Lec 13

Tuesday, October 29, 2019 10:56

Recap: trees

tree model: $\hat{f}(x) = \sum_{\ell=1}^{L} \mathbb{I}[x \in R_\ell] \hat{\beta}_\ell$

$R_1, \dots, R_L$ are tree partition of feature space

Motivation: interpretable & handles high dims (like linear model) but also flexible (like kNN / kernel)

How to fit? first focus on regression

Consider $\hat{f}$'s regression error on training data (just like we did in OLS)

$$\sum_{i=1}^{n} (\hat{f}(x_i) - y_i)^2 = \sum_{i=1}^{n} \sum_{\ell=1}^{L} \mathbb{I}[x_i \in R_\ell] (\hat{\beta}_\ell - y_i)^2$$

$$= \sum_{\ell=1}^{L} \sum_{i: x_i \in R_\ell} (\hat{\beta}_\ell - y_i)^2$$

min over $\hat{\beta}_\ell$

$$\rightarrow \hat{\beta}_\ell = \frac{1}{|R_\ell|} \sum_{i: x_i \in R_\ell} y_i$$

$$|R_\ell| = \sum_i \mathbb{I}[x_i \in R_\ell]$$

So if we fix $R_1, \dots, R_L$ & opt $\hat{\beta}$

We get

$$\sum_{i=1}^{n} (\hat{f}(x_i) - y_i)^2 = \sum_{\ell=1}^{L} I_{SSE}(\{y_i : x_i \in R_\ell\})$$

$$I_{mse}(\{y_1, \dots, y_k\}) = \sum_{i=1}^{k} \left(y_i - \frac{1}{k} \sum_{i'=1}^{k} y_{i'}\right)^2$$

$= k \cdot \mathrm{Var}(\{y_1, \cdots, y_k\})$

Target for regression tree learning:

find a tree w/ leaf regions that have minimal sum of impurities ISSE.

Hard optimization problem.

So: instead solve greedily

Recursive Partitioning Algo

Input: $(X_1, Y_1), \cdots, (X_n, Y_n)$, impurity fn $I(\{\})$

Find the best $j = 1, \cdots, p$ & $k' = 1, \cdots, k-1$

$$S_{jk'}^{L/R} = \left\{ (X_i, Y_i) : \begin{array}{l} i = 1, \cdots, k \text{ s.t.} \\ X_{ij} \lessgtr \frac{X_{k'j} + X_{k'+1,j}}{2} \end{array} \right\}$$

$\lessgtr$: $L: <$, $R: >$

looking at sorted values of $X_{k'j}$

to minimize $I(S_{jk'}^L) + I(S_{jk'}^R)$

Then: recurse on $S_{jk'}^L$, $S_{jk'}^R$

or "stop"

For Classification: $(Y_i \in \{1, \cdots, m\})$

The gini impurity

$$I_{\text{gini}}(\{y_1, \cdots, y_k\}) = k \sum_{j=1}^m \hat{p}_j (1 - \hat{p}_j)$$

where $\hat{p}_j := \frac{1}{k} \sum_i \mathbb{1}[y_i = j]$

For $Y_i \in \{0,1\}$: $I_{Gini} = 2\hat{p}_1(1-\hat{p}_1)$

The entropy impurity

$$I_{entropy}(\{y_1,\dots,y_K\}) = -\sum_{j=1}^{m} \hat{p}_j \log(\hat{p}_j)$$

Motivation:

$$\sum_{l=1}^{L} I_{entropy}(\{Y_i : i \in R_l\}) = \min_{\beta_{0:L}} \left(\text{negative log lik of data under } \hat{f}\right)$$

When to stop partitioning?

- Can stop at a given max depth
- Can stop at a given min leaf size

How to choose these params? CV!

Other options: pruning

Ensembles

Warm up exercise for bagging

Suppose we had $\hat{f}_1, \dots, \hat{f}_B$ predictors that estimate $f^*(x) = \mathbb{E}[Y \mid X = x]$

$\forall b = 1,\dots,B$ $\quad |\mathbb{E}[\hat{f}_b(x) - f^*(x)]| \le$ "bias"

$\quad \mathrm{Var}\, \hat{f}_b(x) \le$ "variance"

$\quad \mathrm{Cov}(\hat{f}_b(x), \hat{f}_{b'}(x)) = 0 \quad \forall b \neq b'$

Consider $\hat{f}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(x)$

Then: $|\mathbb{E}[\hat{f}(x) - f^*(x)]|$

$= |\mathbb{E}[\frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(x) - f^*(x)]|$

$= |\frac{1}{B}\sum_{b=1}^{B} \mathbb{E}[\hat{f}_b(x) - f^*(x)]|$

$\leq \frac{1}{B}\sum_{b=1}^{B} |\mathbb{E}[\hat{f}_b(x) - f^*(x)]|$

$\leq$ "bias"

$\mathrm{Var}\,\hat{f}(x) = \mathrm{Var}\left(\frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(x)\right)$

$= \frac{1}{B^2}\mathrm{Var}\left(\sum_{b=1}^{B} \hat{f}_b(x)\right)$

$= \frac{1}{B^2}\sum_{b=1}^{B} \mathrm{Var}\left(\hat{f}_b(x)\right)$

$\leq \frac{1}{B}$ "Variance"

When do we have $\mathrm{Cov}(\hat{f}_b(x), \hat{f}_b(x)) = 0$?

Can happen, e.g., if each $\hat{f}_b$ is fit on separate, independent datasets.

Let's try to simulate this!

Bagging (bootstrap aggregating)

Given a sample $(x_1, y_1), \ldots, (x_n, y_n)$

a bootstrap subsample is a sample of n pts from this sample w/ replacement

For $j = 1, \ldots, n$:

- draw i_j at random from $\{1, \dots, n\}$

Return $(X_{i_1}, Y_{i_1}), \dots, (X_{i_n}, Y_{i_n})$

Note: - will have duplicates
- will miss some datapoints

Bagged trees:

For $b = 1, \dots, B$:

- Train a CART on a new bootstrap subsample of our data to get $\hat{f}_b$

Return $\hat{f}(x) = \frac{1}{B} \sum_b \hat{f}_b(x)$

Usually let depth limit $= \infty$
min # pts per leaf $= 1$

Random Forests

Idea: introduce extra randomness, in addition to bagging, to get $\text{Cor}(\hat{f}_b, \hat{f}_{b'})$ even smaller

Recall the recursive partitioning algo:

Input $X_1, Y_1, \dots, X_n, Y_n$

Find the best split $X_j \le b$ $\forall j = 1, \dots, p$ $\forall b \in \mathbb{R}$

New randomized version:

Input: —"—

Pick p' features out of $\{1, \dots, p\}$ at random from $\{1, \dots,$

Find the best split $X_{\cdot} \le$ $\forall$ [illegible]

if $\lambda_j = 0$ for ...

RF = (bagging of B trees w/ random subset of features considered at each node)

Usually $p' = \sqrt{p}$

Note: one tree in the ensemble may use all features at different nodes

Regression: we take avg of tree predictions

Classification: we take plurality vote over tree class predictions

or: avg of predicted probabilities