Bagging and Random Forests

MATH-412 - Statistical Machine Learning

Ensembling or aggregation

- given predictors $\hat{f}_1, \ldots, \hat{f}_B$
- let $\boldsymbol{\omega} = (\omega_b)_{1 \leq b \leq B} \in \triangle_B$ with $\triangle_B = \big\{ \boldsymbol{u} \in \mathbb{R}_+^B \mid \sum_b u_b = 1 \big\}$
- if $a \mapsto \ell(a, y)$ is a convex loss function then so is $\mathcal{R}$.

In that case, by convexity

$$\mathcal{R}\Big(\sum_{b=1}^{B} \omega_b \hat{f}_b \Big) \leq \sum_{b=1}^{B} \omega_b \, \mathcal{R}\big(\hat{f}_b\big)$$

In particular, if $\bar{f} = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b$ then

$$\mathcal{R}(\bar{f}) \leq \frac{1}{B} \sum_{b=1}^{B} \mathcal{R}(\hat{f}_b).$$

Bagging

Bagging="Bootstrap aggregating"

The Bootstrap

Classical Bootstrap : sample n datapoints uniformly at random from the original dataset **with replacement**.

Bagging

Given a training set $D_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$

- For each b,
  - sample a bootstrap training set $D_n^{(b)}$
  - use a learning scheme $\mathscr{A}$ to learn a predictor $\hat{f}_b$
- Return the aggregate predictor $\bar{f} = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b$

Note that by construction the predictors $\hat{f}_b$ are identically distributed but not independent.

Note that $D_n^{(b)}$ is in fact a multi-set since the same training pair can appear multiple times.

Bagging for the square loss

For the square loss

$$\mathbb{E}\big[\big(\hat{f}(x) - f^*(x)\big)^2\big] = \mathsf{Var}(\hat{f}(x)) + \big(\mathbb{E}[\hat{f}(x)] - f^*(x)\big)^2$$

For bagging, with $\bar{f} = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_b$, we have

- $\mathbb{E}[\bar{f}(x)] = \mathbb{E}[\hat{f}_b(x)]$ so $\mathsf{Bias}(\bar{f}(x)) = \mathsf{Bias}(\hat{f}_b(x))$.
- but if $\sigma_x^2 = \mathsf{Var}(\hat{f}_b(x))$ and $\rho_x = \mathrm{corr}(\hat{f}_b(x), \hat{f}_{b'}(x))$ with $b \neq b'$ then

$$\mathsf{Var}(\bar{f}(x)) = \Big(\rho_x + (1-\rho_x)\tfrac{1}{B}\Big)\sigma_x^2 \leq \sigma_x^2 = \mathsf{Var}(\hat{f}_b(x)).$$

$\rightarrow$ So if ρ_x is very small this can be advantageous.

Note however that if $\hat{f}$ is the predictor learnt from D_n, then in general we would expect intuitively that $\mathsf{Var}(\hat{f}_b(x)) \geq \mathsf{Var}(\hat{f}(x))$ because $\hat{f}_b$ is based only on a subset of the data. For the same reason, we will have $\mathsf{Bias}(\hat{f}_b(x)) = \mathsf{Bias}(\bar{f}(x)) \geq \mathsf{Bias}(\hat{f}(x))$. So ρ_x needs to be sufficiently small for bagging to be worth it.

Bagging of different predictors

Bagging is not very useful for some predictors

- E.g., not at all for the estimation of the mean $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$.
- It is usually considered not so useful for linear smoothers,

$\rightarrow$ although it can provide confidence bands...

- But it is useful for high variance low bias predictors like *unpruned trees*

Algorithm for bagging of trees :

- Learn an unpruned tree predictor $\hat{f}_b$ from each bootstrap training set
- Compute $\bar{f}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}_b(x)$
- For binary classification, and if a binary prediction is needed, use the

$$\textit{majority vote :} \quad 1_{\{\bar{f}(x)\geq\frac{1}{2}\}}.$$

Random forests (Breiman, 2001)

Key idea : Use bagging but decrease correlations further by randomizing the tree construction.

Algorithm :

- Bootstrap the data B times
- Change tree growth algorithm : before each split, randomly select m of the p variables. The subset changes at each split !
- Stop when leaves contain $\leq k$ points (e.g. $k = 5$ for regression, $k = 1$ for classification)
- Compute the final prediction by aggregation

Choice of m : $m \approx \sqrt{p}$ for classification and $m \approx p/3$ for regression seem to work well.

Intuition : by forcing the trees to split on random variables, we are decreasing the probability that they choose the same splits and we are decreasing correlation.

Out-of-bag (OOB) cross-validation

- Each bootstrap sample contains approximately a fraction $1 - \frac{1}{e} \approx 63\%$ of the data.
- As in CV, we can use the remaining 37% as a validation set to estimate the risk of $\hat{f}_b$

$$\widehat{\mathcal{R}}_{V_b}(\hat{f}_b) = \frac{1}{|V_b|} \sum_{i \in V_b} \ell(\hat{f}_b(x_i), y_i) \qquad \text{with} \qquad V_b = D_n \backslash D_n^{(b)}.$$

- The **average out-of-bag (OOB) risk estimate** is defined as

$$\widehat{\mathcal{R}}_{\text{CVOOB}} = \frac{1}{B} \sum_{b=1}^{B} \widehat{\mathcal{R}}_{V_b}(\hat{f}_b).$$

- We have $\mathbb{E}[\widehat{\mathcal{R}}_{\text{CVOOB}}] = \mathbb{E}[\mathcal{R}(\hat{f}_b)] = \mathbb{E}[\frac{1}{B}\sum_{b=1}^{B} \mathcal{R}(\hat{f}_b)] \geq \mathbb{E}[\mathcal{R}(\bar{f})]$, provided $\mathcal{R}$ is convex.

$\widehat{\mathcal{R}}_{\text{CVOOB}}$ can overestimate $\mathcal{R}(\bar{f})$ but the quadratic risk is typically dominated by the bias; in general, this overestimation does not affect hyperparameter selection (Janitza and Hornung, 2018)

Note that the cost of computing each $\widehat{\mathcal{R}}_{V_b}(\hat{f}_b)$ is very small compared to learning $\hat{f}_b$.

Variable Importance

In order to determine which variables are important in a random forest, several measures of variable importance have been proposed.

Gini VI score : sum of all impurity decreases obtained by splitting according to that variable

- Sadly, often computed on in-bag samples $\rightarrow$ risk of overfitting
- Has a bias in favor continuous variables and discrete variables with many values.

Permutation VI score

$$\text{Imp}_k = \frac{1}{B}\sum_{b=1}^{B}\left[\widehat{\mathcal{R}}_{V_{b,k}}(\hat{f}_b) - \widehat{\mathcal{R}}_{V_b}(\hat{f}_b)\right]$$

where $V_{b,k}$ is the OOB sample in which the values of the kth variable have been permuted among the validation data : x_{ik} is replaced by $x_{\sigma(i)k}$ with σ a permutation of the indices of the data in V_b.

- Tends to assign too much importance to positively correlated variables.
- Creates impossible datapoints by breaking the correlation structure between variables.

Conditional permutation VI score (not commonly used, but available in `partykit` in R)

- Permutes within a grid defined by the splits associated with the most correlated variables to the variable of interest. This might constrain too much permutations.

Final remarks on Random Forests

- It has been claimed that RF do not overfit :
  - There is obviously no overfitting by increasing B since all trees are learned independently.
  - There can be some overfitting due to lack of pruning but this is often moderate.
  - RF can perform poorly if m is too small ($\hat{f}_b$ is too randomized).
- A random forest associates to each data point a feature vector $\phi(x) = \left(1_{\{x \in R_j^{(b)}\}}\right)_{j,b}$
- There are strategies to accommodate missing data.
- There are other algorithms for random forest including algorithms based on honest trees in which the training data is split in two sets, one to construct the tree and the other to obtain unbiased estimates of the leafs values. See e.g. Biau and Scornet (2016); Athey et al. (2019).
- A strong competitor of RF are boosted trees.

References I

Athey, S., Tibshirani, J., Wager, S., et al. (2019). Generalized random forests. *The Annals of Statistics*, 47(2) :1148–1178.

Biau, G. and Scornet, E. (2016). A random forest guided tour. *Test*, 25(2) :197–227.

Janitza, S. and Hornung, R. (2018). On the overestimation of random forest's out-of-bag error. *PloS one*, 13(8) :e0201904.