Lec 7

Thursday, September 19, 2019 11:16

Recap:

- Bias-Var tradeoff
- Subset (feature) selection for OLS
  gives a range of complexities
- Cross-val as a way to
  a) Estimate the risk of any algo
  b) choose a complexity param
  (we often use the 1-std-err rule)

Shrinkage

Subset selection is very discrete
→ good for interpretation
→ potentially (not necessarily) bad for prediction

Shrinkage is a more cts way to trade off bias & var

$$\hat{\beta}^{ridge} = \underset{\beta \in \mathbb{R}^{p+1}}{\arg\min} \|Y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

excluding the intercept

after standardizing the data so all β on same sc

Shrinks $\underbrace{\hat{\beta}^{OLS}}_{\text{most complex model}}$ toward $\underbrace{\text{sample mean of } Y}_{\text{simplest model}}$

At one extreme $\lambda = 0$: $\hat{\beta}^{ridge} = \hat{\beta}^{OLS}$

— // — $\lambda = \infty$: $\hat{\beta}^{ridge}$ = intercept model at sample mean of Y

In low: shrinking toward 0 prevents β

from trying to reach far off observations using extreme slopes

$$\hat{\beta}^{ridge} = (X^T X + \Lambda)^{-1} X^T Y$$

$$\Lambda = \begin{pmatrix} 0 & & & 0 \\ & \lambda & & \\ & & \ddots & \\ 0 & & & \lambda \end{pmatrix}$$

Lasso

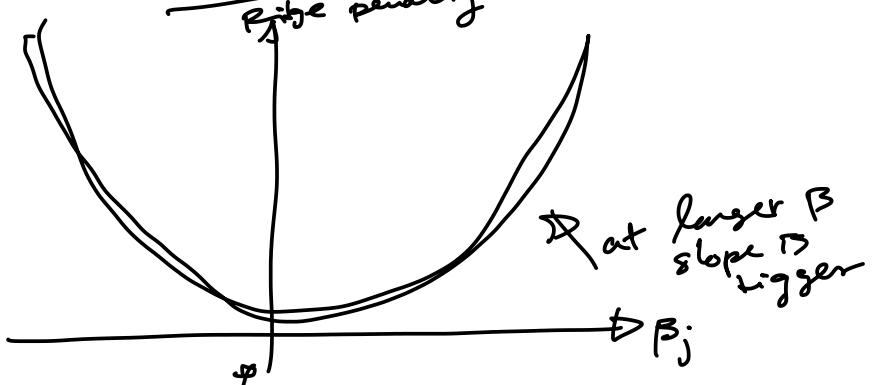
Combine shrinkage & subset selection

$$\hat{\beta}^{lasso} = \underset{\beta \in \mathbb{R}^{p+1}}{\operatorname{argmin}} \left(\|Y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right)$$

exclude intercept

Idea:

ridge penalty



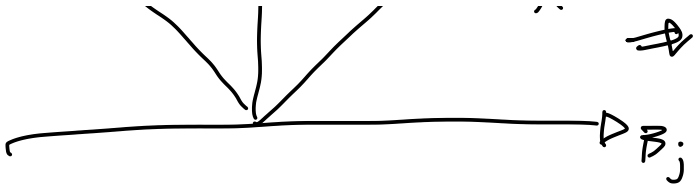
at larger β slope is bigger

at 0, have slope 0

$\Rightarrow$ pay essentially nothing to go from $\beta_j = 0$ to $\beta_j = \varepsilon$

lasso penalty

the absolute val of the slope is +1 everywhere



$\Rightarrow$ We start paying immediately for making β_j nonzero

This encourages 0 coeffs

No closed form for lasso

But can use grad descent methods (after full brea

In Sklearn, this is implemented in

sklearn.linear_model.lasso-path

compute lasso for all λ's simultaneously

Naive Bayes

Another ex of classification algo

$$\mathbb{P}(Y=j \mid X=x) = \frac{\mathbb{P}(X=x \mid Y=j)\,\mathbb{P}(Y=j)}{\sum_k \mathbb{P}(X=x \mid Y=k)\,\mathbb{P}(Y=k)}$$

(Bay
lan

$$= \frac{f_j(x)\,\pi_j}{\sum_k f_k(x)\,\pi_k}$$

π_j — fraction of label-j examples

$f_j(x)$ — prob/density of $X=x$ in the population of label-j example

$f_j(x) = f_j(x_1, \dots, x_p)$ a fn from $\mathbb{R}^p$ to $\mathbb{R}$

hard to estimate for large (even moderate) p

Naive assumption

$$f_j(x) = f_{j1}(x_1)\, f_{j2}(x_2) \cdots f_{jp}(x_p)$$
$$= \prod_{k=1}^{p} f_{jk}(x_k)$$

meaning: Assuming that $X_1, \dots, X_p$ are statistically independent given Y

$$\Rightarrow \mathbb{P}(Y=j \mid X=x) = \frac{f_j(x)\,\pi_j}{\sum_k f_k(x)\,\pi_k}$$

$$= \frac{\pi_j \prod_{l=1}^{p} f_{jl}(x_l)}{\sum_k \pi_k \prod_{l=1}^{p} f_{kl}(x_l)}$$

Naive Bayes classifier:

estimate each of π_j, f_{jl} $\forall j, l$
plug in above to get $\hat{\mathbb{P}}(Y=j \mid X=x)$
& mimic the Bayes classifier
by maximizing $\hat{\mathbb{P}}(Y=j \mid X=x)$

For class frequencies, use empirical frequencies

$$\hat{\pi}_j = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[Y_i = j]$$

What about f_{jl}?

Discrete features:

just use empirical frequencies

$$\hat{f}_{jl}(x_l) = \frac{\sum_{i=1}^{n} \mathbb{I}[y_i = j, X_{il} = x_l]}{\sum_{i=1}^{n} \mathbb{I}[y_i = j]}$$

What if $\hat{f}_{jl}(x_l) = 0$ while $\hat{f}_{kl}(x_l) \neq 0$?

Get zero prob no matter what happens w/ other p-1 features

One soln: Laplace smoothing

just pad the data

$$\hat{f}_{jl}(x_l) = \frac{\left(\sum_{i=1}^{n} \mathbb{I}[y_i = j, X_{il} = x_l]\right) + \alpha}{\left(\sum_{i=1}^{n} \mathbb{I}[y_i = j]\right) + \alpha}$$

$\alpha \geq 0$

<u>Continuous features</u>

Several options:

- kernel density estimation

  <u>next time</u>

- parametric density estimate

<u>Eg.</u> Assume $f_{jk}(x_k) = \varphi\left(\frac{x_k - \mu}{\sigma}\right)$

pdf of std normal

set $\hat{\mu}, \hat{\sigma}$ to the sample mean & std dev

of $\left\{ X_{ik} : \begin{array}{l} i = 1, \dots, n \\ Y_i = j \end{array} \right\}$