Mathematical techniques in data science

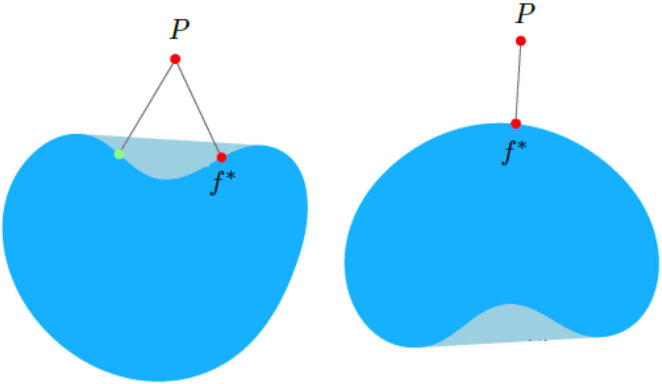
Lecture 31: Bootstrapping, bagging, random forests

Boosting

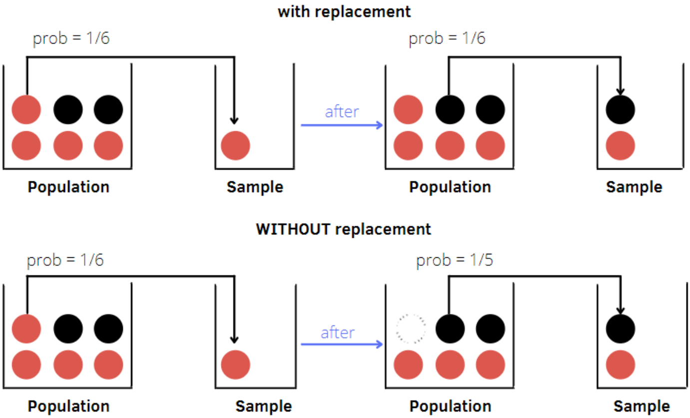
- Decision trees
- Bagging
- Random forests
- Boosting

Main idea: we can combine weak learners into a single strong learner

Convexification of the hypothesis space



Sampling with replacement



Bootstrap

Bootstrapping: General statistical method that relies on resampling data with replacement.

Idea: Given data (y_i, x_i), $i = 1, \ldots, n$, construct *bootstrap samples* by sampling n of the observations **with replacement** (i.e., allow repetitions):

$$\begin{array}{ccc}
\text{Sample 1} & \text{Sample 2} & \text{Sample 3} \\
(y_{i_1}, x_{i_1}) & (y_{j_1}, x_{j_1}) & (y_{k_1}, x_{k_1}) \\
(y_{i_2}, x_{i_2}) & (y_{j_2}, x_{j_2}) & (y_{k_2}, x_{k_2}) \\
\vdots & \vdots & \vdots \\
(y_{i_n}, x_{i_n}) & (y_{j_n}, x_{j_n}) & (y_{k_n}, x_{k_n})
\end{array}$$

Bagging

Bagging:(bootstrap aggregation) Suppose we have a model $y \approx \hat{f}(x)$ for data $(y_i, x_i) \in \mathbb{R}^{p+1}$.

1. Construct $B \in \mathbb{N}$ bootstrap samples.
2. Train the method on the b-th bootstrap sample to get $\hat{f}^{*b}(x)$.
3. Compute the average of the estimators:

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{i=1}^{B} \hat{f}^{*b}(x).$$

- Bagging is often used with regression trees.
- Can improve estimators significantly.

Bagging

Note: Each bootstrap tree will typically involve different features than the original, and might have a different number of terminal nodes.

The bagged estimate is the average prediction at x from these B trees.

For classification: Use a majority vote from the B trees.

Random forests

- Idea of bagging: average many noisy but approximately unbiased models, and hence reduce the variance.
- However, the bootstrap trees are generally correlated.
- Random forests improve the variance reduction of bagging by reducing the correlation between the trees.
- Achieved in the tree-growing process through random selection of the input variables.
- Popular method.

Random forests

Random forests: Each time a split in a tree is considered, a random selection of m predictors is chosen as split candidates from the full set of p predictors.

- Typical value for m is $\sqrt{p}$.
- We construct $T_1, \ldots, T_B$ trees using that method on bootstrap samples. The **random forest (regression) predictor** is

$$\hat{f}^B_{\text{rf}}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x).$$

For classification: use majority vote.

Advantages

- accurate and robust
- difficult to interpret compared to a decision tree
- does not suffer from the overfitting problem
- usually have built-in relative feature importance

Disadvantages

- slow in generating predictions because it has multiple decision trees
- difficult to interpret compared to a decision tree

Group work: Random forests

- use the Boston housing dataset
- fit a decision tree
- fit a random forest
- investigate feature importance