Statistical Learning II (5-11-2015)

1. Organization
2. Warm up:
    a) statistical decision theory
    b) ERM
3. Regression I: least squares
    a) 3 interpretations:
        ERM, ML, projection
    b) computation
    c) bias-variance decomposition
4. k-Fold Cross-validation

1. Organization

- Anyone more time on exam?
- Next week: start 11:15, work on homework 1 after lecture
- Lecture notes on website (not Blackboard)

2. Warm up

supervised learning

classification (e.g. spam, digits)

regression (e.g. fitting polynomials)

$$T = \begin{pmatrix} Y_1 \\ X_1 \end{pmatrix}, \ldots, \begin{pmatrix} Y_N \\ X_N \end{pmatrix}$$

$\hookrightarrow \hat{f} \in \mathcal{F}$ and predict $\hat{Y} = \hat{f}(X)$ for new X

(2)

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:

$$f_B = \underset{f}{\arg\min}\ EPE(f)$$

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

classification: $f_B(x) = \underset{\hat{y}}{\arg\max}\ Pr(Y = \hat{y} | X)$

Estimators:

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

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

2b) Empirical Risk Minimization

$$\hat{f} = \underset{f \in \mathcal{F}}{\arg\min}\ \frac{1}{N} \sum_{i=1}^{N} L(Y_i, f(X_i))$$

classification: - minimize #mistakes

- usually cannot compute efficiently for 0/1-loss.

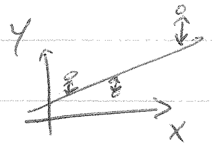
regression: - minimize sum of squared errors

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

3. Regression I: least squares

Linear model: $\hat{Y} = x^T \hat{\beta} = \hat{f}(x)$

$$RSS(\beta) = \sum_{i=1}^{N} (Y_i - x_i^T \beta)^2$$



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

a) Interpretations

I. ERM for squared error

$$\mathcal{F} = \{ f_\beta(x) = x^T \beta \mid \beta \in \mathbb{R}^p \}$$

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

II. Maximum Likelihood for Gaussian Errors

Model: $Y = x^T \beta + \varepsilon \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2)$

$$p_\beta(Y_i \mid X_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(Y_i - x_i^T \beta)^2}{2\sigma^2}}$$

$\boxed{\log(a \cdot b) = \log a + \log b}$

$$\hat{\beta} = \underset{\beta}{\operatorname{argmax}} \prod_{i=1}^{N} p_\beta(Y_i \mid X_i)$$

$$= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} -\log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(Y_i - x_i^T \beta)^2}{2\sigma^2}} \right)$$

$$= \underset{\beta}{\operatorname{argmin}} \ N \cdot \left(-\log \frac{1}{\sqrt{2\pi\sigma^2}}\right) + \sum_{i=1}^{N} -\log e^{-\frac{(Y_i - x_i^T \beta)^2}{2\sigma^2}}$$

$$= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{1}{2\sigma^2} (Y_i - x_i^T \beta)^2$$

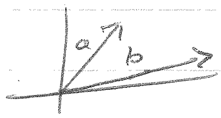
$$= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} (Y_i - x_i^T \beta)^2$$

noise magnitude σ^2 does not depend on X_i

So least squares is often associated with assumption of Gaussian errors, but it can be usefull more widely! (see I or III)

III L_2 - Projection

$a, b \in \mathbb{R}^N$



L_2-norm of a: $\|a\| = \sqrt{\sum_{i=1}^{N} a_i^2}$ is <u>length</u> of a

Distance between a and b: $\|a-b\| = \sqrt{\sum_{i=1}^{N} (a_i - b_i)^2}$

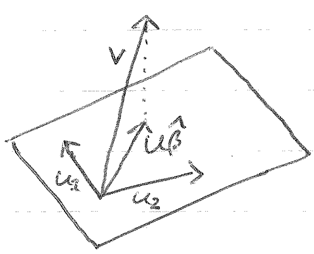
$$V = \begin{pmatrix} Y_1 \\ \vdots \\ Y_N \end{pmatrix} \qquad \begin{array}{c} p \text{ features} \\ U = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \\ x_{N1} & & x_{Np} \end{pmatrix} \\ \uparrow \qquad \uparrow \\ u_1 \;\cdots\; u_p \end{array} = \begin{pmatrix} x_1^T \\ \vdots \\ x_N^T \end{pmatrix}$$

$$(U\beta)_i = x_i^T \beta$$

$$RSS(\beta) = \sum_{i=1}^{N} (Y_i - x_i^T \beta)^2 = \|V - U\beta\|^2$$

$$\hat\beta = \underset{\beta}{\operatorname{argmin}}\, RSS(\beta) = \underset{\beta}{\operatorname{argmin}} \|V - U\beta\|^2 = \underset{\beta}{\operatorname{argmin}} \|V - U\beta\|$$

minimizes distance between V and $U\beta = u_1\beta_1 + \dots + u_p\beta_p$



$\{U\beta : \beta \in \mathbb{R}^p\}$

Least squares projects V onto hyperplane $\{U\beta : \beta \in \mathbb{R}^p\}$

Can be sensible even for non-Gaussian errors!

~~($\hat\beta$ unique $\iff$ $u_1, \dots, u_p$ linearly independent)~~

3b Computing the Least Squares Estimate $\hat{\beta}$

Before we start: minimizer $\hat{\beta}$ <u>not unique</u> if $u_1, \ldots, u_p$ not linearly independent (see projection picture) handle by pre-processing features

I. Gradient

$$a \in \mathbb{R}^p \qquad \nabla h(a) = \begin{pmatrix} \frac{\delta h(a)}{\delta a_1} \\ \vdots \\ \frac{\delta h(a)}{\delta a_p} \end{pmatrix}$$

generalizes derivative to functions of more than one variable.

If h convex, then $\hat{a}$ such that $\nabla h(\hat{a}) = \vec{0}$ is a minimum

II. Applied to RSS(β)

$$\frac{\delta RSS(\beta)}{\delta \beta_j} = \frac{\delta}{d\beta_j} \sum_{i=1}^{N} \left(y_i - \sum_j x_{ij} \cdot \beta_j\right)^2 = \sum_{i=1}^{N} 2\left(y_i - \sum_j x_{ij}\beta_j\right)(-x_{ij})$$

$$= 2 \sum_{i=1}^{N} x_{ij} \left(x_i^T \beta - y_i\right)$$

$$\nabla RSS(\beta) = 2 \sum_{i=1}^{N} x_i \left(x_i^T \beta - y_i\right) = 2\left(\sum_{i=1}^{N} x_i x_i^T\right)\beta - 2\sum_{i=1}^{N} x_i y_i$$

$$\nabla RSS(\hat{\beta}) = \vec{0}: \quad \left(\sum_{i=1}^{N} x_i x_i^T\right)\hat{\beta} = \sum_{i=1}^{N} x_i y_i$$

Suppose only one feature ($p=1$): $\hat{\beta} = \dfrac{\sum_i x_i y_i}{\sum_i x_i^2}$

What is $\hat{\beta}$ if this feature is always 1? $\leftarrow$ mean of y_i

6

For general p: $\left(\sum_{i=1}^{N} x_i x_i^T\right)_{j,k} = \sum_i x_{ij} \cdot x_{ik} = u_j^T u_k = (\mathcal{U}^T\mathcal{U})_{j,k}$

$$\sum_i x_i x_i^T = \mathcal{U}^T\mathcal{U}$$

$$\sum_{i=1}^{N} x_i y_i = \mathcal{U}^T v$$

Thus: $\mathcal{U}^T\mathcal{U}\hat{\beta} = \mathcal{U}^T v$

$$\hat{\beta} = (\mathcal{U}^T\mathcal{U})^{-1}\mathcal{U}^T v$$

What if $u_1, \dots, u_p$ are not linearly independent?
Then inverse does not exist!

3c. Bias-Variance Decomposition

Goal: understand how more features = larger model $\mathcal{F}$ affects $\mathop{\mathbb{E}}_T[EPE(\hat{f})]$: (balance between under- and overfitting)

Thm. Let $\hat{f}$ be any estimator. Let $\bar{f} = \mathop{\mathbb{E}}_T[\hat{f}]$. Then

$$\mathop{\mathbb{E}}_T \mathbb{E}PE(\hat{f}) = EPE(f_B) \leftarrow \text{Bayes error}$$

$$+ \mathop{\mathbb{E}}_X (f_B(x) - \bar{f}(x))^2 \leftarrow \text{bias (typically } \cancel{\text{larger}} \text{ smaller with larger } \mathcal{F})$$

$$+ \mathop{\mathbb{E}}_T \mathop{\mathbb{E}}_{X,Y} (\hat{f}(x) - \bar{f}(x))^2 \leftarrow \text{variance (typically larger with larger } \mathcal{F})$$

Proof: see lecture notes week 1. □

(show polynomial slides + bias-variance picture from book)

Suppose linear model is correct:

$$y = x^T \cdot \beta^* + \varepsilon \quad \text{with independent noise s.t. } \mathbb{E}[\varepsilon] = 0$$

Conditional on features $\mathcal{U}$ in training set:

$$\mathbb{E}_T[\hat{\beta} \mid \mathcal{U}] = \mathbb{E}_T[(\mathcal{U}^T\mathcal{U})^{-1}\mathcal{U}^T v \mid \mathcal{U}]$$

$$= (\mathcal{U}^T\mathcal{U})^{-1}\mathcal{U}^T \, \mathbb{E}[v \mid \mathcal{U}]$$

$$= (\mathcal{U}^T\mathcal{U})^{-1}\mathcal{U}^T \mathcal{U}\beta^*$$

$$= \beta^*$$

$$\mathbb{E}_T[\hat{\beta}] = \mathbb{E}_{\mathcal{U}}\,\mathbb{E}_{T|\mathcal{U}}[\hat{\beta} \mid \mathcal{U}] = \beta^*$$

Bias is 0, because $\bar{f}(x) = \mathbb{E}_T[x^T\hat{\beta}] = x^T\,\mathbb{E}_T[\hat{\beta}] = x^T\beta^*$

$= f_B(x)$.

$\Uparrow$

if model correct!!

Suppose model correct + Gaussian noise.

$$y = x^T\beta^* + \varepsilon \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

$\mathrm{EPE}(f_B) = \sigma^2$

bias $= 0$

variance $\to \sigma^2 \cdot \frac{p}{N}$ as $N \to \infty$ (book p. 26)

$$\mathbb{E}_T[\mathrm{EPE}(\hat{f})] \to \sigma^2 + 0 + \sigma^2 \cdot \frac{p}{N} \quad \text{as } N \to \infty$$

model correct + Gausian noise!

still OK for pretty large p!

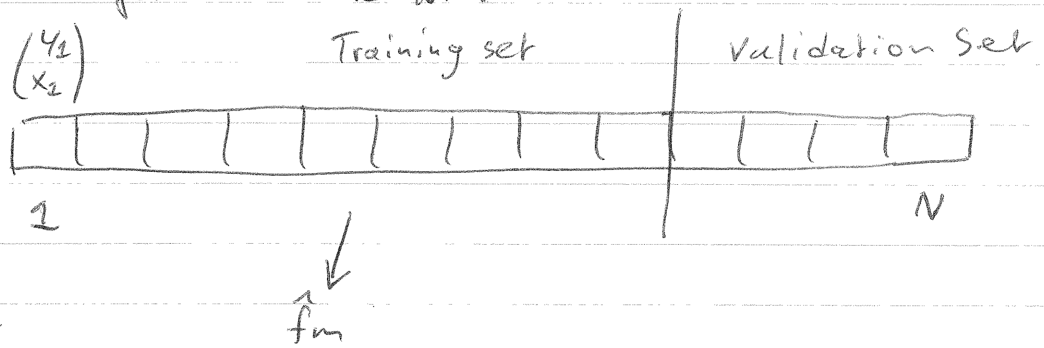
(beat curse of dimensionality by imposing linear model)

(discuss bias-variance trade-off)

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 $\rightarrow \hat{f}_m$ 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.



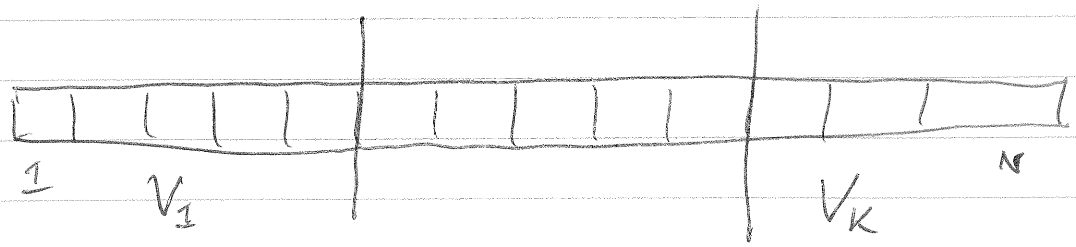
Hold-out estimate:
- Randomly set aside validation set V
- Construct $\hat{f}_m$ on rest of training data for all m
- Use ERM on validation set:

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

- Works well if ~~validset~~ 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 $k = 1, \dots, K$:

- 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"