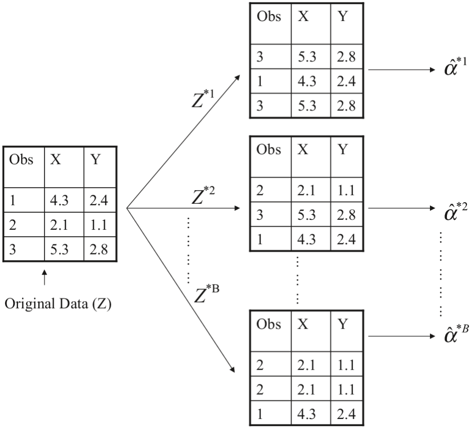
Bagging, Random Forests, Extra Trees

Bagging

Bootstrap

Figure: Bootstrap process[1]



[1] James et al. (2013)

Bootstrap

Bootstrap: Sampling B samples of size n with replacement from original data set

Applications

- Estimate the variability of model parameters
  - e.g. standard errors of regression coefficients
- Estimate test error with training data
  - Fit model on bootstrap samples and predict original training set
  - ".632" & ".632$^+$" estimator
- Construct an ensemble of learners for prediction
  - **Bagging**: Bootstrap Aggregating
  - Train prediction models on bootstrap samples

Bootstrap Aggregating

Bagging process

1. Draw B bootstrap samples from the training data
2. Build a prediction model $\hat{f}^{*b}(x)$ for each sample using a base learner
3. Compute the combined prediction $\hat{f}_{bag}(x)$ over all samples
    - Regression: $\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x)$
    - Classification: $\hat{G}_{bag}(x) = \arg\max_k \hat{f}_{bag}(x)$

$\rightarrow$ Averaging over multiple predictions reduces variance / increases prediction accuracy

Bootstrap Aggregating

Observations in each bootstrap sample

$$
\begin{aligned}
P(\text{obs } i \in \text{ sample } b) &= 1 - \left(1 - \frac{1}{n}\right)^n \\
&\approx 1 - e^{-1} \\
&= 0.632
\end{aligned}
$$

Out-of-bag (OOB) Error

- Sampling with replacement leads to models based on subsets of the data
- Unused (OOB) observations can be used for test error estimation
    1. Generate predictions for case i using models where i was OOB
    2. Average predictions for i and estimate test error
    3. Compute OOB error over all cases

Bagging Trees

Algorithm 1: Bagging Trees

```
Set number of trees B;
Define stopping criteria;
for b = 1 to B do
    draw a bootstrap sample from the training data;
    assign sampled data to root node;
    if stopping criterion is reached then
        end splitting;
    else
        find the optimal split point among the predictor space;
        split node into two subnodes at this split point;
        for each node of the current tree do
            continue tree growing process;
        end
    end
end
```

Bagging Trees

Figure: Bagging Trees

(a) b = 1 (b) b = 2 (c) b = 3

Bagging Trees

General motivation: Assume training observations (x_i, y_i) from a distribution $\mathcal{P}$ and bootstrap data x_i^*, y_i^* sampled from $\mathcal{P}$

$$
\begin{aligned}
\mathrm{E}_{\mathcal{P}}[Y - \hat{f}^*(x)]^2 &= \mathrm{E}_{\mathcal{P}}[Y - \mathrm{E}_{\mathcal{P}}\hat{f}^*(x) + \mathrm{E}_{\mathcal{P}}\hat{f}^*(x) - \hat{f}^*(x)]^2 \\
&= \mathrm{E}_{\mathcal{P}}[Y - \mathrm{E}_{\mathcal{P}}\hat{f}^*(x)]^2 + \mathrm{E}_{\mathcal{P}}[\hat{f}^*(x) - \mathrm{E}_{\mathcal{P}}\hat{f}^*(x)]^2 \\
&\geq \mathrm{E}_{\mathcal{P}}[Y - \mathrm{E}_{\mathcal{P}}\hat{f}^*(x)]^2
\end{aligned}
$$

$\rightarrow$ Suggests that Bagging decreases mean-squared error

bNN

Bagged nearest neighbors

1. For $b = 1$ to B do
   1. Draw a bootstrap sample b from the training data
   2. Identify the K nearest neighbors of test example
   3. Estimate $\hat{f}^{*b}(x)$ (regression: $\frac{1}{K}\sum y_i$, classification: $\frac{1}{K}\sum I(y_i = j)$)
2. Average all $\hat{f}^{*b}(x)$ to obtain $\hat{f}_{bag}(x)$

kNN vs. bNN

- kNN not much affected by bagging
- Resample sizes (Hall & Samworth 2005)
  - (m out of n) Bootstrapping: Improvement if resample size $< 0.69n$
  - Subsampling (w/o replacement): Improvement if resample size $< 0.5n$