Statistical Learning I

1. Organization
2. Statistical Learning Intro:
   a) supervised learning: classification, regression (overfitting)
   b) simplistic: linear regression for classification (overfitting 2)
   c) nearest neighbour (overfitting 3)
3. Statistical Decision Theory:
   EPE (overfitting 4), Bayes optimal decision
4. Curse of Dimensionality
5. Bias-Variance (overfitting 5)
(6. Short History)
~~4. Exercise: least squares vs maximum likelihood~~

1. Organization

Tim van Erven
TA: Dirk van der Hoeven

www.timvanerven.nl/teaching/statlearn2019

Cparticipant list, show book, BSc math students need permission from exam committee to take this course!

"Computer Age Statistical Inference" Efron, Hastie

* 2 large homework assignments > 5.5
* open book exam: > 5.5

HW1: Nov 8 - Nov 24
HW2: Nov 25- Dec 15

(lectures + work on homework, first four weeks longer sessions, last four weeks shorter)

Room 407/409 until 13h15
Room 174 after 14h15

Enroll in Blackboard + Usis!

(2)

2a) Supervised Learning

Training data $T = \begin{pmatrix} y_1 \\ x_1 \end{pmatrix}, \ldots, \begin{pmatrix} y_N \\ x_N \end{pmatrix}$

x_i : feature vectors
y_i : class / response variable (teacher shows us what desired response y_i is for input x_i)

Goal: learn function $\hat{f} : X \to y$
from model $\mathcal{F}$ = some set of functions

Evaluate $\hat{f}$ on test data:
new X from same source,
predict corresponding Y by $\hat{Y} = \hat{f}(X)$

Assume: $\begin{pmatrix} y_i \\ x_i \end{pmatrix}$ pairs are independent samples
from same probability distribution P^*

Avoid further assumptions on P^*!
(P^* can be very complicated.)

Classification: $\begin{pmatrix} G_1 \\ X_1 \end{pmatrix}, \begin{pmatrix} G_2 \\ X_2 \end{pmatrix}, \ldots, \begin{pmatrix} G_N \\ X_N \end{pmatrix}$

G is a categorical variable. (e.g. yes/no or apple/orange/pear)

$G \to Y$

yes → 1
no → 0

yes → +1
no → −1

apple → 1
orange → 2
pear → 3

apple → $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$
orange → $\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$
pear → $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

"dummy coding"
"one-hot coding"

Linear Regression:

Y_i is real-valued ($X_{i,j}$ in reals or use dummy coding)
model $\mathcal{F}$ can be represented as linear functions:

$$\hat{Y}_i = f_\beta(X_i) = \beta_0 + \sum_{j=1}^{p} X_{i,j} \cdot \beta_j \qquad (\beta = (\beta_0, \beta_1, \ldots, \beta_p))$$

$$= X_i^T \beta \qquad (X_{i,0} = \text{always } 1)$$

Least Squares: choose $\hat{\beta}$ to minimize

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

N.B. X_i random design, not set by experimenter

(slides)

Important trick: make more features

* Given $X_i \in \mathbb{R}$
* Make $\tilde{X}_{i,0} = 1$
  $\tilde{X}_{i,1} = X_i$
  $\tilde{X}_{i,2} = X_i^2$
  $\tilde{X}_{i,3} = X_i^3$
  etc.

Model linear in new features, but not in original features!

(polynomial slides)

OVERFITTING is central concern of statistical learning!

2b) Linear Regression for Classification

N.B. Logistic regression is better!

$G_i \in \{\text{blue}, \text{orange}\} \rightarrow y_i \in \{0, 1\}$

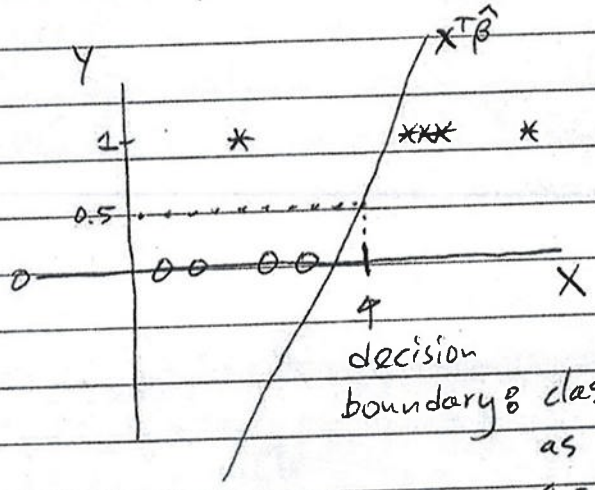
$x_i \in \mathbb{R}^2$

Simple idea:

- use linear regression on this data
- convert back to binary predictions:

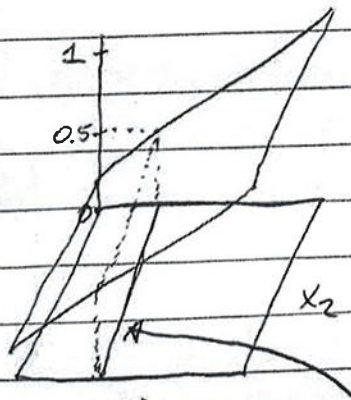
$$x^T\hat{\beta} > 0.5 \Rightarrow \hat{f}(x) = 1$$

$$x^T\hat{\beta} \leq 0.5 \Rightarrow \hat{f}(x) = 0$$

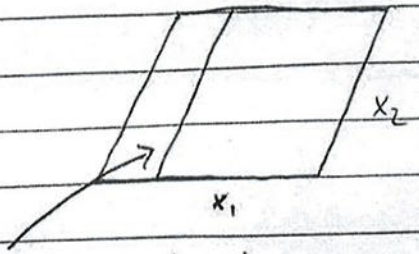


(1 feature)

decision boundary: classify everything left of boundary as 0 = blue, everything on right as 1 = orange.



(2 features)



decision boundary: classify as 0 on one side, as 1 on other side.

(show first figure from book)

- Many errors on training set. Avoidable?
- Maybe if we construct more features:
  $x_1, x_2, x_1^2, x_2^2, x_1 \cdot x_2, x_1^2 \cdot x_2, \ldots$
  Too few: underfitting
  Too many: overfitting

2C) K-Nearest Neighbour Classification

To classify new X:
- Ask K points in T closest to X
- Choose most common class

For two classes: $y_i \in \{0, 1\}$

$\hat{y} = \frac{1}{K} \sum_{x_i \in N_K(x)} y_i$ ⇐ estimates $Pr(Y=1|X)$ by average in region around X

$\hat{y} > 0.5 \Rightarrow$ predict 1
$\hat{y} \leq 0.5 \Rightarrow$ predict 0

K small: very flexible decision boundary, but very unstable estimates based on few neighbours (risks overfitting)
K large: stable estimates, but inflexible decision boundary (risks underfitting)

Model $\mathcal{F}$ is <u>implicit!</u>

If neighbourhoods do not overlap: $\frac{N}{K}$ neighbourhoods
⇒ effective number of parameters, because estimate $\hat{y}$ for each neighbourhood.

(Show pictures book.
What is best K?
Need statistical decision theory!)

3. Statistical Decision Theory

Learn $\hat{f}$ from training data $\begin{pmatrix} Y_1 \\ X_1 \end{pmatrix}, \dots, \begin{pmatrix} Y_N \\ X_N \end{pmatrix} \overset{\text{i.i.d.}}{\sim} p^*$
Evaluate on new $\begin{pmatrix} Y \\ X \end{pmatrix} \sim p^*$

Loss function $L(Y, \hat{Y})$ measures how bad it is if truth is Y, but we predict $\hat{Y} = \hat{f}(X)$ (smaller is better)

Regression: $L(Y, \hat{Y}) = (Y - \hat{Y})^2$ "squared error"
Classification: $L(Y, \hat{Y}) = \begin{cases} 0 & \text{if } \hat{Y} = Y \\ 1 & \text{if } \hat{Y} \neq Y \end{cases}$ "0/1-loss"

Other choices possible! (Depends on what is important in your application.)

$(X, Y) \sim p^*$ random, so want small **expected prediction error**: ← often also called "risk"

$$EPE(f) = \mathbb{E}_{X,Y \sim p^*}[L(Y, f(X))]$$

← we want $\hat{f}$ such that $\mathbb{E}_{T}[EPE(\hat{f})]$ as small as possible

Classification:

$$EPE(f) = Pr(Y \neq f(X)) \cdot 1 + Pr(Y = f(X)) \cdot 0$$
$$= Pr(Y \neq f(X))$$

Bayes Optimal Decisions

Bayes optimal f minimizes $EPE(f)$ over all possible functions f:

$$f_B = \operatorname{argmin}_f \mathbb{E}_{X,Y}[L(Y, f(X))]$$

$$f_B(X) = \operatorname{argmin}_{\hat{Y}} \mathbb{E}_{p^*(Y|X)}[L(Y, \hat{Y})]$$

- Unknown, because depends on P^*
- No learning alg can do better (by definition)
- Nothing to do with "Bayesian statistics"

Classification with 0/1-loss:

$$f_B(x) = \underset{\hat{y}}{\operatorname{argmin}}\ P^*(y \neq \hat{y} \mid x)$$

$$= \underset{\hat{y}}{\operatorname{argmax}}\ P^*(y = \hat{y} \mid x)$$

$$= \begin{cases} 1 & \text{if } P^*(y=1 \mid x) > 0.5 \\ 0 & \text{if } P^*(y=1 \mid x) \leq 0.5 \end{cases} \quad \text{(for two classes)}$$

Regression with Squared Error:

$$f_B(x) = \underset{\hat{y}}{\operatorname{argmin}}\ \underset{P^*(y|x)}{\mathbb{E}}\left[(y-\hat{y})^2\right]$$

$$= \underset{P^*(y|x)}{\mathbb{E}}[y] = \mathbb{E}[y \mid x]$$

If $y = f^*(x) + \text{noise}$ (noise: mean 0)

Then $f_B(x) = \mathbb{E}[y \mid x] = f^*(x) + \mathbb{E}[\text{noise} \mid x] = f^*(x)$

Empirical Risk Minimization

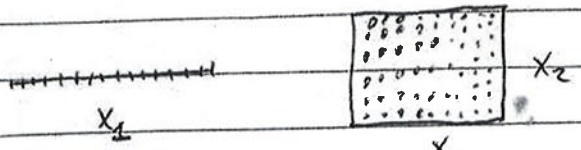
- Approximate P^* by empirical distribution of T (training data)
  and choose $f \in \mathcal{F}$ that minimizes

$$\frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i))$$

Is this always best? No!

1. Overfitting (see K-NN picture train and test error as function of K)
2. Restricted to model $\mathcal{F}$ (potential underfitting)

4. Curse of Dimensionality

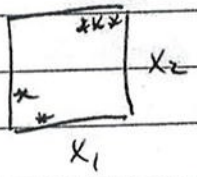


1 feature, N samples

2 features, N^2 samples for same coverage

... p features, N^p samples for same coverage

Conclusion: In high dimensions you only see samples from very small part of domain.



("A high-dimensional space is a lonely place."
- Bernard Schölkopf)

5. Bias-Variance Decomposition for Squared Error

Understand balance between under- and overfitting.

Thm: Let $\bar{f}(x) = \mathop{\mathbb{E}}_{T}[\hat{f}(x)]$. Then

$$\mathop{\mathbb{E}}_{T}[EPE(\hat{f})] = \mathop{\mathbb{E}}_{X}[Var(Y|X)] \quad \leftarrow \text{Bayes error } EPE(f_B)$$

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

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