Statistical Learning II (9-11-2017)

1. Organization
2. Warm up
   a) Statistical Decision Theory
   b) ERM → least squares
3. K-fold Cross-validation
4. Model Selection
   a) Intro
   b) Best-subset
   c) Ridge Regression
   d) Lasso
5. Homework 1

1. Organization

- Anyone more time on exam?
- Usis / Blackboard / website

2. Warm up

supervised learning

- classification (e.g. spam, digits)
- regression (e.g. fitting polynomials)

$$T = \begin{pmatrix} y_1 \\ x_1 \end{pmatrix}, \dots, \begin{pmatrix} y_N \\ x_N \end{pmatrix}$$

↳ $\hat{f} \in \mathcal{F}$ and predict $\hat{y} = \hat{f}(x)$ for new x

2a Statistical Decision Theory

$EPE(f) = \underset{X,Y}{\mathbb{E}}[L(Y, f(X))]$ measures quality of f

regression: $L(Y, f(X)) = (Y - f(X))^2$ "squared error"

classification:

$$L(Y, f(X)) = \begin{cases} 0 & \text{if } f(X) = Y \\ 1 & \text{if } f(X) \neq Y \end{cases} \quad \text{"0/1-loss"}$$

Bayes optimal predictor:

$$f_B = \underset{f}{\operatorname{argmin}} \; EPE(f)$$

regression: $f_B(x) = \mathbb{E}[Y|X]$

classification: $f_B(x) = \underset{\hat{Y}}{\operatorname{argmax}} \Pr(Y = \hat{Y}|X)$

Estimators:

$\hat{f}$ depends on $\mathcal{T}$ $\Rightarrow$ $EPE(\hat{f})$ depends on $\mathcal{T}$

$\Rightarrow$ evaluate $\underset{\mathcal{T}}{\mathbb{E}}[EPE(\hat{f})]$

2b) Empirical Risk Minimization

$$\hat{f} = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \; \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i))$$

classification: - minimize #mistakes

- usually cannot compute efficiently for 0/1-loss

regression: - minimize sum of squared errors

- can compute if $\mathcal{F}$ is linear model

4. k-Fold Cross-Validation

Hyperparameter $m \in \{1, \ldots, M\}$

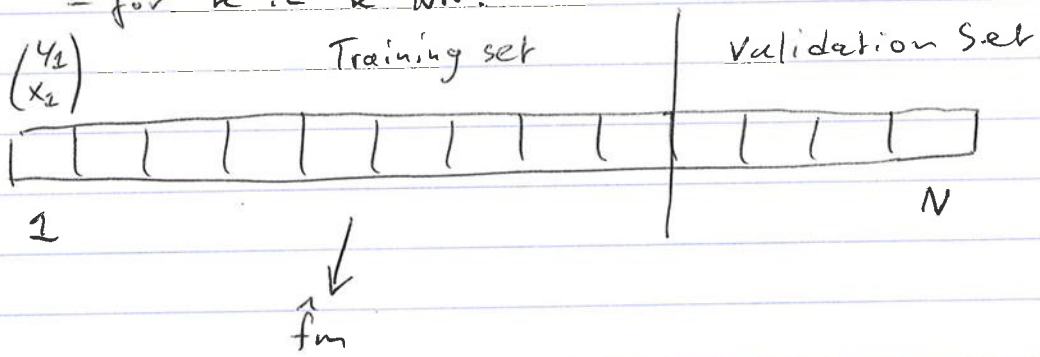
E.g. * m is degree of polynomial in linear regression
* m is k in k-nearest neighbour

Training set → $\hat{f}_m$ for each m.

~~What happens if I look at the fit on training data~~
~~1) Construct least squares estimate $\hat{f}_m$ for each degree polynomial~~

What happens if I look at the fit on the training data to choose m (i.e. I use ERM)?

- for degree of polynomial.
- for k in k-NN.



Hold-out estimate:

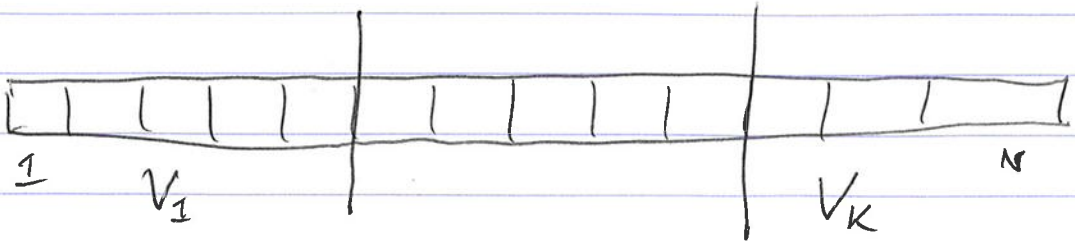
- Randomly set aside validation set V
- Construct $\hat{f}_m$ on rest of training data for all m
- Use ERM on validation set:

$$\hat{m} = \underset{m}{\operatorname{argmin}} \frac{1}{|V|} \sum_{i \in V} L(Y_i, \hat{f}_m(X_i))$$

- Works well if V is large enough, but that reduces training set a lot

K-Fold Cross validation:

Try to get away with smaller validation set by averaging multiple choices.



For $k = 1, \ldots, K$:

- train $\hat{f}_m$ on all data except V_k (for each m)
- evaluate on V_k: (use V_k as validation set)

$$\hat{L}_m^k = \frac{1}{|V_k|} \sum_{i \in V_k} L(y_i, \hat{f}_m(x_i))$$

$$\hat{m} = \underset{m}{\operatorname{argmin}} \frac{1}{K} \sum_{k=1}^{K} \hat{L}_m^k$$

(average over all choices of validation set)

Often train final $\hat{f}_{\hat{m}}$ on all data, or take the ~~mean~~ average of the ~~estimates~~ K estimators found for $\hat{m}$ during cross-validation.

$K = N$: "leave-one-out cross-validation"

(story about Mervyn Stone and LOO.)

5. Model Selection for Regression Intro

Suppose many features p. Maybe even $p \gg N$!

often not satisfied with least squares, because we want:

1. better prediction accuracy:

   least squares often has low bias, but high variance

2. better interpretability:

   If many features, which ones are most important?

   E.g. If we have 20 000 genes, want to know which ones are (most) relevant for prediction.

2. Best-subset Selection

Want to use only m most important features.
Select $m \leq p$ features as follows:

1. Run least squares for each of the $\binom{p}{m} \approx \left(\frac{p}{m}\right)^m$ subsets of m features.
2. Choose the subset such that $RSS(\hat{\beta})$ is smallest.

Different ways to choose m. E.g. use cross-validation.

Problems:

* Computation: exponentially many subsets in m
* Variance: discrete choice of subset; if multiple subsets approximately equally good, then which one of those is chosen varies a lot if we would sample a training set multiple times.

Computable approximations:

* Forward Stepwise: start with no variables,
  greedily choose one variable to add
  until we have m variables
* Backward Stepwise: start with all variables,
  greedily choose one variable to remove
  until we have m left.

3. Ridge regression

a) Definition

We set the first feature to 1: $X = \begin{pmatrix} 1 \\ x_2 \\ x_3 \\ \vdots \\ x_p \end{pmatrix}$

ridge regression: $\hat{\beta}_{ridge} = \underset{\beta}{\operatorname{argmin}} \; RSS(\beta) + \lambda \sum_{j=2}^{p} \beta_j^2$

$$= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \left(Y_i - \beta_1 - \sum_{j=2}^{p} x_{ij} \beta_j\right)^2 + \lambda \sum_{j=2}^{p} \beta_j^2$$

alternative formulation: ~~If $\hat{\beta}_{LS}$ unique~~ If $\hat{\beta}_{LS}$ unique:

$$\hat{\beta}_{ridge} = \underset{\beta}{\operatorname{argmin}} \; RSS(\beta)$$

$$\text{subject to } \sum_{j=2}^{p} \beta_j^2 \leq t$$

~~If $\hat{\beta}_{LS}$ unique then:~~

One-to-one relation between λ and t ~~[illegible]~~

b) Motivation

* More stable/less variance than least squares for highly correlated features

Example: two features the same: $x_2 = x_3$

Then $RSS(\beta) = \sum_{i=1}^{N} (Y_i - \beta_1 - \sum_{j=2}^{p} x_{ij} \beta_j)^2$ the same for

1.] $\beta = (\beta_1, \beta_2, \beta_3, \ldots, \beta_p)$ and

2.] $\beta = (\beta_1, \beta_2 + a, \beta_3 - a, \ldots, \beta_p)$ for any a.

so least squares does not distinguish

ridge regression prefers equal weights (situation 2)

For highly correlated features it makes the weights approximately equal

* $\hat{\beta}^{ridge}$ is always uniquely defined for $\lambda > 0$

Because $RSS(\beta) + \lambda \sum_{j=2}^{p} \beta_j^2$ is strictly convex.

Warning: need to normalize features
- If we make feature x_j 10 times as small, then least squares makes $\hat{\beta}_j$ 10 times as big and its predictions don't change.
- This does not hold for ridge regression, so need to center and rescale features to standard range

c) Computation

- Can interpret as least squares with fake training data.

$$\begin{pmatrix} y_{N+1} \\ x_{N+1} \end{pmatrix}, \dots, \begin{pmatrix} y_{N+p-1} \\ x_{N+p-1} \end{pmatrix}$$

$$y_{N+j} = 0 \qquad x_{N+j} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \sqrt{\lambda} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow (j+1)\text{-th position} \qquad \text{for } j = 1, \dots, p-1$$

$$(y_{N+j} - x_{N+j}^T \beta)^2 = \lambda \beta_{j+1}^2$$

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & & \\ x_{N1} & & x_{Np} \end{pmatrix} \qquad y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$$

with fake training data:

$$\bar{X} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & & \\ x_{N} & & x_{Np} \\ 0 & \sqrt{\lambda} & 0 & 0 & \cdots \\ 0 & 0 & \sqrt{\lambda} & 0 & \cdots \\ 0 & 0 & 0 & \sqrt{\lambda} & \\ 0 & 0 & 0 & 0 & \sqrt{\lambda} \end{pmatrix} \qquad \bar{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$\hat{\beta}_{LS} = (X^T X)^{-1} X^T y$$

$$\hat{\beta}_{ridge} = (\bar{X}^T \bar{X})^{-1} \bar{X}^T \bar{y}$$

$$= \left(X^T X + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \lambda \end{pmatrix}\right)^{-1} Xy$$

$$= (X^T X + \lambda A)^{-1} Xy \qquad A = \begin{pmatrix} 0 & & & \\ & 1 & & 0 \\ & & 1 & \\ & 0 & & \ddots \\ & & & & 1 \end{pmatrix}$$

Different from book, because deals with intercept by pre-processing the data, but is equivalent.

4. Lasso

a) Definition

$$\hat{\beta}_{Lasso} = \underset{\beta}{\operatorname{argmin}} \; RSS(\beta) + \lambda \sum_{j=2}^{p} |\beta_j|$$

If $\hat{\beta}_{LS}$ is unique, then

$$\hat{\beta}_{Lasso} = \underset{\beta}{\operatorname{argmin}} \; RSS(\beta)$$

$$\text{subject to } \sum_{j=2}^{p} |\beta_j| \leq t$$

One-to-one relation: $t \leftrightarrow \lambda$

b) Motivation

* Sets many weights β_j to 0 to find only most important features: good for prediction and interpretation

* $p \gg N$ allowed

NB Need to normalize features

c) Computation

- Exist multiple efficient algorithms
- LARS computes solutions for all values of t in one pass

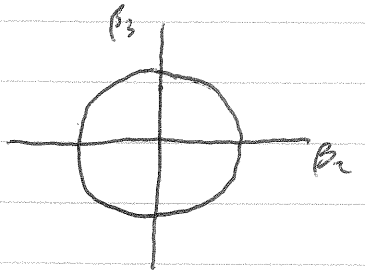
5. Comparison of Ridge, Lasso, Best-subset Selection

a) Ridge, Lasso vs Best-subset

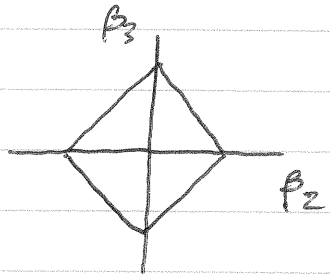
↑ small changes in data cause small changes in estimate, so less variance than best-subset.

b) Ridge vs Lasso

Ridge $\sum_{j=1}^{p} \beta_j^2 \leq t$

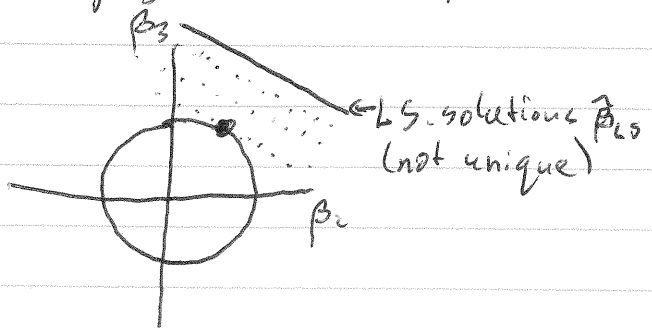


Lasso $\sum_{j=1}^{p} |\beta_j| \leq t$

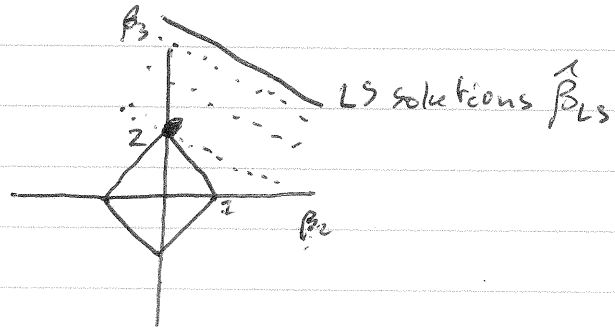


* Figure 3.11

* For highly correlated features:



β_2 and β_3 get about equal weight



minor noise determines which of corners 1 and 2 is chosen: one variable gets all weight, the other 0.

6. Homework 1

* Goal: try out linear regression methods on real data: predicting housing prices

* One of the features is racist
  Bad? Depends on what you do with predictions.

* Statistician's responsibility goes beyond technical aspects. Management does not know details of the methods used!