Day 5; 9/21/2018

Bias - Variance trade-off

Prediction for $x = x_0$ squared loss, averaged over all training and future data

$$\text{Error}(x_0) = \underset{\text{x is fixed; Y is random}}{E}[Y - \hat{h}_\delta(x_0)]^2 = E[(Y - h(x_0)) + (h(x_0) - E[\hat{h}_\delta(x_0)]) + (E[\hat{h}_\delta(x_0)] - \hat{h}_\delta(x_0))]^2$$

$$= \underbrace{E[Y - h(x_0)]^2}_{\text{irreducible error}} + \underbrace{E[h(x_0) - E\{\hat{h}_\delta(x_0)\}]^2}_{\text{bias}^2} + \underbrace{E[E\{\hat{h}_\delta(x_0)\} - \hat{h}_\delta(x_0)]^2}_{\text{variance}^2}$$

- cross-products cancel out

bias^2: model mis-specification

variance^2: variance of the sampling distribution of parameter estimates

[Ave model bias]^2 [Ave estimation uncertainty]^2

Specific to linear regression:

$Y = X\delta + \varepsilon$, $\text{Var}(\varepsilon) = \sigma^2 I$

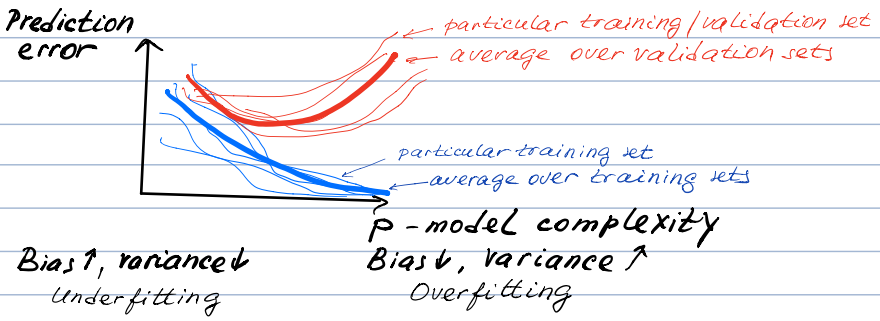
$$\text{Error}(x_0) = \sigma_\varepsilon^2 + E[h(x_0) - E\{\hat{h}_\delta(x_0)\}]^2 + \| X(X^TX)^{-1}x_0 \|_2^2 \sigma_\varepsilon^2$$

Average error, over all observed values x_0

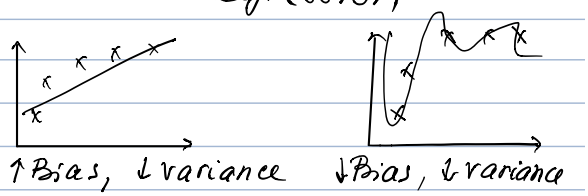
$$\frac{1}{N}\sum_{i=1}^{N} \text{Error}(x_0) = \sigma_\varepsilon^2 + \frac{1}{N}\sum_{i=1}^{N}[h(x_0) - E\{\hat{h}_\delta(x_0)\}]^2 + \frac{p}{N}\sigma_\varepsilon^2$$

How to estimate this error in practice? $\hat{\sigma}_\varepsilon^2 = \sum_{i=1}^{N} e_i^2$

Training data	$p \uparrow \Rightarrow$	$\downarrow$	$\downarrow$	- training set to estimate $\hat{h}_\delta$ & $\hat{\sigma}_\varepsilon^2$
future data	$p \uparrow \Rightarrow$	—	$\downarrow$	$\uparrow$ - training set to estimate $\hat{h}_\delta$, an independent set to estimate $\hat{\sigma}_\varepsilon^2$



Example: polynomial regression



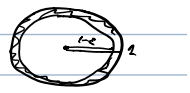
In high dimensions, we are likely to overfit: curse of dimensionality

In p dimensions, each observation is unique

A sphere with radius 1 in p dimensions: what fraction of the volume lies between $r = 1$ and $r = 1 - \varepsilon$? $V_p(r) = \text{constant}_p \cdot r^p$

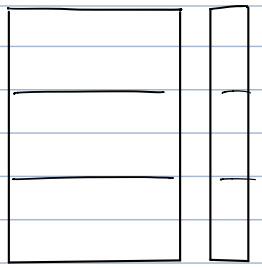
$$\frac{V_p(1) - V_p(1-\varepsilon)}{V_p(1)} = 1 - (1-\varepsilon)^p \xrightarrow[p \uparrow]{} 1 \quad \text{(even for small } \varepsilon\text{)}$$

$\Rightarrow$ most of volume is in the most distant points



Goal: minimize prediction error on future data

subset selection
regularization



- model fitting / training
- model development: evaluate competing models
- model evaluation: evaluate final model

Subset selection

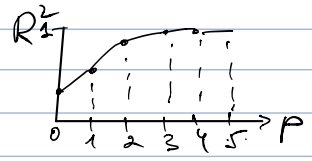
(1) Explore the space of candidate models - add feature engineering
(2) Evaluate the candidate models on the dev. set, keep the best performing
(3) Evaluate the selected model on the evaluation set

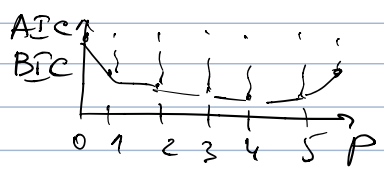
Exploring the model space in (1):
- Exhaustive search of $\binom{k}{p}$ predictors
- Forward stepwize selection
- Backward stepwize selection

} many heuristics exist

Example: forward stepwize selection
- Denote M_0 the null model with no predictors
- For $k = 0, \dots, p-1$:
  - consider all $p-k$ models that augment M_k with one predictor
  - $M_k \leftarrow$ the best among $p-k$ models
- select the model among $M_0, \dots, M_p$ that performs best on dev set

} R^2, AIC, BIC





$$AIC = \frac{SSE}{MSE_p} + 2p$$

$$BIC = \frac{SSE}{MSE_p} + \log N \, p$$

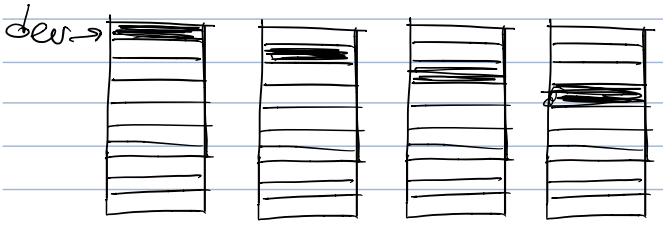
$$AIC, BIC = -2 \log likelihood + kp = \frac{\sum (y_i - \hat{y}_i)^2}{\hat{\sigma}^2} + kp$$

$2 \rightarrow \log N$

$\hat{\sigma}^2$ ← estimated using the saturated model

If N is relatively small - cross-validation

Iterate partitioning into train and dev sets



Problem: each iteration does not always have the same best model.

"Consensus" model may not perform as well

10-fold cross-validation: repeat 10 folds

Frequently used as an intermediate parameter tunning step

Statistical regularization: ridge regression

$y = X\beta + \varepsilon$, $\mathrm{Var}(\varepsilon) = \sigma^2 I$

Recall: Normal equations $\hat{\beta} = \arg\min_{\beta} (Y - X\beta)^T (Y - X\beta)$

Analytical solution: $\frac{dJ(\beta)}{d\beta} = \frac{d}{d\beta}(Y - X\beta)^T(Y - X\beta) \overset{\text{set to}}{=} 0$

$X^T X\beta - X^T Y = 0 \Rightarrow \hat{\beta} = (X^T X)^{-1} X^T Y$

($(X^T X)^{-1}$: singular if $N < p$ or collinear columns)

Properties of $\hat{\beta}$: $E\{\hat{\beta}\} = E\{(X^T X)^{-1} X^T (X\beta + \varepsilon)\} = \beta$ (over all data)

$\mathrm{Var}\{\hat{\beta}\} = \mathrm{Var}\{(X^T X)^{-1} X^T (X\beta + \varepsilon)\} = \sigma^2 (X^T X)^{-1} X^T I X (X^T X)^{-1} = \sigma^2 (X^T X)^{-1}$

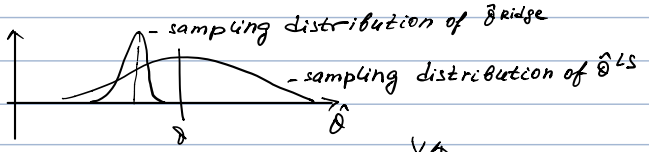
I.e.: - $\hat{\beta}$ is a linear combination of Y, is unbiased; $\mathrm{Var} = \sigma^2 (X^T X)^{-1}$

- can show that has min variance among all unbiased estimators (Gauss-Markov theorem)

Can improve the estimation when $X^T X$ is (nearly) singular: } can invert

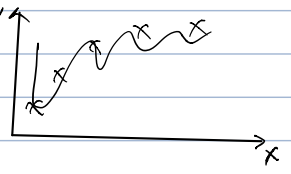
define $\hat{\beta}_{ridge} = (X^T X + \lambda I)^{-1} X^T Y$ — biased, but smaller variance }

(λ: pre-defined constant $\lambda > 0$)



- the coefficients are smaller in absolute value $\Rightarrow$ **shrinkage estimates**

Intuition: polynomial regression



$h_\beta(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4$ (*)

If we shrink β_3 and β_4 closer to 0 $\Rightarrow$ $h_\beta(x)$ is closer to quadratic

The same solution can be obtained using (1) penalized estimation

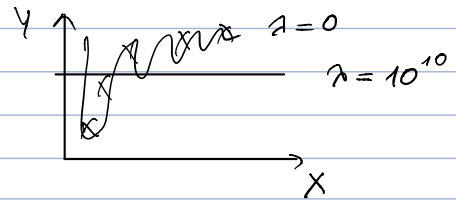
(2) constrained optimization

(1) $\hat{\beta} = \arg\min$ (New loss function $J_\lambda(\beta)$) $(Y - X\beta)^T(Y - X\beta) + \lambda \|\beta\|_2^2$ ($= \sum_{j=1}^{p} \beta_j^2$)

Solution as function of λ:

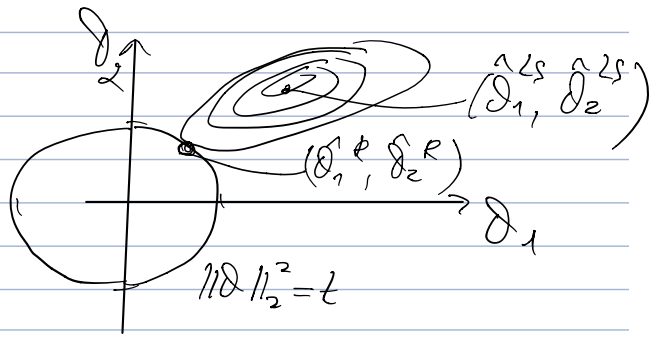
$\frac{d}{d\beta} J_\lambda(\beta) = 0 \qquad X^T X\beta - X^T Y + \lambda\beta = 0 \Rightarrow (X^T X + \lambda I)\beta - X^T Y = 0$

λ is a hyperparameter



(2) $\hat{\beta} = \arg\min (Y - X\beta)^T(Y - X\beta)$, s.t. $\|\beta\|_2^2 \le t$

($t = \|\hat{\beta}(\lambda)\|^2$)



Even though we have an analytical solution, can also solve numerically by gradient descent

Numeric parameter estimation in Ridge regression: batch gradient descent

- Initialize θ
- Repeat until convergence {
  - For $j = 1, \dots, p$ {

$$\theta_j \leftarrow \theta_j + \alpha \frac{d}{d\theta_j}\left[\sum_{i=1}^{N}(y_i - x^T\theta)^2 + \lambda \sum_{k=1}^{p}\theta_k^2\right]$$

  }
}

Details of the update:

$$\theta_j \leftarrow \theta_j - \alpha\left[2\sum_{i=1}^{N}(y_i - x^T\theta)x_{ij} + 2\lambda\theta_j\right]$$

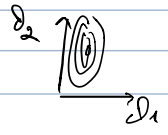
$$\theta_j \leftarrow \theta_j(1 - 2\alpha) - 2\alpha\sum_{i=1}^{N}(y_i - x^T\theta)x_i$$

usually < 1 ⇒ shrinkage

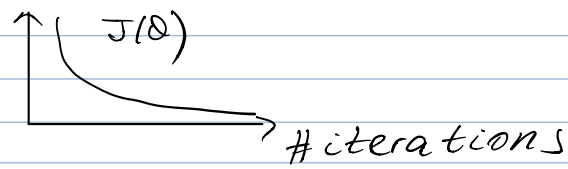
same as in least squares

Practical considerations for faster convergence:

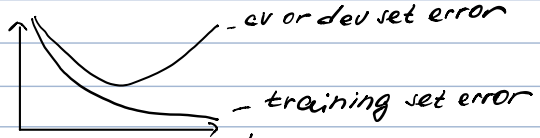
- scaling features (parameter values are comparable



- centering features (no intercept)
- trace plots of $J(\theta)$



Q: how to select λ? Try a range of λ; use dev set or cross-valid. to find the optimal point



model complexity (# params)	Low	high
λ	high	Low
Bias	high	Low
Variance	Low	high

$N \uparrow$	→	fix high variance
$p \downarrow$	→	—"—
$p \uparrow$	→	fix high bias
$\lambda \downarrow$	→	fix high bias
$\lambda \uparrow$	→	fix high variance