인공지능을 위한 이론과 모델링

Introduction

서울대 통계학과 장원철

담당교수

- 장원철

- 25동 323호
- wcjang@snu.ac.kr
- Office Hours: 수 오후 2-4시

- 김건희

- 302동 327호
- gunhee@snu.ac.kr

- 김지수

- 25동 212호
- jkim82133@snu.ac.kr
- Office Hours: 수 오후 2-4시

담당조교

- 신민섭

- jpoth1729@snu.ac.kr
- 25동 402호
- Module 1& 3 담당

과목 홈페이지

- eTL: 숙제, 시험, 각종 공지사항
- Github (https://sfamsnu.github.io/fall23/): 강의노트
- 강의노트는 반드시 본인이 프린트를 한 후 가지고 올 것!

강의구성

Date	Topic	Reading	HW due
9월 4일	Introduction and Review		
9월 6일	Density Estimation		
9월 11일	Nonparametric Regression I		
9월 13일	Nonparametric Regression II		
9월 18일	High-dimensional Regression		
9월 20일	Classification I		HW 1 due (9/20)
9월 25일	Classification II		
9월 27일	Clustering I		
10월 2일	임시공휴일		
10월 4일	Clustering II		
10월 6일	**Exam 1**		HW 2 due (10/6)
10월 9일	한글날 (공휴일)		
10월 11일	Probabilistic Graphical Models		
10월 16일	Bayesian Networks I		
10월 18일	Bayesian Networks II		
10월 23일	Markov Random Fields		HW 3 due (10/25)
10월 25일	Unified View of BN and MRF		
10월 30일	Gaussian Network Models		
11월 1일	Causality I		
11월 6일	Causality II		
11월 8일	**Exam 2**		HW 4 due (11/8)
11월 13일	Nonparametric Bayesian Inference		
11월 15일	Concentration Inequality		
11월 20일	Minimax Theory I		
11월 22일	Minimax Theory II		HW 5 due (11/24)
11월 27일	Conformal Prediction		
11월 29일	Differential Privacy		
12월 4일	Wasserstein Distance and Optimal Transport		
12월 6일	High-dimensional Two Sample Testing		HW 6 due (12/8)
12월 11일	Dimension Reduction		
12월 13일	**Exam 3**		

- 3개의 Module로 구성
- Module I : 장원철
- Module 2 : 김건희
- Module 3 : 김지수

평가

- 숙제 (20%): 격주 총 6회
  - 과제의 프로그래밍 부분은 R 또는 파이썬을 사용하여 제출한다.
- 시험 (75%)
  - 10월 6일 (금) 6-8pm
  - 11월 8일 (수) 6-8pm
  - 12월 13일 (수) 6-8pm
- 출석/수업참여 (5%)

선수과목

- 수리통계 1 (326.311)
- 데이터 마이닝 방법 및 실습 (326.413)/기계학습 개론(4190.428)

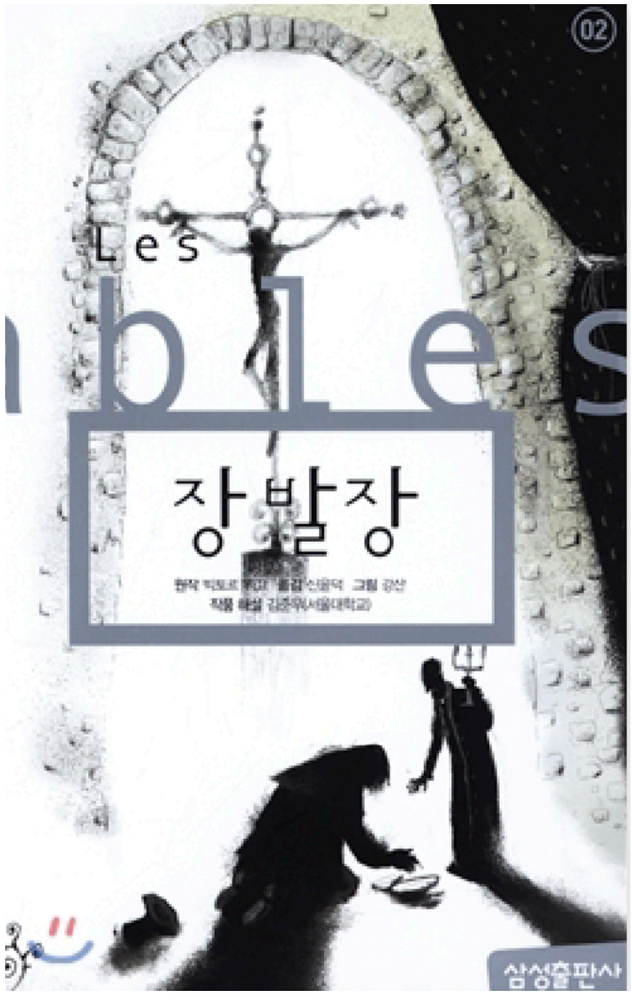
참고문헌

- Bishop, C. (2006). Pattern Recognition and Machine Learning. Springer. ISBN 978- 0387310732.
- Murphy, K. (2021). Probabilistic Machine Learning: An Introduction. MIT Press. ISBN 978-0-262-046824
- Shalev-Shwartz and Ben-David (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press. ISBN 978-1107057135.
- Wasserman, L. (2004). All of Statistics: Concise Course in Statistical Inference. Springer. ISBN 978-0387402727.
- Wasserman, L. (2005). All of Nonparametric Statistics. Springer. ISBN 978-0387251455.
- Zhang, T. (2023). Mathematical Analysis of Machine Learning Algorithms. Cambridge University Press. ISBN 978-1009098380

온라인 수업시간에 준수해야 할 사항

- 수업시간에 다른사람과 대화등으로 수업을 방해하지 않는다.
- 수업에 늦으면 다른 학생들에게 방해되지 않게 조용하게 들어와 앉는다.
- 전자기기(아이패드, 갤럭시 탭, 노트북 등)는 필기용으로 사용하는 때에만 수업 시간에 사용할 수 있다. 휴대폰은 어떤 경우에서 사용을 금지한다.
- 특별한 이유없이 결석이 잦을 경우 수업참여 점수가 0점처리된다.

장발장 vs 레미제라블



Module 1

Module 1 강의구성

Date	Topic	Reading	HW due
9월 4일	Introduction and Review		
9월 6일	Density Estimation		
9월 11일	Nonparametric Regression I		
9월 13일	Nonparametric Regression II		
9월 18일	High-dimensional Regression		
9월 20일	Classification I		HW 1 due (9/20)
9월 25일	Classification II		
9월 27일	Clustering I		
10월 2일	임시공휴일		
10월 4일	Clustering II		
10월 6일	**Exam 1**		HW 2 due (10/6)

What is Machine Learning?

- A computer program is said to learn from experience **E** with respect to some class of tasks **T**, and performance measure **P**, if its performance at tasks in **T** as measured by **P**, improves with experience **E**. - *Tom Michell*

- The main goals of machine learning are
  - Develop statistical models and estimation procedures that are scalable (computationally efficient)
  - Make effect use of available data (statistically efficient) to make accurate prediction

Art and Science of ML

- The choice of methodology for a problem is usually based on intuition and experience gained in practice - This is the **art** part of ML

- Understanding the nature of models is the **science** part of ML

- **Science** can inform **art** via theoretical analysis of statistical models to help the choice of models

- Intuition and experience can give insight into the properties to be proved

Types of Machine Learning

- **"Pure" Reinforcement Learning** (cherry)
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning** (icing)
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Unsupervised/Predictive Learning** (cake)
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

Yann LeCun's NIPS'16 talk

Supervised Learning

Classification

- In supervised learning, the task *T* is to learn a mapping f from inputs (features, predictors, covariates) $x \in \mathcal{X}$ to outputs (labels, targets, responses) $y \in \mathcal{Y}$.
- The experience *E* is a set of *N* pairs $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$, known as the training set.
- The performance measure *P* depends on the type of output.
- If $\mathcal{Y} = \{1,2,\ldots,C\}$, then this is a classification problem.
- A common performance measure for classification is the misclassification rate $\mathcal{L}(\theta) \equiv \frac{1}{N}\sum_{i=1}^{N} I(y_n \neq f(x_n;\theta))$

Supervised Learning

Fisher's Iris flowers

- Predictors: sepal length, sepal width, petal length, petal width
- Response: Type of Iris flowers (setosa, Versicolor, Virginica)
- Sample size, *N* =150,
- *N*umber of predictors, *D*=4

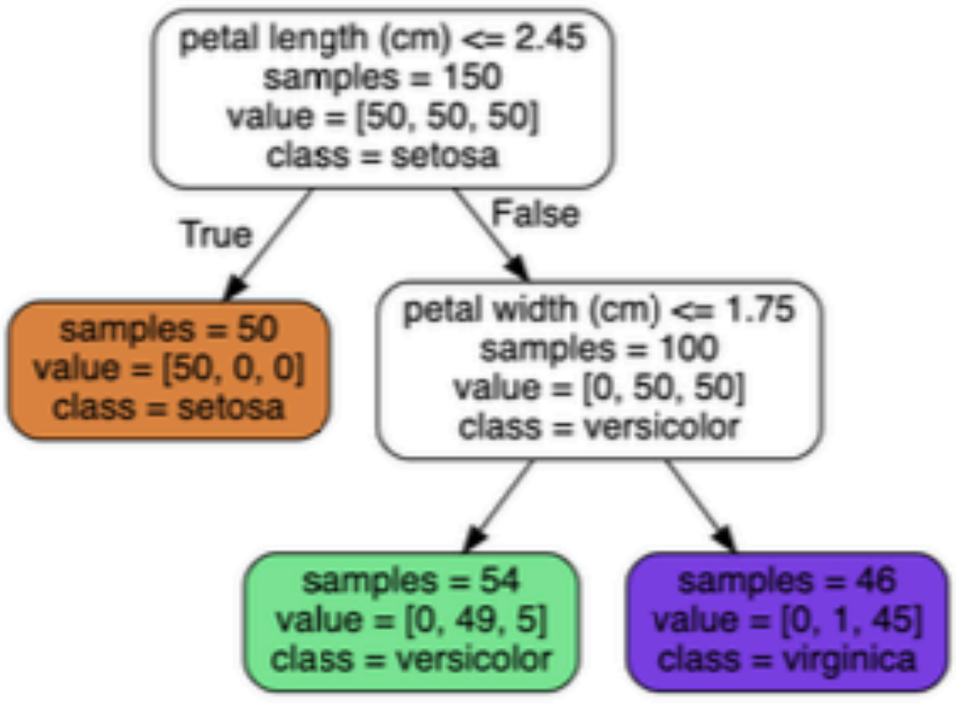


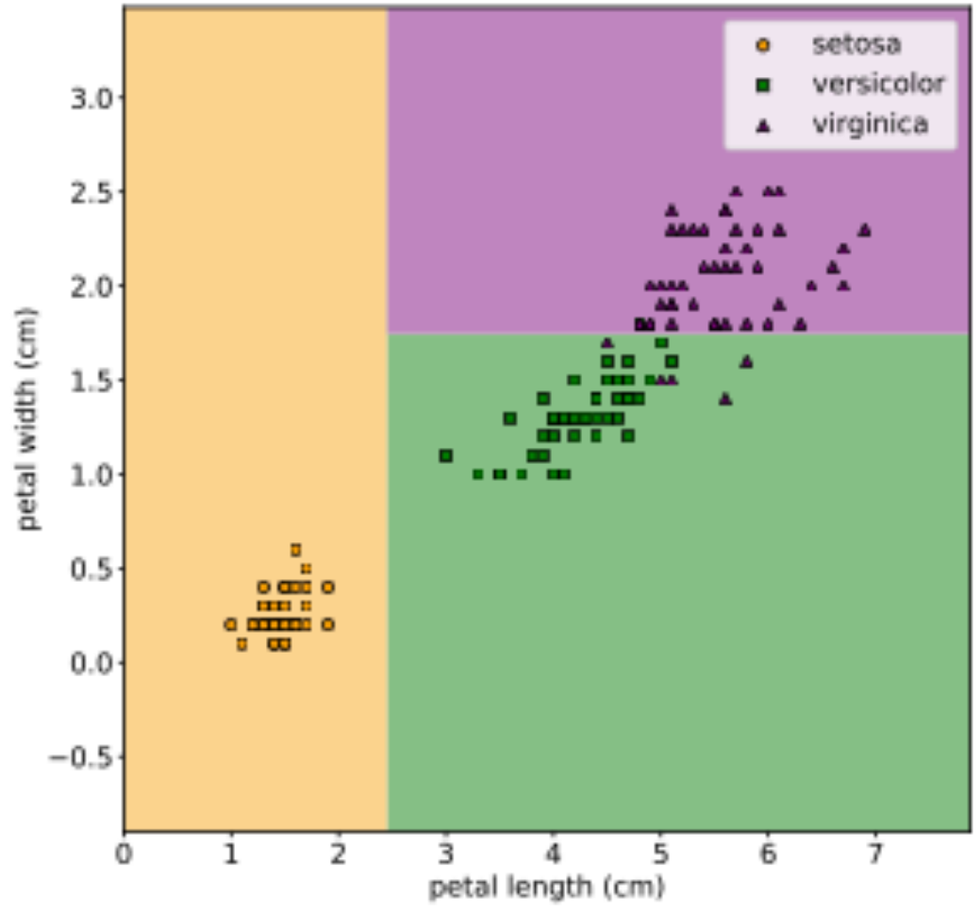
index	sl	sw	pl	pw	label
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
	...				
50	7.0	3.2	4.7	1.4	Versicolor
	...				
149	5.9	3.0	5.1	1.8	Virginica

Supervised Learning

Fisher's Iris flowers

- Decision tree and decision boundary





Supervised Learning

Empirical risk minimization

- Define the empirical risk, a generalized performance measure.
  $\mathscr{L}(\theta) \equiv \frac{1}{N}\sum_{i=1}^{N} \ell(y_n, f(x_n;\theta))$ where $\ell(y,\hat{y})$ is a loss function.

- To fit the best model is to find the optimal parameters that minimizes the empirical risk on the training set:

$$\hat{\theta} = \text{argmin}_\theta \mathscr{L}(\theta) = \text{argmin}_\theta \frac{1}{N}\sum_{n=1}^{N} \ell(y_n, f(x_n,\theta))$$

- This is called **empirical risk minimization**.

Supervised Learning

Regression

- Suppose $y \in R$, then this is known as the regression problem.
- For regression, the common choice of the loss function is the quadratic loss: $\ell(y, \hat{y}) = (y - \hat{y})^2$.
- Another common choice is the negative log probability: $\ell(y, f(x;\theta)) = -\log p(y \,|\, f(x;\theta))$.
- A simple linear regression can be expressed as follows:
- $f(x;\theta) = b + wx$ where w is the slope, b is the intercept, and $\theta = (w, b)$

Supervised Learning

Overfitting and generalization

- We can rewrite the empirical risk as follows:

$$\mathscr{L}(\theta, D_{\text{train}}) = \frac{1}{|D_{\text{train}}|} \sum_{(x,y)\in D_{\text{train}}} \ell(y, f(x;\theta))$$

- Population risk: $\mathscr{L}(\theta; p^*) \equiv E_{p^*(x,y)}[\ell(y, f(x;\theta))]$ where p^* is the true joint distribution of (x, y).

- The difference between the population risk and empirical risk is called the generalization gap.

- Test risk: $\mathscr{L}(\theta, D_{\text{test}}) = \frac{1}{|D_{\text{test}}|} \sum_{(x,y)\in D_{\text{test}}} \ell(y, f(x;\theta))$

Supervised Learning

Overfitting and generalization

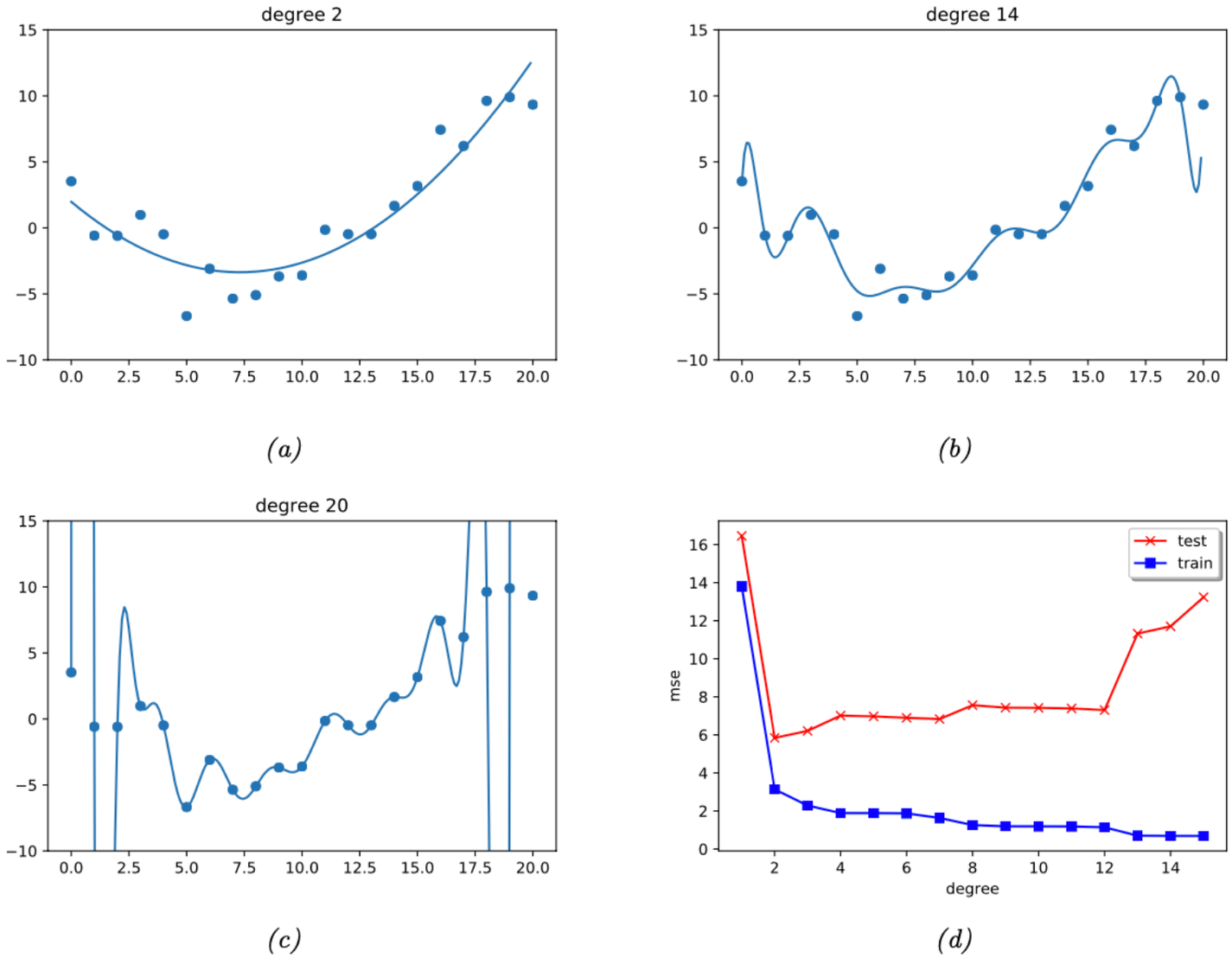


Figure 1.7: (a-c) Polynomials of degrees 2, 14 and 20 fit to 21 datapoints (the same data as in Figure 1.5). (d) MSE vs degree. Generated by code at figures.probml.ai/book1/1.7.

Supervised Learning

No free lunch (NFL) theorem

- No free lunch theorem: there is no single best model that works optimally for all kinds of problems.

- It is important to have many models and algorithms so we can choose the best model from them.

- A good model should have small sample complexity for many distributions p^*.

- Sample complexity: the number of training-samples that it needs in order to successfully learn a target function.

Unsupervised Learning

Clustering

- What is a cluster?
  - Mode by the mean shift algorithm
  - Level set by density-based clustering

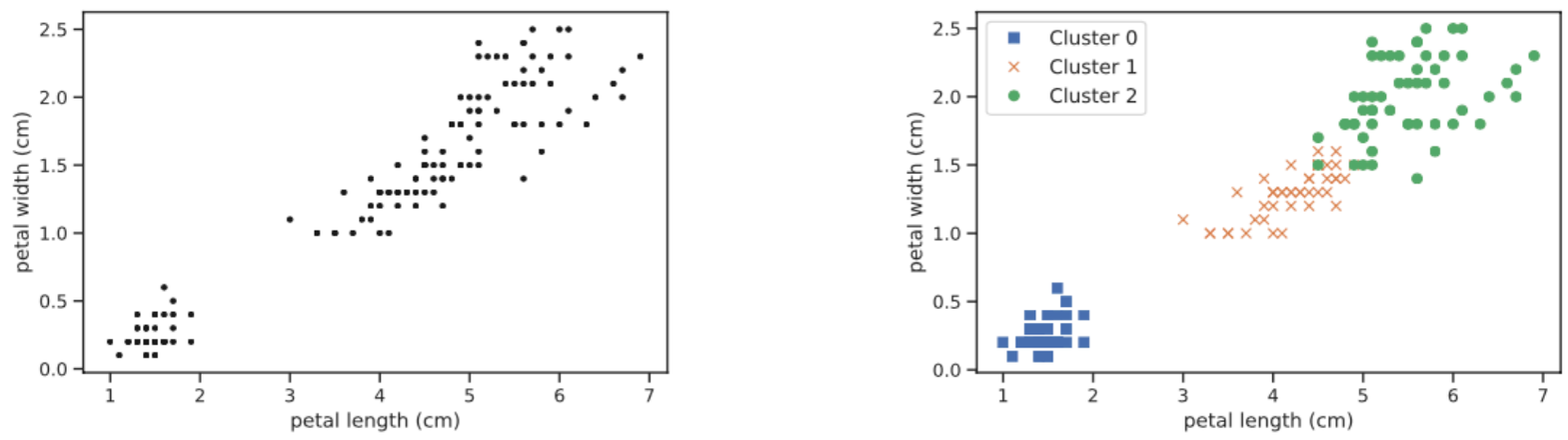


Figure 1.8: (a) A scatterplot of the petal features from the iris dataset. (b) The result of unsupervised clustering using $K = 3$. Generated by code at figures.probml.ai/book1/1.8.

Unsupervised Learning

The curse of dimensionality

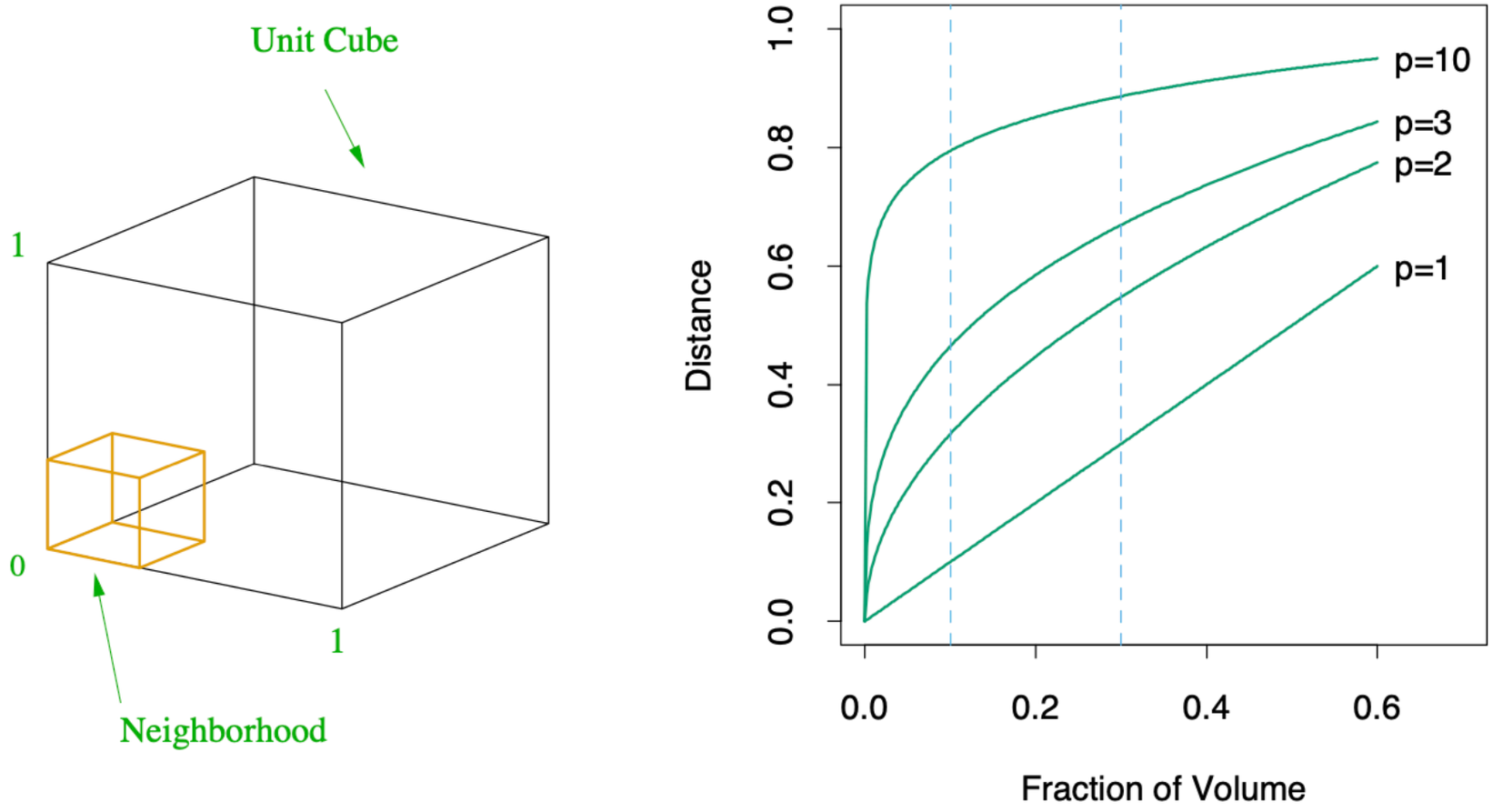
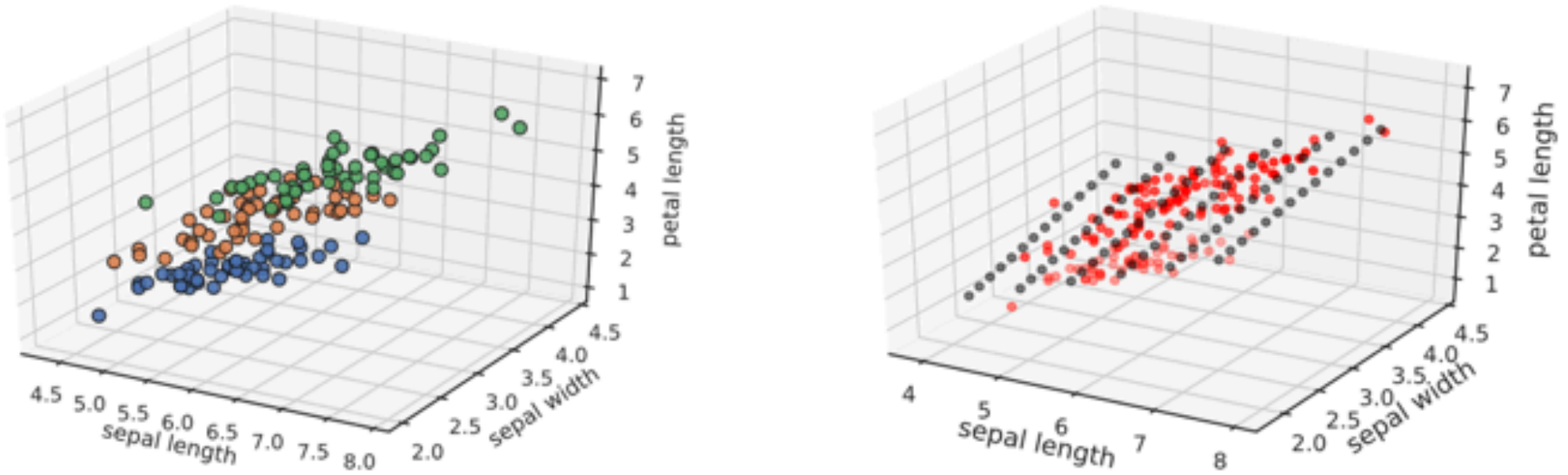


FIGURE 2.6. *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover* 80% *of the range of each coordinate to capture* 10% *of the data.*

Unsupervised Learning

Evaluating unsupervised learning



(a) (b)

Figure 1.9: (a) Scatterplot of iris data (first 3 features). Points are color coded by class. (b) We fit a 2d linear subspace to the 3d data using PCA. The class labels are ignored. Red dots are the original data, black dots are points generated from the model using $\hat{\boldsymbol{x}} = \mathbf{W}\boldsymbol{z} + \boldsymbol{\mu}$*, where* $\boldsymbol{z}$ *are latent points on the underlying inferred 2d linear manifold. Generated by code at figures.probml.ai/book1/1.9.*

Unsupervised Learning

Evaluating unsupervised learning

- A common method for evaluating unsupervised models is to measure the probability assigned by the model to unseen test examples.
- The negative log likelihood of the data:

$$\mathscr{L}(\theta, \mathscr{D}) = -\frac{1}{|\mathscr{D}|} \sum_{x \in \mathscr{D}} \log p(x \,|\, \theta).$$

Reinforcement Learning

- Alpha Go!
- The **agent** has to learn how to interact with its environment.
- The difference from supervised learning is that the agent receives an *occasional* **reward**.

(a) (b)

Figure 1.10: Examples of some control problems. (a) Space Invaders Atari game. From https://gym.openai.com/envs/SpaceInvaders-v0/. (b) Controlling a humanoid robot in the MuJuCo simulator so it walks as fast as possible without falling over. From https://gym.openai.com/envs/Humanoid-v2/.

Statistics vs Machine Learning

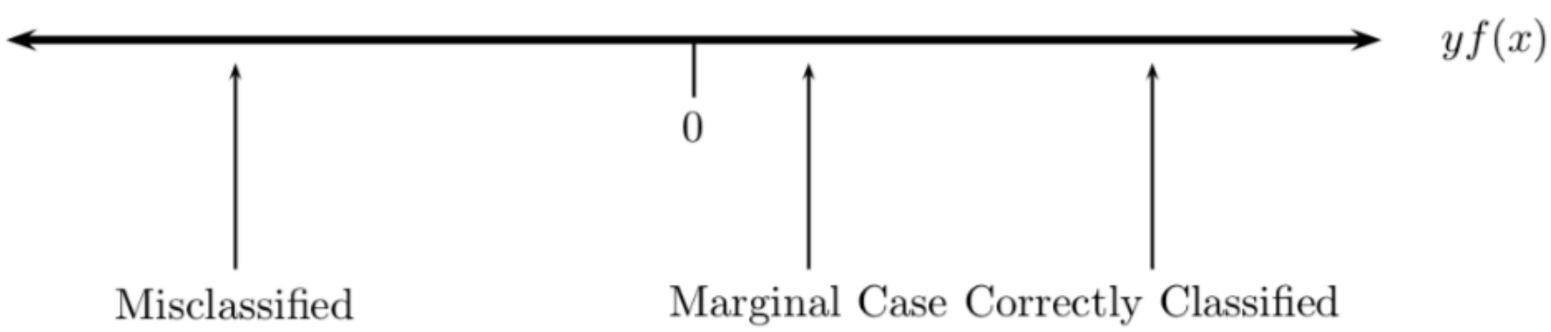
Statistics	Computer Science	Meaning
estimation	learning	using data to estimate an unknown quantity
classification	supervised learning	predicting a discrete Y from $X \in \mathcal{X}$
clustering	unsupervised learning	putting data into groups
data	training sample	$(X_1, Y_1), \ldots, (X_n, Y_n)$
covariates	features	the X_i's
classifier	hypothesis	a map from covariates to outcomes
hypothesis	—	subset of a parameter space Θ
confidence interval	—	interval that contains unknown quantity with a prescribed frequency
directed acyclic graph	Bayes net	multivariate distribution with specified conditional independence relations
Bayesian inference	Bayesian inference	statistical methods for using data to update subjective beliefs
frequentist inference	—	statistical methods for producing point estimates and confidence intervals with guarantees on frequency behavior
large deviation bounds	PAC learning	uniform bounds on probability of errors

Think Statistically!

- What does it mean for one classifier to be better than another?
- Why is one classifier better than another?
- Why do some prediction methods work well in certain high dimensional problem?
- What is the role of the *margin* or the hard to classify cases?
- What are kernel methods? How do they relate to older methods?
- How do we pick tuning parameters in prediction algorithm?
- Which is more important, choosing a good prediction algorithm or choosing the tuning parameters within a given algorithm?

Case Study I: The Margin

- In classification problems, the probability of a misclassification is $R = \Pr(\text{sign}(f(X) \neq Y)) = \Pr(Yf(X) < 0)$.
- Here $Y \in \{-1, 1\}$ is a binary response variable and $f(X)$ is a function of a covariate, or feature X.
- The function $yf(x)$ is called the margin. If the margin is small, then we have a difficult classification problem.



Tsybakov noise condition

- The behavior at the margin is quantified by the Tsybakov noise condition $\Pr(|m(x) - 1/2| \leq t) \leq Ct^{\alpha}$.
- Here $m(x) = \mathbb{E}(Y|X = x)$ is the regression function and *Y*=1 if $m(x) > 1/2$, otherwise *Y*=-1.
- If α is large, then the decision boundary $\{x : m(x) = 1/2\}$is well defined!
- Sometimes the assumption is more important than the choice of method in analyzing data.

Case Study II: Kernels

- Suppose we observe $(X_1, Y_1), \ldots, (X_n, Y_n)$ where $Y_i \in \{0,1\}$.
- Define a classifier as follows:

$$h(X) = \begin{cases} 1 \text{ if } \|X - \bar{X}_1\| \leq \|X - \bar{X}_0\|, \\ 0 \text{ if } \|X - \bar{X}_1\| > \|X - \bar{X}_0\|. \end{cases}$$

- Here $\bar{X}_k$ is the average of X_i with $Y_i = k$.
- We can improve the above classifier with transformation ϕ.
- Define the kernel $k(x, z) = < \phi(x), \phi(z) >$.

Case Study II: Kernels

- With a kernel trick, we can define a new classifier

$$h(X) = \begin{cases} 1 \text{ if } \hat{p}_1(X) \geq \hat{p}_0 + c, \\ 0 \text{ if } \hat{p}_1(X) < \hat{p}_0 + c\,. \end{cases}$$

where $\hat{p}_k(x) = \sum_{i \in I_k} k(x, X_j)/|I_k|$ and $I_k = \{i : Y_i = k\}$.

- A common choice of the kernel is $k(x, z) = \exp(-\|x - z\|/2\sigma^2)$
- In this case, the above classifier is equivalent to LDA.
- Furthermore, there is a hidden tuning parameter σ in the Gaussian kernel.