Tree-based Methode (regression)

Bachir El Khadir

<2016-02-23 Tue>

Contents

1 Tree

Definition 1. $\mathcal{F} := \{f(x) = \sum_{j=1}^{M} \beta_j 1_{x \in R_j}\}$ *Where* $R_1, \ldots, R_M$ *for a* tree partition

Definition 2. *A parititon of the input space X that can be formed by recursively applying the following 2 rules*

- *Choose a cell of the current partition*
- *Split the chosen cell into two daughters by binary splitting along one dimension (one variable)*

Definition 3. $\hat{R}(f) = \frac{1}{n} \sum_{i=1}^{M} (Y_i - f(X_i))^2$ *minimizing this quantity leads to overfitting, so we need to regularize. For example we can restrict the search space to* $\mathcal{F}_{K_{min}} = \{f(x) = \sum_{j=1}^{M} \beta_j 1_{x \in R_j}\}$ *and* R_j *contains at least* K_m*in data points (e.g* $K_{\min} = 5$*)*

Computation: Conbinatoric! (NP-Hard) In practice we use a Greedy Algorithm.

Definition 4. *Grow a tree recursively by repeating the following steps: for each terminal node of the tree, until the minimal node size $K_{\min}$ is achieved*

1. *Pick a variable / split point which decreases $\hat{R}(f)$ the most*
2. *Split the node into two daughters*

Still overfits.

1.1 Prune The tree

1. Givena full grown tree T_0, find an internal node which after collapsing the subtree into iteself, will increase $\hat{R}(f)$ the least.
2. Collapse the subtree into this internal node. We get a new tree T, reoeat tihs process we get a sequence of new trees $T_0, T_1, \ldots$.
3. Pick one tree by minimizing $\hat{R}(\hat{f}_T) + \lambda|T|$, where λ is obtained by CV tuning, $|T|$ the number of nodes in T.

1.2 Pros and Cons of Tree

- Pro: Simple a interpretable
- Cond: Fitten functions are non smooth: theoritically extremely challenging (no persistensy result)

2 Bagging (Bootstrap Aggregation)

For $b = 1, \ldots, B$ a. Draw a boostrap ${Z^*_{1:n}}^{(b)}$ of size n from $Z_{1:n}$ b. Fit a regression tree on the bootstrapped data (with minimum node size $K_{\min}$, no pruning) Output: $\hat{f}^{\text{bagging}} = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^b(x)$

2.1 Bagging vs Tree

1. $\hat{f}^{\text{bagging}}$ has the same bias as $\hat{f}^b(x)$, but potentially smaller variance.
2. The larger B is, the better (but diminishing return)
3. Works well only if $\hat{f}^1(X), \ldots, \hat{f}^B(x)$ are decorrelated.

3 Random Forest

For $b = 1, \dots, B$

1. a. Draw a boostrap ${Z_{1:n}^*}^{(b)}$ of size n from $Z_{1:n}$

b. Fit a regression tree on the bootstrapped data by recursively repeating the following steps for each termainl node of the tree until the minimum node of size K_min is achieved

i) Select m variables at random ii) Picj the best variable / split point aming these m variables iii) Split the node into 2 daughters

1. Output: \$ Output the ensemble of fitted tree functions $\hat{f}^1, \dots, \hat{f}^B$.
   $\hat{f}^{\mathrm{RT}} = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^b(x)$