BAGGING → RANDOM FORESTS ; BOOSTING

13/4/21

BAGGING (Bootstrap Aggregation)

Bootstrap is a general resampling technique, used in statistics to measure uncertainty (eg to measure standard errors of estimates).

IDEA: Assume that we have n independent observations $z_1, \dots, z_n$, all with the same variance σ^2. Then:

$$\mathrm{Var}(\bar{z}) = \mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} z_i\right) = \frac{1}{n^2}\mathrm{Var}\left(\sum_{i=1}^{n} z_i\right) \overset{z_i \text{ independent}}{=} \frac{1}{n^2}\sum_{i=1}^{n}\sigma^2 = \frac{\sigma^2}{n} < \sigma^2$$

Averaging observations reduces variance.

How could we make it work in our context?

Consider B independent training sets. We can then build B models $\hat{f}_1, \dots, \hat{f}_B$ and average them to reduce the variance:

$$\hat{f}_{avg}(x) = \frac{1}{B} \sum_{i=1}^{B} \hat{f}_i(x)$$

(x: new observation)

But we don't have B independent training sets!

→ This is where bootstrap comes in!

BOOTSTRAP Build B training sets from the single dataset that I have.

How is it done: repeatedly sampling with replacement from the dataset n times.

B bootstrapped samples $\longrightarrow$ B models $\hat{f}_1^*, \dots, \hat{f}_B^*$

$\longrightarrow$ average prediction

$$\hat{f}_{avg}(x) = \frac{1}{B} \sum_{i=1}^{B} \hat{f}_i^*(x)$$

Some remarks:

1. For classification, $\hat{f}_i^*(x)$ can be taken both as $\hat{p}_i^*(c|x)$ or you consider the class predicted by each tree (under a specific threshold) and $\hat{f}_{avg}(x)$ defined as majority vote.

(2) The B bootstrapped samples are not independent, they'll generally be correlated so we will not have a full $\frac{1}{B}$ reduction in variance.

(3) Bagging has no effect on bias, but it does reduce variance (unusually)

(4) Individual trees are grown deep and not pruned

(5) B should be taken as large as possible (no issues with overfitting)

OUT-OF-BAG (OOB) ERROR ESTIMATION

Test prediction accuracy on (unseen) test data.

- We could use CV. However, with bagging there is an alternative.
- Let's go back to resampling. Pick one observation x_i in the training data.

What is the probability that x_i is not included in a particular bootstrapped sample:

$$\left(\frac{n-1}{n}\right)^n$$

$$\lim_{n \to \infty} \left(\left(\frac{n-1}{n}\right)^n\right) = e^{-1} \simeq 0.368$$

So each bootstrapped sample uses approximately 63% of observations.

Idea : Use the remaining 37% observations for testing.

For any observation x_i, predict response using all trees for which it is OOB observation. So you get about $\frac{1}{3}B$ predictions for x_i and we average these predictions. Do this will all n observations and calculate the error (OOB error).

Two disadvantages of bagging:

① Loss of readability: average of trees is not a tree.

VARIABLE IMPORTANCE PLOT

For each predictor: add up the reduction in RSS/Gini/Entropy/... by splits on that predictor, averaged across all B trees.

② Bootstrapped samples are correlated.

Under independence: $\text{Var}(\bar{z}) = \frac{\sigma^2}{n}$

$z_1, \dots, z_n$ have same variance σ^2 and equal correlation $\rho > 0$.

$$\text{Var}\left(\sum_{i=1}^{n} z_i\right) = \sum_{i,j} \text{Cov}(z_i, z_j) = \sum_{i=1}^{n} \text{Var}(z_i) + 2 \sum_{i<j} \text{Cov}(z_i, z_j)$$

$$= n\sigma^2 + n(n-1)\rho\sigma^2$$

$$\Rightarrow \text{Var}(\bar{z}) = \frac{\sigma^2}{n} + \frac{n-1}{n}\rho\sigma^2 = \rho\sigma^2 + \frac{1-\rho}{n}\sigma^2 \xrightarrow{n \to \infty} 0$$

$\rho = \text{Cor}(z_i, z_j) = \frac{\text{Cov}(z_i, z_j)}{\sigma^2}$

RANDOM FORESTS

Question: Why would trees grown under bagging be correlated?

Answer If there is a small number of strong predictors, then almost all trees will have these at first splits and end up very similar to each other.

Idea Before each split in each tree, draw m features out of the p predictors and look at splits of these m features only

Remark If there is a strong predictor, only $\frac{m}{p}$ trees on average will have at the first split.

CHOICE OF m

Two tuning parameters : ① minimal node size (usually 2 or 5)
② m

Some choices:

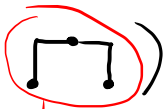
① $m = p$ → bagging

② Default choices:
Classification: $m = \lfloor \sqrt{p} \rfloor$
Regression: $m = \max\{\lfloor \frac{p}{3} \rfloor, 1\}$

③ Best procedure (if computationally feasible) is to tune this parameter by cross-validation, as it is connected to bias/variance trade-off.
Let us say that we have K strong predictors.
- The larger K is, the smaller m needs to be to reduce correlation
- The lower K is and/or many irrelevant features, the higher m should be

BOOSTING

Boosting is based on the idea of aggregating trees that are sequentially grown.

Idea: Use weak learners (simple, biased models eg a tree with a single stump ) and update these "slowly" by fitting a decision tree to the "residuals" of the previous model.

$J_m = 2$

GRADIENT BOOSTING ALGORITHM (regression trees)

Let $L(y_i, \hat{f}(x_i)) = (y_i - \hat{f}(x_i))^2$ denote the loss function.

① Initialix $f_0(x) = \underset{\gamma}{\text{argmin}} \sum_{i=1}^{n} (y_i - \gamma)^2$, ie $f_0(x) = \bar{y}$

② For $m = 1, \ldots, M$:

(a) Calculate $r_{im} = y_i - f_{m-1}(x_i)$, $i = 1, \ldots, n$ (residuals)

N.B. $r_i = -\frac{1}{2} \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}$ $\quad \frac{d}{d\gamma}(y_i - \gamma)^2 = -2(y_i - \gamma)$

(b) Fit a regression tree to the residuals r_{im}, giving terminal regions R_{jm}, $j = 1, \ldots, J_m$

(c) For $j = 1, \ldots, J_m$, compute

$$\gamma_{jm} = \underset{\gamma}{\text{arg min}} \sum_{x_i \in R_{jm}} L(y_i - f_{m-1}(x_i), \gamma),$$

ie γ_{jm} are the average residuals in region R_{jm}

(d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{m} \gamma_{jm} 1(x \in R_{jm})$

③ Output : $\hat{f}(x) = \hat{f}_M(x)$

Remarks

① M gives the number of trees fitted. Unlike bagging, here it is important.
If M is too large, it leads to overfitting.
M can be chosen by cross-validation

② Addition of a learning rate parameter in step (d):

$$f_m(x) = f_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$$

$0 < \nu < 1$

ν and M work against each other. A good strategy is to choose ν small (< 0.1), and let M grow until some optimal value is reached.

③ Fix $J_m = J$ for all trees. Find J by cross-validation. Usually $J = 2$ or $J = 3$ work quite well.

④ Compared to bagging, unimportant predictors may never show up. So boosting may also offer some variable selection

⑤ Extension to stochastic gradient boosting algorithm, that includes bagging in the procedure.
At each step m, the tree is grown on a subsample of the data → This gives a OOB error to evaluate the improvement of the m^{th} update (choice of M).