I	-0.04	-0.05	-0.04	-0.04	-0.04	-0.04	-0.05	-0.04	0.96	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
3	D	-0.04	-0.05	-0.04	0.96	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
4	N	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.04	-0.04	-0.04	-0.05	-0.04	-0.04	0.95	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
5	G	-0.04	-0.05	-0.04	-0.04	-0.04	-0.04	0.95	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Gradient Boosting for Classification

$$h_A(x) = \begin{cases} 0.37 & \textit{feature } 10 \textit{ of } x \leq 2.0 \\ -0.07 & \textit{feature } 10 \textit{ of } x > 2.0 \end{cases}$$

$$h_B(x) = \begin{cases} -0.07 & \textit{feature } 14 \textit{ of } x \leq 5.0 \\ 0.22 & \textit{feature } 14 \textit{ of } x > 5.0 \end{cases}$$

$$\ldots$$

$$h_Z(x) = \begin{cases} -0.07 & \textit{feature } 8 \textit{ of } x \leq 8.0 \\ 0.35 & \textit{feature } 8 \textit{ of } x > 8.0 \end{cases}$$

$$\begin{aligned} F_A &:= F_A + \rho_A h_A \\ F_B &:= F_B + \rho_B h_B \\ &\ldots \\ F_Z &:= F_Z + \rho_Z h_Z \end{aligned}$$

Gradient Boosting for Classification

round 2

i	y	Y_A	Y_B	Y_C	Y_D	Y_E	Y_F	Y_G	Y_H	Y_I	Y_J	Y_K	Y_L	Y_M	Y_N	Y_O	Y_P	Y_Q	Y_R	Y_S	Y_T	Y_U	Y_V	Y_W	Y_X	Y_Y	Y_Z
1	T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
2	I	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	D	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	N	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
5	G	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

i	y	F_A	F_B	F_C	F_D	F_E	F_F	F_G	F_H	F_I	F_J	F_K	F_L	F_M	F_N	F_O	F_P	F_Q	F_R	F_S	F_T	F_U	F_V	F_W	F_X	F_Y	F_Z
1	T	-0.15	-0.14	-0.12	-0.14	-0.03	0.28	-0.14	-0.04	1.49	-0.07	-0.11	-0.08	-0.14	-0.17	-0.13	-0.13	-0.04	-0.11	-0.07	1.05	0.19	0.25	-0.16	-0.09	0.33	-0.14
2	I	-0.15	0.16	-0.12	-0.14	-0.03	-0.08	0.33	-0.04	-0.07	-0.07	-0.11	-0.08	-0.14	-0.17	-0.13	-0.13	-0.04	-0.11	-0.07	-0.11	-0.07	-0.15	-0.16	-0.09	-0.13	-0.14
3	D	-0.15	0.16	-0.12	-0.14	-0.03	-0.08	0.1	-0.04	-0.07	-0.07	-0.11	-0.08	-0.14	-0.17	-0.13	-0.13	-0.04	0.19	-0.07	-0.11	-0.07	-0.15	-0.16	-0.09	-0.13	-0.14
4	N	-0.15	-0.14	-0.12	-0.14	-0.03	-0.08	0.1	-0.04	-0.07	-0.07	0.46	-0.08	-0.14	0.5	-0.13	-0.13	-0.04	-0.11	-0.07	-0.11	-0.07	-0.15	0.25	-0.09	-0.13	-0.14
5	G	-0.15	0.16	-0.12	-0.14	-0.03	-0.08	0.33	-0.04	-0.07	-0.07	-0.11	-0.08	-0.14	-0.17	-0.13	-0.13	-0.04	0.19	-0.07	-0.11	-0.07	-0.15	-0.16	-0.09	-0.13	-0.14
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

i	y	P_A	P_B	P_C	P_D	P_E	P_F	P_G	P_H	P_I	P_J	P_K	P_L	P_M	P_N	P_O	P_P	P_Q	P_R	P_S	P_T	P_U	P_V	P_W	P_X	P_Y	P_Z
1	T	0.03	0.03	0.03	0.03	0.03	0.04	0.03	0.03	0.15	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.09	0.04	0.04	0.03	0.03	0.05	0.03
2	I	0.04	0.05	0.04	0.04	0.04	0.04	0.06	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
3	D	0.04	0.05	0.04	0.04	0.04	0.04	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
4	N	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.06	0.04	0.03	0.06	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.03	0.05	0.04	0.03	0.03
5	G	0.03	0.05	0.04	0.04	0.04	0.04	0.06	0.04	0.04	0.04	0.04	0.04	0.04	0.03	0.04	0.04	0.04	0.05	0.04	0.04	0.04	0.04	0.03	0.04	0.04	0.04
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

i	y	$Y_A - P_A$	$Y_B - P_B$	$Y_C - P_C$	$Y_D - P_D$	$Y_E - P_E$	$Y_F - P_F$	$Y_G - P_G$	$Y_H - P_H$	$Y_I - P_I$	$Y_J - P_J$	$Y_K - P_K$	$Y_L - P_L$	$Y_M - P_M$	$Y_N - P_N$	$Y_O - P_O$	$Y_P - P_P$	$Y_Q - P_Q$	$Y_R - P_R$	$Y_S - P_S$	$Y_T - P_T$	$Y_U - P_U$	$Y_V - P_V$	$Y_W - P_W$	$Y_X - P_X$	$Y_Y - P_Y$	$Y_Z - P_Z$
1	T	-0.03	-0.03	-0.03	-0.03	-0.03	-0.04	-0.03	-0.03	-0.15	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	0.91	-0.04	-0.04	-0.03	-0.03	-0.05	-0.03
2	I	-0.04	-0.05	-0.04	-0.04	-0.04	-0.04	-0.06	-0.04	0.96	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
3	D	-0.04	-0.05	-0.04	0.96	-0.04	-0.04	-0.05	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
4	N	-0.03	-0.03	-0.03	-0.03	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.06	-0.04	-0.03	0.94	-0.03	-0.03	-0.04	-0.04	-0.04	-0.04	-0.04	-0.03	-0.05	-0.04	-0.03	-0.03
5	G	-0.03	-0.05	-0.04	-0.04	-0.04	-0.04	0.94	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.03	-0.04	-0.04	-0.04	-0.05	-0.04	-0.04	-0.04	-0.04	-0.03	-0.04	-0.04	-0.04
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Gradient Boosting for Classification

round 100

i	y	Y_A	Y_B	Y_C	Y_D	Y_E	Y_F	Y_G	Y_H	Y_I	Y_J	Y_K	Y_L	Y_M	Y_N	Y_O	Y_P	Y_Q	Y_R	Y_S	Y_T	Y_U	Y_V	Y_W	Y_X	Y_Y	Y_Z
1	T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
2	I	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	D	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	N	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
5	G	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

i	y	F_A	F_B	F_C	F_D	F_E	F_F	F_G	F_H	F_I	F_J	F_K	F_L	F_M	F_N	F_O	F_P	F_Q	F_R	F_S	F_T	F_U	F_V	F_W	F_X	F_Y	F_Z
1	T	-3.26	-2.7	-2.2	-2.22	-2.48	-0.31	-2.77	-1.19	2.77	0.1	-1.49	-1.02	-1.64	-0.8	-2.4	-3.57	-0.9	-2.45	-0.2	4.61	0.5	-0.71	-1.21	-0.24	0.49	-1.66
2	I	-1.64	-1.09	-2.29	-1.8	0.45	-0.43	2.14	-1.56	1.19	1.09	-1.5	-0.5	-3.64	-3.98	-0.39	-2.3	1.42	-0.59	0.27	-2.88	-1.96	-1.67	-4.38	-2.06	-2.95	-1.76
3	D	-2.45	0.18	-3.01	0.18	-2.79	-1.7	-2.21	0.43	-1.12	0.32	0.67	-2.16	-2.91	-2.76	-1.92	-3.04	-1.47	-0.48	-1.48	-1.25	-2.25	-3.23	-4.38	0.17	-2.95	-2.65
4	N	-3.95	-3.38	-0.22	-0.94	-1.33	-1.38	-1.22	-0.12	-2.33	-3.13	0.58	-0.65	-0.25	2.96	-2.84	-1.82	0.19	0.55	-1.22	-1.25	0.45	-1.8	0.11	-0.69	-1.6	-3.78
5	G	-3.14	-0.04	-2.37	-0.78	0.02	-2.68	2.6	-1.48	-1.93	0.42	-1.44	-1.45	-3.36	-3.98	-0.94	-3.42	1.84	1.44	0.62	-1.25	-1.33	-4.41	-4.71	-2.62	-2.15	-1.09
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

i	y	P_A	P_B	P_C	P_D	P_E	P_F	P_G	P_H	P_I	P_J	P_K	P_L	P_M	P_N	P_O	P_P	P_Q	P_R	P_S	P_T	P_U	P_V	P_W	P_X	P_Y	P_Z
1	T	0	0	0	0	0	0.01	0	0	0.13	0.01	0	0	0	0	0	0	0	0	0.01	0.79	0.01	0	0	0.01	0.01	0
2	I	0.01	0.01	0	0.01	0.06	0.02	0.32	0.01	0.12	0.11	0.01	0.02	0	0	0.03	0	0.16	0.02	0.05	0	0.01	0.01	0	0	0	0.01
3	D	0.01	0.11	0	0.11	0.01	0.02	0.01	0.14	0.03	0.12	0.17	0.01	0	0.01	0.01	0	0.02	0.05	0.02	0.03	0.01	0	0	0.11	0	0.01
4	N	0	0	0.02	0.01	0.01	0.01	0.01	0.03	0	0	0.05	0.02	0.02	0.59	0	0	0.04	0.05	0.01	0.01	0.05	0.01	0.03	0.02	0.01	0
5	G	0	0.03	0	0.01	0.03	0	0.42	0.01	0	0.05	0.01	0.01	0	0	0.01	0	0.19	0.13	0.06	0.01	0.01	0	0	0	0	0.01
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

i	y	Y_A-P_A	Y_B-P_B	Y_C-P_C	Y_D-P_D	Y_E-P_E	Y_F-P_F	Y_G-P_G	Y_H-P_H	Y_I-P_I	Y_J-P_J	Y_K-P_K	Y_L-P_L	Y_M-P_M	Y_N-P_N	Y_O-P_O	Y_P-P_P	Y_Q-P_Q	Y_R-P_R	Y_S-P_S	Y_T-P_T	Y_U-P_U	Y_V-P_V	Y_W-P_W	Y_X-P_X	Y_Y-P_Y	Y_Z-P_Z
1	T	-0	-0	-0	-0	-0	-0.01	-0	-0	-0.13	-0.01	-0	-0	-0	-0	-0	-0	-0	-0	-0.01	0.21	-0.01	-0	-0	-0.01	-0.01	-0
2	I	-0.01	-0.01	-0	-0.01	-0.06	-0.02	-0.32	-0.01	0.88	-0.11	-0.01	-0.02	-0	-0	-0.03	-0	-0.16	-0.02	-0.05	-0	-0.01	-0.01	-0	-0	-0	-0.01
3	D	-0.01	-0.11	-0	0.89	-0.01	-0.02	-0.01	-0.14	-0.03	-0.12	-0.17	-0.01	-0	-0.01	-0.01	-0	-0.02	-0.05	-0.02	-0.03	-0.01	-0	-0	-0.11	-0	-0.01
4	N	-0	-0	-0.02	-0.01	-0.01	-0.01	-0.01	-0.03	-0	-0	-0.05	-0.02	-0.02	0.41	-0	-0	-0.04	-0.05	-0.01	-0.01	-0.05	-0.01	-0.03	-0.02	-0.01	-0
5	G	-0	-0.03	-0	-0.01	-0.03	-0	0.58	-0.01	-0	-0.05	-0.01	-0.01	-0	-0	-0.01	-0	-0.19	-0.13	-0.06	-0.01	-0.01	-0	-0	-0	-0	-0.01
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Things not covered

- How to choose proper learning rates. See [Friedman, 2001]
- Other possible loss functions. See [Friedman, 2001]

References I

Breiman, L. (1999).
Prediction games and arcing algorithms.
Neural computation, 11(7):1493–1517.

Breiman, L. et al. (1998).
Arcing classifier (with discussion and a rejoinder by the author).
The annals of statistics, 26(3):801–849.

Freund, Y. and Schapire, R. E. (1997).
A decision-theoretic generalization of on-line learning and an application to boosting.
Journal of computer and system sciences, 55(1):119–139.

Freund, Y., Schapire, R. E., et al. (1996).
Experiments with a new boosting algorithm.
In *ICML*, volume 96, pages 148–156.

References II

Friedman, J., Hastie, T., and Tibshirani, R. (2000).
Special invited paper. additive logistic regression: A statistical view of boosting.
Annals of statistics, pages 337–374.

Friedman, J. H. (2001).
Greedy function approximation: a gradient boosting machine.
Annals of Statistics, pages 1189–1232.

Schapire, R. E. and Freund, Y. (2012).
Boosting: Foundations and Algorithms.
MIT Press.