

# Bootstrap, Bagging, Random Forest & Adaboost

Shubham Chandel and Xintian Han

New York University

April 15, 2020

# What is random?

- statistic

# What is random?

- statistic is any function of the data. It is random since we're considering the data is random.

# What is random?

- parameter

# What is random?

- parameter is any function of the distribution. It is not random.

# What is random?

- point estimator

# What is random?

- point estimator is some statistic which is to serve as a “best guess” of an unknown parameter. It is random.

# What is random?

- sampling distribution



# What is random?

- sampling distribution is not random

# What is random?

- standard error

# What is random?

- standard error is a parameter of the sampling distribution. It is not random.

# What is random?

- bias and variance of a point estimator

# What is random?

- bias and variance of a point estimator are still parameters of the sampling distribution of the point estimator. It is not random.

# What is random?

- bootstrap sample

# What is random?

- bootstrap sample are random samples from the data. It is random.

Suppose we have samples  $X_1, \dots, X_N$  from  $\text{Normal}(\mu, \sigma^2)$ .

- What is the maximum likelihood estimator  $\hat{\mu}, \hat{\sigma}^2$  of the parameter  $\mu, \sigma^2$ ?
- Are the maximum likelihood estimator unbiased? If not, how do we fix it?
- What is the sampling distribution of  $\hat{\mu}$ ? What is the variance of the estimator  $\hat{\mu}$ ?



## Bias and Variance

Suppose we have samples  $X_1, \dots, X_N$  from  $\text{Normal}(\mu, \sigma^2)$ .

- What is the maximum likelihood estimator  $\hat{\mu}, \hat{\sigma}^2$  of the parameter  $\mu, \sigma^2$ ?

$$\hat{\mu} = \bar{X} = \frac{1}{N} \sum_{i=1}^N X_i; \quad \hat{\sigma}^2 = \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^2$$

- Are the maximum likelihood estimator unbiased? If not, how do we fix it?

$$\mathbb{E}\hat{\mu} = \mu; \quad \mathbb{E}\hat{\sigma}^2 = \frac{N-1}{N} \sigma^2$$

So  $\hat{\mu}$  is unbiased but  $\hat{\sigma}^2$  is not unbiased.  $\frac{1}{N-1} \sum_{i=1}^N (X_i - \bar{X})^2$  is an unbiased estimator of  $\sigma^2$ .

- What is the sampling distribution of  $\hat{\mu}$ ? What is the variance of the estimator  $\hat{\mu}$ ?  
Sampling distribution is  $\text{Normal}(\mu, \sigma^2/N)$ . Variance of the estimator is  $\sigma^2/N$ .

Let  $X_1, \dots, X_{2n+1}$  be an i.i.d. sample from a distribution. To estimate the median of the distribution, you can compute the sample median of the data.

- 1 How do we compute an estimate of the variance of the sample median?
- 2 How do we compute an estimate of a 95% confidence interval for the median.

# Bootstrap

Let  $X_1, \dots, X_{2n+1}$  be an i.i.d. sample from a distribution. To estimate the median of the distribution, you can compute the sample median of the data.

- ① How do we compute an estimate of the variance of the sample median?
  - ① Draw  $B$  bootstrap samples  $D^1, \dots, D^B$  each of size  $2n+1$ . The samples are formed by drawing uniformly with replacement from the original data set  $X_1, \dots, X_{2n+1}$ . We will make a total of  $B(2n+1)$  draws.
  - ② For each  $D^i$  compute the corresponding median  $\hat{m}_i$ .
  - ③ Compute the sample variance of the  $B$  medians  $m_1, \dots, m_B$ .
- ② How do we compute an estimate of a 95% confidence interval for the median.
  - ① Draw  $B$  bootstrap samples  $D^1, \dots, D^B$  each of size  $2n+1$ . The samples are formed by drawing uniformly with replacement from the original data set  $X_1, \dots, X_{2n+1}$ . We will make a total of  $B(2n+1)$  draws.
  - ② For each  $D^i$  compute the corresponding median  $\hat{m}_i$ .
  - ③ Compute the 2.5% and 97.5% sample quantiles of the list  $\hat{m}_1, \dots, \hat{m}_B$ . Use these as the estimates of the left and right endpoints of the confidence interval, respectively.

# Adaboost Algorithm

Given training set  $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ .

- ① Initialize observation weights  $w_i = 1, i = 1, 2, \dots, n$ .
- ② For  $m = 1$  to  $M$ :
  - ① Base learner fits weighted training data and returns  $G_m(x)$
  - ② Compute **weighted empirical 0-1 risk**:

$$\text{err}_m = \frac{1}{W} \sum_{i=1}^n w_i 1(y_i \neq G_m(x_i)) \quad \text{where } W = \sum_{i=1}^n w_i.$$

- ③ Compute  $\alpha_m = \ln\left(\frac{1-\text{err}_m}{\text{err}_m}\right)$  [**classifier weight**]
  - ④ Set  $w_i \leftarrow w_i \cdot \exp[\alpha_m 1(y_i \neq G_m(x_i))]$ ,  $i = 1, 2, \dots, n$  [**example weight adjustment**]
- ③ Output  $G(x) = \text{sign}\left[\sum_{m=1}^M \alpha_m G_m(x)\right]$ .

# Adaboost Algorithm

Decide whether each of the statements below is true or false. <sup>1</sup>

- If a weak classifier has a weighted error rate  $\epsilon \leq 1/3$ , it can only misclassify up to  $1/3$  of the training points.
- The error rate of the ensemble classifier never increases from one round to the next.
- Adaboost accounts for outliers by lowering the weights of training points that are repeatedly misclassified.
- When you update weights, the training point with the smallest weight in the previous round will always increase in weight.

---

<sup>1</sup>From MIT exams

# Adaboost Algorithm

Decide whether each of the statements below is true or false.

- If a weak classifier has a weighted error rate  $\epsilon \leq 1/3$ , it can only misclassify up to  $1/3$  of the training points.

**False.** We use weighted error rate.

- The error rate of the ensemble classifier never increases from one round to the next.

**False.** There is no guarantee for error rate not to increase from one round to the next.

- Adaboost accounts for outliers by lowering the weights of training points that are repeatedly misclassified.

**False.** Adaboost will increase the weights of training points that are repeatedly misclassified.

- When you update weights, the training point with the smallest weight in the previous round will always increase in weight.

**False.** It will increase weight only if it is misclassified in the current round.

# Adaboost Algorithm

Consider building an ensemble of decision stumps  $G_m$  with the AdaBoost algorithm,<sup>2</sup>

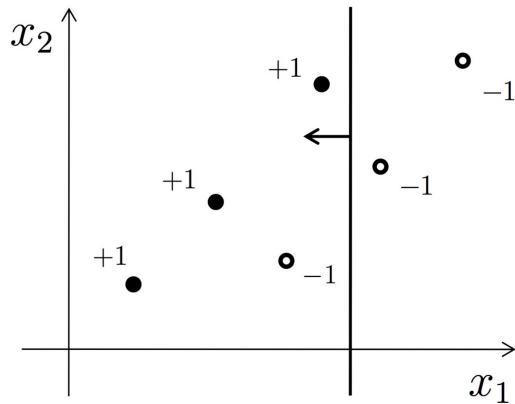
$$f(x) = \text{sign} \left( \sum_{m=1}^M \alpha_m G_m(x) \right)$$

Figure displays a few labeled point in two dimensions as well as the first stump we have chosen. A stump predicts binary  $\pm 1$  values, and depends only on one coordinate value (the split point). The little arrow in the figure is the normal to the stump decision boundary indicating the positive side where the stump predicts  $+1$ . All the points start with uniform weights.

- Circle all the point(s) in Figure 1 whose weight will increase as a result of incorporating the first stump (the weight update due to the first stump).
- Draw in the same figure a possible stump that we could select at the next boosting iteration. You need to draw both the decision boundary and its positive orientation.
- Will the second stump receive higher coefficient in the ensemble than the first? In other words, will  $\alpha_2 > \alpha_1$ ? Briefly explain your answer. (no calculation should be necessary).

<sup>2</sup>From CMU exams

# Adaboost Algorithm





# Adaboost Algorithm

- Circle all the point(s) in Figure 1 whose weight will increase as a result of incorporating the first stump (the weight update due to the first stump).  
(sol.) The only misclassified negative sample.
- Draw in the same figure a possible stump that we could select at the next boosting iteration. You need to draw both the decision boundary and its positive orientation.  
(sol.) The second stump will also be a vertical split between the second positive sample (from left to right) and the misclassified negative sample, as drawn in the figure.
- Will the second stump receive higher coefficient in the ensemble than the first? In other words, will  $\alpha_2 > \alpha_1$ ? Briefly explain your answer. (no calculation should be necessary).  
(sol.)  $\alpha_2 > \alpha_1$  because the point that the second stump misclassifies will have a smaller relative weight since it is classified correctly by the first stump.