EPFL, Autumn 2024
Yoav Zemel
Guillaume Obozinski

Ho Yun
Shivang Sachar

Statistical Machine Learning

Exercise sheet 12

Practical Exercise

Exercise 12.1 (Trees, bagging and random forest) In this exercise you will work on fitting tree-based methods in R. We will be working on the two datasets: `Hitters` is a data set that contains 20 statistics on 322 players from the 1986 and 1987 seasons (regression problem), while `Heart` contains 14 heart health-related characteristics on 303 patients (classification problem).

(a) Load the two datasets with the help of the R template given on moodle. Perform a random 70/30 training/validation split of the dataset.

(b) Use the `Hitters` dataset from the `ISLR` package, the `rpart()` and the `prp` plot command to demonstrate a regression tree that fits a continuous response: the log salary of each player based on number of years in the league and number of hits the previous season. Specify for your tree to have 2 levels of branches. What log salary does the tree predict for a new hitter who had been in the league for 5 seasons and had 110 hits the previous season?

(c) If beforehand, we do not want to specify how many levels of branches to have for our decision tree, what could we do? Use `rpart` to cross-validate the results of the tree and trim the tree (i.e. pruning). Use `rpart.plot` to visualise the tree.

(d) Use the `predict` function to estimate the log salary for every hitter in the testing set as well as calculate the total error.

(e) Now, with the `Heart` dataset, we attempt to predict whether a patient has heart disease (AHD) based on some health-related characteristics. After removing the identification number, fit a decision tree.

(f) Calculate the confusion matrix for this predictor.

(g) Going back to the `Hitters` dataset, use the complete.cases function to select only the 184 observations in the training set that have a complete set of variables, and fit a bagged tree predictor by using the `randomForest` command from the `randomForest` package by setting the `mtry` equal to the number of predictors.

(h) Calculate the sum of squared errors the same way we did for trees, and compare your error to part (d). Which method is better?

(i) Repeat (g) and (h) for the `Heart` dataset.

(j) Repeat (g), (h) and (i) but with a random forest predictor.

(k) The 'out-of-bag' error measure can be used to verify that we have planted enough trees. Random forests do not begin to overfit as the number of trees increases; it just has to be verified that the error has stabilized. By using the `plot` command, we can plot OOB error as a function of the number of trees. Have you used enough trees for the OOB error to have stabilized?

(l) Which variables were the most important in calculating the estimates. Call the `importance` and `varImpPlot` functions.

Exercise 12.2 (Partial Dependence Function) The partial dependence function is a tool used in machine learning to understand the relationship between a predictor variable (or a specified subset of predictor variables) and the target variable while averaging out the effects of other predictors. It provides insights into how a specific feature influences the predictions made by a model, such as a Random Forest.

The partial dependence function represents the average predicted response as a function of one or more predictor variables, while marginalizing over the distribution of other predictors in the dataset. Consider the subvector $\boldsymbol{X}_{\mathcal{S}}$ of $l < p$ of the input variables $\boldsymbol{X}^T = (X_1, X_2, \ldots, X_p)$, indexed by $\mathcal{S} \subset \{1, 2, \ldots, p\}$. Let $\mathcal{C}$ be the complement set of $\mathcal{S}$. The partial dependence function for a general function $f(X)$ is defined as

$$f_{\mathcal{S}}(\boldsymbol{X}) = \mathrm{E}_{\boldsymbol{X}_{\mathcal{C}}} f(\boldsymbol{X}_{\mathcal{S}}, \boldsymbol{X}_{\mathcal{C}}).$$

It can be estimated by the estimator

$$\overline{f}_{\mathcal{S}}(\boldsymbol{X}) = \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{X}_{\mathcal{S}}, \boldsymbol{x}_{i\mathcal{C}}),$$

where $\{\boldsymbol{x}_{1\mathcal{C}}, \ldots, \boldsymbol{x}_{n\mathcal{C}}\}$ are the values of $\boldsymbol{X}_{\mathcal{C}}$ occuring in the training data.

(a) What does the term *marginalizing over the distribution of other predictors* mean in the context of the partial dependence function?

(b) Use the `partialPlot()` function from the `randomForest` package to plot the partial dependence of `Hits` on `Salary` for the `Hitters` dataset in the previous exercise. What trend do you observe in the partial dependence plot for `Hits`?