目录

插图

机器学习面试笔记- 随机森林算法

ShenHengheng

本章主要讲解随机森林算法, 主要有以下部分组成: 随机森林算法的定义, 随机森林算法的几个要素: 包括变量重要性的衡量, OOB(out-of-bags) 和接近程度的衡量等, 最后会讲解有关随机森林算法的过拟合问题.

1 简介

在讲解之前, 需要梳理以下知识点:(1) 自动聚合算法 (Bagging, Boostrap Aggregation) (2) 提升算法 (Boosting) (3) 随机森林.

首先 Bagging 算法是机器学习中一种常用的降低预测方差、提高稳定性的方法。尤其对于那些高方差, 低偏差的算法有很好地效果, 比如常见的 Tree Bagging 就是将多个决策树的结果平均后的模型来改善决策树的高方差问题. 算法步骤如下:

1、从训练样集中有放回的选取 n 个样本，然后把这 n 个样本作为训练集并训练模型。(bootstrap 的过程)

2、把第 1 步重复 m 次，从而得到 m 个不同的预测模型。

3、利用这 m 个模型对测试集进行预测，将 m 个预测结果取平均值。(aggregating 的过程)

(Boosting) 提升算法不像 bagging 算法, 提升算法中的弱分类器在迭代的过程中不断地变强, 主要通过给所有的弱分类器不同的权重来控制强弱. 比如 AdaBoosting, GBM(Gradient Boosting Method) 算法. Boosting 算法的实验效果要比 Bagging 算法要好, 因此目前 Boosting 正在逐渐地取代 Bagging 算法, 并且 Boosting 算法成为很多任务的首选算法.

随机森林算法是基于 Bagging 的算法, 是由 Breiman 在 2001 年提出的. 随机森林算法构建了大量的去相关 (*de-correlated*) 的树 (可以看成弱分类器), 然后将每个树的结果取平均, 在大量的任务的预测中, 随机森林算法的结果和提升算法很接近, 并且随机森林算法简单且易于调试, 也正成为主流算法.

> *Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees. As in bagging, we build a number of decision trees on bootstrapped training samples.*

2 随机森林算法

因为 RF 是基于 Bagging 算法, 现在有必要了解 Bagging 算法的一些知识. 下面我会通过问答的形式进行梳理相关的基础知识, 并且深入了解 RF 在 Bagging 算法中动了哪些手脚.

问题 1: (Bagging 算法为什么能够降低预测方差?)

解答 1: Bagging 算法的核心思想是将有噪音但是接近无预测偏差的模型去平均, 这样下来就能够降低模型的预测方差.

问题 2: (为什么说基于树的算法是 Bagging 的 boostrap 模型的最佳选择?)

解答 2: 因为它们能够捕捉到数据中复杂关系, 并且树分裂的足够深, 这会使得偏差越来越小.

Algorithm 1 *Random Forest for Regression or Classification.*

1. For b = 1 to B:
   (a) Draw a bootstrap sample $Z^{\star}$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by re- cursively repeating the following steps for each terminal node of the tree, until the minimum node size n_min is reached.
      i Select m variables at random from the p variables.
      ii Pick the best variable/split-point among the m.
      iii Split the node into two daughter nodes.
2. Output the ensemble of trees T_{b1}^{B}

To make a prediction at a new point x:
Regression: $\hat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_B(x)$
Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{rf}^{B}(x) = majority\ vote\{\hat{C}_b(x)\}_1^B$.

由于 Bagging 算法生成的树是同分布的 (*i.d.*), 因此在 B 个树上求期望和在单个树上求期望是一样的. 那么只能通过降低 Bagging 中的方差来改进算法. 但是 Boosting 算法恰恰相反,Boosting 算法中的子模型是不断改变的, 因此它们不是同分布的.

An average of B i.i.d. random variables, each with variance σ^2, has variance $\frac{1}{B}\sigma^2$. If the variables are simply i.d. (identically distributed, but not necessarily independent) with positive pairwise correlation ρ, the variance of the average is

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2 \tag{1}$$

通过公式 (Equation 1) 可以发现随着 B 的增大, 第二项消失, 只保留了第一项, 因此 B **的大小会影响树之间的相关性**. 随机森林算法 (Algorithm 1) 的想法是通过减少树之间的相关性来使 Bagging 的方差减少, 而不会过多地增加方差。**这是通过在树的生成过程中随机选择输入变量分裂实现的**。

一般地, 在一个 Boostrap 生成的数据集中作如下操作 (通常 $m = \sqrt{p}$ 甚至选择 1):

Before each split, select $m \leq p$ of the input variables at random as candidates for splitting.

随着 B 个树 $\{T(x;\Theta_b)\}_1^B$ 生长完成, 那么随机森林的预测为:

$$\hat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T(x;\Theta_b) \tag{2}$$

Θ_b characterizes the bth random forest tree in terms of split variables, cutpoints at each node, and terminal-node values. Intuitively, reducing m will reduce the correlation between any pair of trees in the ensemble, and hence by (1) reduce the variance of the average.

2.1 距离度量

3 随机森林算法细节

- For classification, the default value for m is $\lfloor\sqrt{p}\rfloor$ and the minimum node size is one.
- For regression, the default value for m is $\lfloor p/3\rfloor$ and the minimum node size is five.

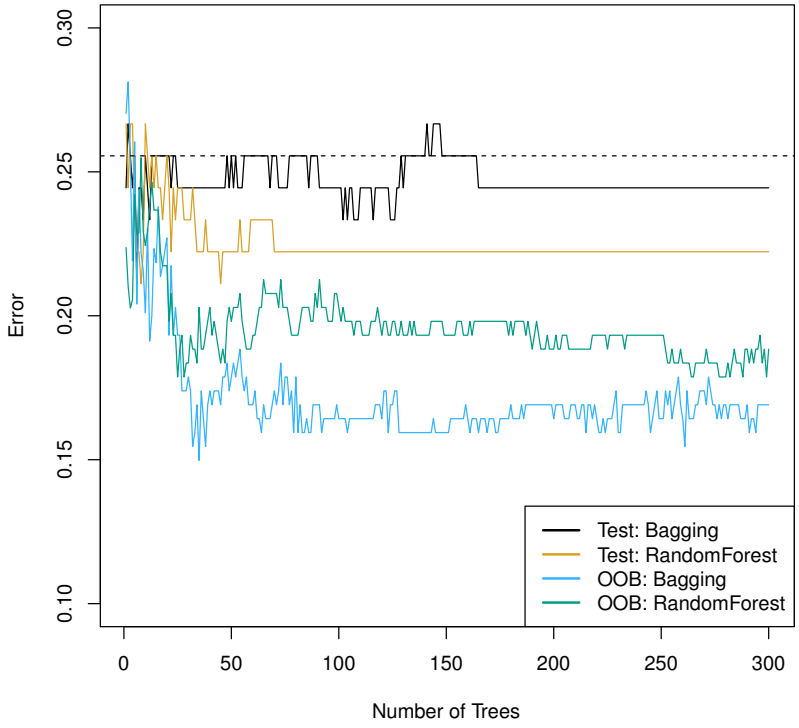

图 1: *Bagging, random forest, and gradient boosting, applied to the spam data. For boosting, 5-node trees were used, and the number of trees were chosen by 10-fold cross-validation (2500 trees). Each "step" in the figure corre- sponds to a change in a single misclassification (in a test set of 1536). N*

3.1 OOB

WHAT IS OOB:

For each observation $z_i = (x_i, y_i)$, construct its random forest predictor by averaging only those trees corresponding to boot- strap samples in which z_i did not appear.

After creating the classifiers ($\mathcal{S}$ trees), for each (X_i, y_i) in the original training set i.e. T, select all T_k which does not include (X_i, y_i). This subset, pay attention, is a set of boostrap datasets which does not contain a particular record from the original dataset. This set is called out-of-bag examples. There are n such subsets (one for each data record in original dataset T). **OOB** classifier is the aggregation of votes ONLY over T_k such that it does not contain (X_i, y_i).

Out-of-bag estimate for the generalization error is the error rate of the out-of-bag classifier on the training set (compare it with known y_i's).

Why is it important? The study of error estimates for bagged classifiers in Breiman [1996b], gives empirical evidence to show that the out-of-bag estimate is as accurate as using a test set of the same size as the training set. Therefore, using the out-of-bag error estimate removes the need for a set aside test set.

3.2 变量重要性

Variable importance plots can be constructed for random forests in exactly the same way as they were for gradient-boosted models.

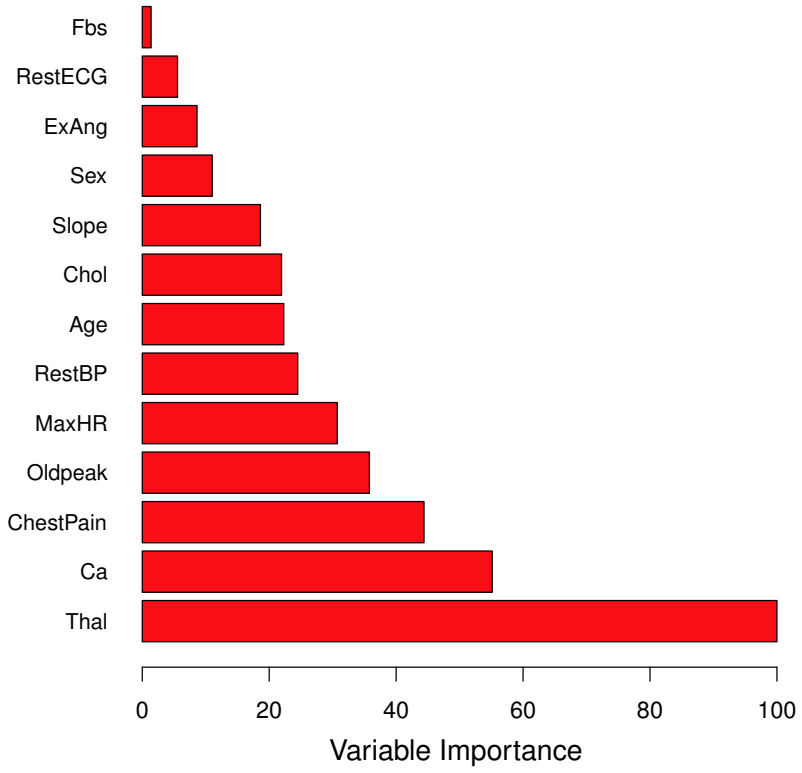

图 2: *Variable importance plots for a classfication random forest algorithm*

3.3 Proximity Plots

3.4 随机森林算法的过拟合问题

4 随机森林算法分析