Lec 6

Tuesday, September 17, 2019 10:54

Recap: What makes a good supervised learning algorithm?

Data

$$\mathcal{D} = \{X_1, Y_1, \dots, X_n, Y_n\} \longrightarrow \boxed{\mathcal{A}} \longrightarrow \boxed{\hat{f}} \longrightarrow \hat{Y} \approx Y$$

$$X \to \hat{f}$$

$$R(\mathcal{A}) = \mathbb{E}_{\mathcal{D}}\left[R(\hat{f})\right] \qquad \hat{f} = \mathcal{A}(\mathcal{D})$$

$$= \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{X,Y}\left[(\hat{f}(X) - Y)^2\right]\right]$$

$$= \mathbb{E}_X \operatorname{Var}(Y|X) + \mathbb{E}_X \operatorname{Err}(X)$$

$$\operatorname{Err}(x) = \mathbb{E}_{\mathcal{D}}\left[\left(\hat{f}(x) - f^*(x)\right)^2\right]$$

$$= \underbrace{\operatorname{Var} \hat{f}(x)}_{\text{variance}} + \big(\underbrace{\mathbb{E}_{\mathcal{D}} \hat{f}(x) - f^*(x)}_{\text{bias}}\big)^2$$

Subset Selection

Find a subset of features that predict Y well.

Why?

- Avoid overfitting
- Trade off bias/var
- Interpretability

Best subset

For the rest of lecture

$\beta \in \mathbb{R}^{p+1}$ $\quad$ β_0 intercept

$\beta_{1:p}$ coeffs

Define $\text{supp}(\beta) = \{ j = 1, \cdots, p : \beta_j \neq 0 \}$

$$\text{supp}\left(\begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}\right) = \{1, 2\}$$

for any set $S \subseteq \{1, \cdots, p\}$

$$\hat{\beta}_S = \underset{\text{supp}(\beta) \subseteq S}{\text{argmin}} \| \underline{Y} - \underline{X}\beta \|_2^2 \leftarrow$$

OLS on the features $\{x_j : j \in S\}$ & an intercept

$$\hat{\beta}_{best\text{-}k} = \underset{|\text{supp}(\beta)| \leq k}{\text{argmin}} \| \underline{Y} - \underline{X}\beta \|_2^2$$

$$= \underset{\substack{|S| \leq k \\ S \subseteq \{1, \cdots, p\}}}{\text{argmin}} \| \underline{Y} - \underline{X}\hat{\beta}_S \|_2^2$$

$$\hat{\beta}_{best\text{-}p} = \hat{\beta}_{OLS}$$

Computing $\hat{\beta}_{best\text{-}k}$ is hard for p, k large

$$\binom{p}{k} \approx \left(\frac{p}{k}\right)^k$$

Idea: sequentially add/remove the most "influential" features

Large coeff $\rightarrow$ more influential?

But: subject to scaling

$\Rightarrow$ need to normalize

For $\hat{\beta}$ given by OLS, define the Z-score

$$z_j = \frac{\hat{\beta}_j}{\hat{\sigma} \cdot \sqrt{v_j}} \qquad \hat{\sigma}^2 = \frac{1}{n-k-1} \sum_i (Y_i - \hat{Y}_i)^2 \qquad v_j = \left((X^T X)^{-1} \right)_{jj}$$

General rule thumb: $|z_j| \geq 2 \Rightarrow$ "significant"

$< 2 \Rightarrow$ "insignificant"

Backward- & Forward-Stepwise Regression

BSR:

Start w/ $S = \{1, \dots, p\}$
while $|S| > k$:
- Compute $\hat{\beta}_S$
- Remove from S the feature w/ smallest abs z-score
- Repeat

FSR:

start w/ $S = \{\}$
while $|S| < k$:
- Find $j^* = \operatorname{argmin}_{j \notin S} \| Y - X \hat{\beta}_{S \cup \{j\}} \|_2^2$
- Add j^* to S
- Repeat

At what k to stop?

The AML approach: use cross-validation.

Cross-Validation (CV)

CV is a technique to estimating $R(\mathcal{A})$

Split the data into K folds:

$$\{1,\cdots,n\} = S_1 \cup \cdots \cup S_K$$

s.t. $S_i \cap S_j = \emptyset$

$$\big| |S_i| - |S_j| \big| \le 1$$

$$\hat{f}^{(j)} = \mathcal{A}\left(\{(X_i, Y_i) : i \notin S_j\}\right) \quad j=1,\cdots,K$$

$$CV^{(j)} = \frac{1}{|S_j|} \sum_{i \in S_j} l\left(Y_i, \hat{f}^{(j)}(X_i)\right)$$

$$\hat{R}^{cv}(\mathcal{A}) = \frac{1}{K} \sum_{j=1}^{K} CV^{(j)}$$

Collection of algorithms $\mathcal{A}_1, \cdots, \mathcal{A}_m$

How to choose?

Naïve approach: choose $\mathcal{A}_j$ w/ smallest $\hat{R}^{cv}(\mathcal{A}_j)$

The AML approach (AKA "one-std-err" rule of thumb)

$$\text{Std Err}(\hat{R}^{cv}(\mathcal{A})) = \frac{1}{K}\sqrt{\sum_j \left(\hat{R}^{cv}(\mathcal{A}) - CV^{(j)}(\mathcal{A})\right)^2}$$

Pick the "simplest" algo w/ $\hat{R}^{cv}$ within one std err of the minimal one

Simplest:
- least # of variables
- least complexity
- least high order dependence

- least var