Final_project_Bottino_Poetto_Spagliardi

Manuel Bottino, Patrick Poetto, Jacopo Spagliardi

2023-06-07

Contents

1 Review

1.0.1 Why it works

First of all let's introduce the general idea of bagging. Given a dataset $L = \{(x_n, y_n),\ n = 1, \dots, N\}$ we try to improve a predicting procedure in a "naive" way, ideally we would like to have a sequence of datasets $\{L_k\}$ of training sets each containing N independent observations, then take the average of the sequence $\{\phi(x, L_k)\}$.

$$\phi_A(x, P) = E_L[\phi(x, L)] \tag{1}$$

.

Because ϕ_A depends not only on x but also the underlying probability distribution P from which L is drawn. Since we are missing this last information the only tool we are left with is our dataset L, we instead use P_L the bootstrap approximation of P, it can be defined as the probability mass function with equal probability of extracting each element of the dataset (x_n, y_n). Finally we can define the bootstrap aggregate predictor as:

$$\phi_B(x) = \phi_A(x, P_L) \tag{2}$$

.

In order to understand why bagging works theoretically it can be proved that mean-squared error (MSE) of $\phi_A(x)$ is lower than the mean-squared error averaged over L of $\phi(x, L)$. How much lower the two sides are depends on the inequality:

$$[E_L\phi(x, L)]^2 \leq E_L\phi^2(x, L) \tag{3}$$

This result is true taking advantage of the Jensen's inequality for the specific case in which $g(X) = X^2$, this function is convex ($g'' > 0$), thus $E[Z^2] \geq (E[Z])^2$

1.0.2 Instability

library(bookdown) The inequality @ref(eq:ineq) is a nice starting point to explain what role instability plays. In fact, If $\phi(x, L)$ does not change too much with replicate L the two sides will be nearly equal, and aggregation will not help. The more highly variable the $\phi(x, L)$ are, the more improvement aggregation may produce. Applying this reasoning to our $\phi_B(x)$, if the procedure is unstable, it can give improvement through aggregation. However, if the procedure is stable, then $\phi_B(x) = \phi_A(x, P_L)$ will not be as accurate for data drawn from P as $\phi_A(X, P) \sim \phi(x, L)$.

Let's see how this concept can be translated when we look at a more specific context, like classification. In this instance the predictor $\phi(\boldsymbol{x}, L)$ predicts a label $j \in \{1, ..., J\}$. We first define $Q(j|x) = P(\phi(x, L) = j)$, over many independent replicates of the learning set L, ϕ predicts class label j at input x with relative frequency $Q(j|x)$, and let $P(j|x)$ be the probability that input x generates class j. At this point we can set $\phi^*(x) = \arg\max_j P(j|x)$ (the Bayes predictor) which leads to the following expression for $Q(j|x)$:

$$\begin{cases} 1 \; if \;\; P(j|x) = max_i P(i|x) \\ 0 \; elsewhere \end{cases}$$

Now we have all the ingredients to show the maximum classification rate where:

$$r^* = \int \max_j P(j|x) P_X(dx) \tag{4}$$

where $P_X(dx)$ is the probability distribution of X.

If we know focus on the aggregate of ϕ and define it following the procedure described above $\phi_A(x) = \arg\max_j Q(j|x)$. We have the maximum attainable correct-classification rate of $\phi(x)$, we are missing the corret-classification for x, which for $\phi_A(x)$ is $\sum_j I(\arg\max_j Q(j|x) = j)P(j|x)$ Where I is the indicator function. *Putting all the pieces together* the correct-classification rate for $\phi_A(x)$ is:

$$r_A = \int_{\boldsymbol{x} \in C} max_j P(j|\boldsymbol{x}) P_X(d\boldsymbol{x}) + \int_{\boldsymbol{x} \in C'} [\sum_j (I(\phi_A(\boldsymbol{x}) = j) P(j|\boldsymbol{x})] P_X(\boldsymbol{x}) \tag{5}$$

Even if ϕ is order correct at x its correct classification rate can be far from optimal, but ϕ_A is optimal. Only if ϕ is order-correct for most input of x (it is good), then aggregation can transform it into a nearly optimal predictor. On the other hand, unlike the numerical prediction situation, poor predictors can be transformed into worse ones if we use bagging.

1.0.3 Model - Decision Tree-Based Methods

Tree-based methods partition the feature space into a set of rectangles, and then fit a simple model, say a constant, in each one. This is a simple yet powerful procedure since it is able to disentangle a model into simpler and smaller models and describe his features more accurately than global models do basically using binary conditional clustering. The geometric perspective described before can be seen as a tree where data are run and at each node a test is conducted to see what is the path a covariate should follow until reaching a leaf, which represents the final prediction explained by the constant model. For example, let's say we have p inputs and a response, for each of N observations: that is, (x_i, y_i) for $i = 1, 2, ..., N$, with $x_i = (x_{i1}, x_{i2}, ..., x_{ip})$. The algorithm has decide on the splitting variables and split points, as well as what shape the tree should have. Suppose first that we have a partition into M regions $R_1, R_2, ..., R_M$, and we model the response as a constant c_m in each region: As a criterion for optimal partitional we can minimize the sum of squares $\sum(y_i - f(x_i))^2$. In this way the best $\hat{c}_m$ is just the average of y_i in region R_m: $\hat{c}_m = av(y_i|x_i \in R_m)$. Our model will be then: $f(x) = \sum_{m=1}^{M} c_m \cdot I(x \in R_m)$.

Now finding the best binary partition in terms of minimum sum of squares is generally computationally infeasible so we can set up a CART (classification and regression tree) algorithm starting with the data, a splitting variable j, a split point s, and defining the half planes as: $R_1(j,s) = \{X \mid X_j \leq s\}$ and $R_2(j,s) = \{X \mid X_j > s\}$. Then we seek j and s that solve

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i \in R_1(j,s)}(y_i - c_1)^2 + \min_{c_2}\sum_{x_i \in R_2(j,s)}(y_i - c_2)^2\right]$$

For any choice j and s, the inner minimization is solved by:

$$\hat{c}_1 = \mathrm{av}(y_i \,|\, x_i \in R_1(j,s)) \quad \text{and} \quad \hat{c}_2 = \mathrm{av}(y_i \,|\, x_i \in R_2(j,s))$$

For each j, the split point s can be found very quickly and hence determination of the best pair (j, s) is feasible by brute force. Having found the best split, we partition the data into the two resulting regions and repeat the splitting process on each of the two regions. Then this process is repeated on all of the resulting regions. The question now become: how large should we grow the three? It is pretty straightforward that too many nodes (splits) may overfit the data while doing vice versa may end up not being able to capture them well, resulting in misprediction. One strategy could be to set a lower threshold for the decrease of the sum of squares and stop the splitting when this is reached. However this strategy is too short sighted since a seemingly worthless split may lead to a very good one below. A more robust strategy may be to do kind of the opposite: grow a very large tree and then use a cost-complexity pruning criterion to collapse one internal node at a time from the full tree until the single node tree so that we find a sequence. It is intuitive that the optimal tree must be somewhere in the sequence. Now, with a cross validation selection method, we can find the actual optimal tree just minimizing the cross validated sum of squares.

This is how the CART algorithm for growing decision trees basically works. Note that decision trees are divided in classification and regression trees if the response is a factor or a numerical or continous variable. Below it is proposed a very basic implementation of a CART algorithm using mean squared error cost complexity criterion. The algorithm is not implemented to prune the tree.

visuals bagging application - model setting (paragrafo 5+ fig1)

2 Data Exploratory Analysis (DEA)

This data can be used to compare with cancer genes and label these data to predict and diagnose cancers such as breast cancer and also to diagnose the stage of different cancers. Also, from the expression of these DNAs, artificial intelligence models can be obtained that can predict future mutations in any disease, or find the mechanisms of protein production and their gene ontology, and treat, diagnose and predict the disease such as cancer.

3 References

knitr::write_bib()