Statistical Learning – week 3.1

Joris Bierkens

Delft University of Technology, The Netherlands

15 February 2024

Outline

Course organization

- Course in Q3 + Q4, 6 ECTS
- Lecturer Joris Bierkens, assisted by Chris van Vliet
- Office hour Chris: Monday 13.45-14.45 (except week 3.3, 3.5, 3.6)
- Use the Brightspace forum for course communication
- Communicate your questions and comments publicly on Brightspace when possible
  - avoids repeated questions
  - helps other students
- Final grade: $0.3 \times A + 0.7 \times E$, where
  - A is the average grade for your assignments, and
  - E is your exam grade;
  - both A and E should be sufficient (≥ 5.8) to pass the course.
- Exam: 27 June, 9:00-12:00.
- Resit: 18 July, 9:00-12:00

Assignments

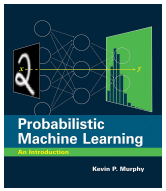
- There will be two assignments in total.
- These will consist of exercises given after class.
- Assignment deadlines: just before the lectures of weeks 3.6 and 4.6.
- Work together (meet up!) in groups of three
- Self-enroll in groups on Brightspace
- In your work:
  - clearly show your intermediate steps,
  - motivate your answer,
  - be to the point.
- Prepare clearly legible handwritten work (scanned, e.g. using CamScanner) or LaTeX.
- Submit using Brightspace by the deadline as a single PDF.
- Not adhering to these guidelines will result in a reduced grade.

Programming

- We will use Python for programming exercises and assignments in this course.
- A background in Python at the level of the course AM1090 is assumed.
- This includes a basic familiarity with the NumPy and MatPlotLib packages.
- To give you an idea of that course, see
  - the book Think Python, `https://greenteapress.com/wp/think-python/`
  - the slides of AM1090 (available on Brightspace).

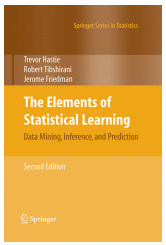
Study material

Main reference

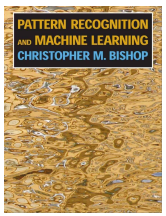


Kevin P. Murphy, Probabilistic Machine Learning - An Introduction, `https://probml.github.io/pml-book/book1.html`

Alternative references



Hastie et al., The Elements of Statistical Learning, 2nd ed., `https://web.stanford.edu/~hastie/ElemStatLearn/`



Bishop, Pattern Recognition and Machine Learning, `https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/`

Lecture notes, will keep track of the lectures; available on Brightspace

What is statistical learning?

- In statistical learning we are interested in discovering relations in high-dimensional and/or large data sets.
- Close relation with machine learning and statistics
  - Compared to machine learning, statistical learning has a more mathematical/statistical focus: study methods as well as underlying theory, with attention for quantification of uncertainty.
  - Compared to classical statistics, the focus is more on computational aspects, large data sets, high-dimensional models and model-free or non-parametric approaches.

Supervised learning vs unsupervised learning

- supervised learning: learning a function $y = f(x)$ or conditional distribution $p(y \mid x)$ based on observed inputs $x_1, \ldots, x_n$ in $\mathbb{R}^d$ and associated outputs $y_1, \ldots, y_n$.
- unsupervised learning: learning a probability distribution $p(x)$ based on observed inputs $x_1, \ldots, x_n$.

Terminology

$x_i \in \mathcal{X}$	$y_i \in \mathcal{Y}$
input	output
independent variable	dependent variable
predictor	outcome
explanatory variable	response variable

- In regression we assume that y assumes continuous values in $\mathcal{Y} \subset \mathbb{R}$.
- In classification we assume that y assumes discrete values in a finite set $\mathcal{Y} \subset \mathbb{R}$.

Regression example: Prostate cancer

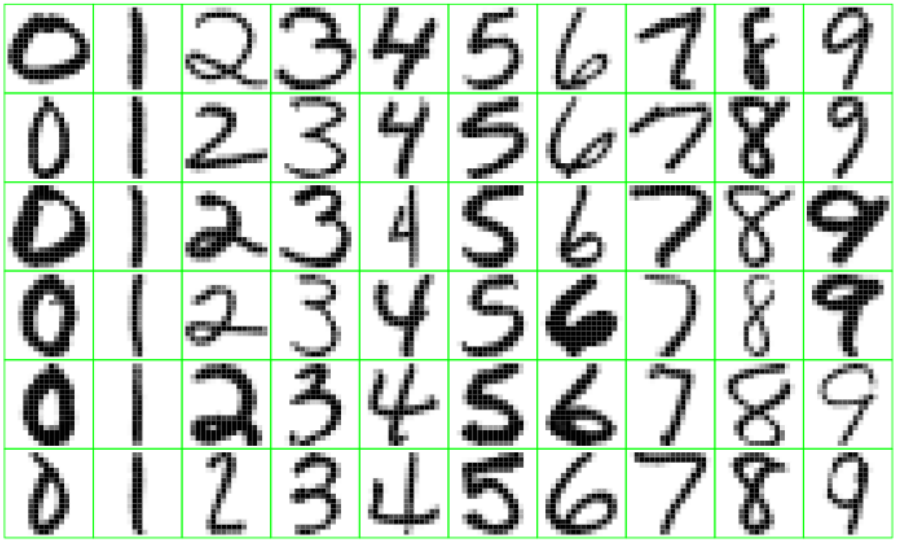
[Hastie et al., Figure 1.1]

response variable (y): `lpsa`

predictors (x_i): `lcavol`, ... , `pgg45`

Classification example: Handwritten digits

[Hastie et al., Figure 1.2]



response variable (y): digit classification `0, 1, ..., 9`

predictors (x_i): pixel value at each position

Probabilistic framework for supervised learning

- Interpret $D_n = \{(X_1, Y_1), \ldots (X_n, Y_n)\}$ as independent realizations of a random variable (X, Y) in $\mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^d \times \mathbb{R}$, with joint probability distribution P.
- In supervised learning we are mostly interested in learning
  - a functional relation between $y = f(x)$, or
  - the conditional distribution $P(Y \in \cdot \mid X)$ of y conditional on X.
- Distribution of X may also be relevant:
  - In practice: for example to design good features, i.e. summaries $\phi(x)$ of the data.
  - In theory: for example to analyze the 'typical' prediction error.

Population model

- The population model $P(dx, dy)$ is a probability distribution over $\mathcal{X} \times \mathcal{Y}$: it takes sets as arguments and satisfies the axioms of probability theory.
- Notation: $P(A, B) = \mathbb{P}(X \in A, Y \in B)$.
- We should assume the population model to be unknown.
- In regression, we assume that P has a probability density function

$$P(A, B) = \int_{x \in A} \int_{y \in B} p(x) p(y \mid x) \, dy \, dx$$

- In classification, we assume a conditional probability mass function $p(y \mid x)$ for $y \in \mathcal{Y}$ and a continuous density $p(x)$, so

$$P(A, B) = \int_{x \in A} \sum_{y \in B} p(y \mid x) p(x) \, dx.$$

Example: Additive noise model

$$Y = f(X) + \varepsilon, \quad \text{where}$$

- a true function $f : \mathbb{R}^d \to \mathbb{R}$, and
- a random variable ε in $\mathbb{R}$, independent of X, with mean zero and variance σ^2.
- The conditional distribution $P(dy \mid x)$ is given by

$$P(A \mid x) = \mathbb{P}(f(x) + \varepsilon \in A).$$

Example: Gaussian noise

In the Gaussian case, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, so we obtain the conditional density

$$p(y \mid x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-(y - f(x))^2/(2\sigma^2)\right).$$

Together with $p(x)$, this specifies the joint model for (X, Y).

Example: Class probabilities

- Classification: outcomes in a finite set $\mathcal{Y}$.
- A generative model for classification could be:
  - class-conditional densities $p(x \mid y)$ for $y \in \mathcal{Y}$, and
  - prior class probabilities $p(y) = \pi_y$.

  Together $p(x \mid y)$ and $p(y)$ specify the joint model $p(x, y) = p(x \mid y)p(y) = \pi_y p(x \mid y)$.
- We can determine the posterior class probabilities $p(y \mid x)$ using the Bayes formula,

$$p(y \mid x) = \frac{p(x, y)}{p(x)} = \frac{p(x \mid y)\pi_y}{\sum_{y' \in \mathcal{Y}} p(x \mid y')\pi_{y'}}.$$

Aims of statistical learning

Given observations $D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, we may want to

1. Estimate a predictive distribution: Determine $\hat{P}(dy \mid x) = \hat{P}(dy \mid x; D_n)$ as an approximation to $P(dy \mid x)$
2. Learn a predictive function: Determine a function $\hat{f}(x) = \hat{f}(x; D_n)$ as an approximation to a 'true function'.

   Note, if we can do (1), a possible approach to (2) is taking

$$\hat{f}(x) = \int y \, \hat{p}(y \mid x) \, dy \quad \text{or} \quad \hat{f}(x) \in \arg\max_y \hat{p}(y \mid x).$$

3. Unsupervised learning: Determine $\hat{p}(x)$ as an approximation of $p(x)$.

Estimation

- Suppose we observe data $D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$.
- Based on this data, we may 'guess' an estimator $\hat{f}(x) = \hat{f}(x; D_n)$ for the function $f(x)$
- This is a random, infinite dimensional object:
  - random, since it depends on the data D_n that we model as being random
  - infinite-dimensional, since it assigns an outcome $\hat{f}(x)$ to every input $x \in \mathbb{R}^d$
- What is the quality of our estimation/prediction?

Loss functions

Suppose we make a prediction $y' \in \mathcal{Y}$ when the true outcome is $y \in \mathcal{Y}$

More generally, suppose we choose an action $a \in \mathcal{A}$ that we wish to compare to a true outcome y.

A loss function is a mapping $L : \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$, where $L(y, a)$ measures the 'loss' we incur when we choose action a when the true outcome is y.

Examples

- Quadratic loss: $L(y, y') = (y - y')^2$, where $\mathcal{Y} = \mathcal{A} = \mathbb{R}$.
- 0-1 loss: $L(y, y') = \mathbb{1}_{y \neq y'} = \begin{cases} 1, & y \neq y', \\ 0, & y = y' \end{cases}$ where $\mathcal{Y} = \mathcal{A} = \{1, \ldots, K\}$.
- Logistic loss: when p is a (predictive) probability of y,

$$L(y, p) = -y \log p - (1 - y) \log(1 - p), \quad \mathcal{Y} = \{0, 1\}, \mathcal{A} = (0, 1).$$

Risk

- Consider a loss function : $\mathcal{Y} \times \mathcal{A}$ and a function $f : \mathcal{X} \to \mathcal{A}$.
- The (population) risk of a function $f : \mathcal{X} \to \mathcal{A}$ is the quantity

$$R[f] = \mathbb{E}_P[L(Y, f(X))].$$

- The conditional risk is given by

$$R[f](x) = \mathbb{E}_P[L(Y, f(X)) \mid X = x].$$

- The Bayes estimator is a function f^* that minimizes $R[f]$, or equivalently $R[f](x)$ for all x.

Mathematical intermezzo: conditional expectation

Conditional expectation

The conditional expectation $\mathbb{E}[Y \mid X]$ is defined as the random variable $h(X)$ which satisfies

$$\mathbb{E}[Yg(X)] = \mathbb{E}[h(X)g(X)]$$

for all functions g. This random variable exists and is a.s. uniquely defined. We then write $\mathbb{E}[Y \mid X = x] := h(x)$.

Exercise

Show that the function f minimizing

$$\mathbb{E}[(Y - f(X))^2]$$

is given by the conditional expectation $f(x) = \mathbb{E}[Y \mid X = x]$. This is called the projection property of conditional expectation.

In particular the Bayes estimator for quadratic loss is given by $f(x) = \mathbb{E}[Y \mid X = x]$.

Empirical Risk Minimization

We only have access to $D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, so cannot compute the Bayes estimator!

Instead let us minimize the Empirical Risk

$$R_n[f] = \mathbb{E}_{P_n}[L] = \frac{1}{n} \sum_{i=1}^{n} L(Y_i, f(X_i)),$$

where $P_n(dx, dy) = \frac{1}{n} \sum_{i=1}^{n} \delta_{(X_i, Y_i)}(dx, dy)$ denotes the empirical distribution function.

Example: Residual Sum of Squares

For quadratic loss, we have

$$R_n[f] = \frac{1}{n} \sum_{i=1}^{n} (Y_i - f(X_i))^2 =: \frac{1}{n} \mathrm{RSS}[f],$$

where RSS is abbreviation for Residual Sum of Squares

Expected Risk

- For an estimator $\hat{f}(x) = \hat{f}(x; D_n)$, the risk $R[\hat{f}]$ is random (why?).
- The Expected Risk considers the expectation with respect to the data,

$$\mathrm{ER}[\hat{f}] = \mathbb{E}_{D_n} R[\hat{f}] = \mathbb{E}_{D_n} \mathbb{E}_P L(Y, \hat{f}(X; D_n)).$$

- Often we are interested in conditional expected risk

$$\mathrm{ER}[\hat{f}](x) = \mathbb{E}_{D_n} \mathbb{E}_P \left[L(Y, \hat{f}(x; D_n) \mid X = x \right].$$

Example: Expected Prediction Error

For quadratic loss, the expected risk $\mathrm{ER}[\hat{f}]$ is also known as expected prediction error (EPE). So for example, conditionally on X, X_i,

$$\mathrm{EPE}[\hat{f}](x) = \mathbb{E}_{D_n} \left[\mathbb{E}_P [(Y - \hat{f}(x))^2 \mid X = x] \right].$$

Mean Squared Error and Expected Prediction Error

Recall from elementary statistics that if T is an estimator for a quantity θ, then the Mean Squared Error is given by

$$\mathrm{MSE}(T;\theta) = \mathbb{E}[(T-\theta)^2] = \underbrace{(\mathbb{E}[T-\theta])^2}_{\mathrm{bias}(T;\theta)^2} + \mathrm{Var}(T).$$

Consider the additive noise model,

$$Y = f(X) + \varepsilon$$

and suppose $\hat{f}(x) = \hat{f}(x; D_n)$ is an estimator of f based on the data D_n.

Expected prediction error (EPE)

$$\begin{aligned}\mathrm{EPE}[\hat{f}](x) &= \mathbb{E}_{D_n}\mathbb{E}_P[(Y-\hat{f}(x))^2 \mid X = x] = \mathrm{MSE}[\hat{f}(x); f(x)] + \mathrm{Var}(\varepsilon) \\ &= \mathrm{bias}^2 + \mathrm{variance} + \mathrm{noise}.\end{aligned}$$

k-nearest neighbours (kNN)

- consider a distance measure $\rho(x, x')$ on $\mathbb{R}^d$, e.g.,

$$\rho(x, x') = \|x - x'\| = \|x - x'\|_2 = \left(\sum_{i=1}^{d} (x_i - x_i')^2 \right)^{1/2}.$$

- data set $D_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$.
- let $N_k(x)$ denote the set of k indices of x_i with the smallest distance $\rho(x_i, x)$ to x.
- for regression, define

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i.$$

- for classification, define $\hat{f}(x)$ to be the majority vote

$$\hat{f}(x) \in \arg\max_{y \in \mathcal{Y}} |\{i \in N_k(x) : y_i = y\}|.$$

- this is a non-parametric approach: there is no finite dimensional parameter indexing the possible functions $\mathcal{F}$.

k-nearest neighbours (kNN) : MSE

- for regression,

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i.$$

- assume the additive noise model

$$y = f(x) + \varepsilon$$

  where $\mathrm{Var}(\varepsilon) = \sigma^2$.

- bias-variance trade-off, assuming $x_1, \ldots, x_n, x$ fixed (exercise) :

$$\mathrm{MSE}(\hat{f}(x); f(x)) = \left(\underbrace{\frac{1}{k} \sum_{i \in N_k(x)} f(x_i) - f(x)}_{\text{bias}} \right)^2 + \underbrace{\frac{\sigma^2}{k}}_{\text{variance}}.$$

- as k grows,
  - the bias (typically) increases,
  - the variance decreases.

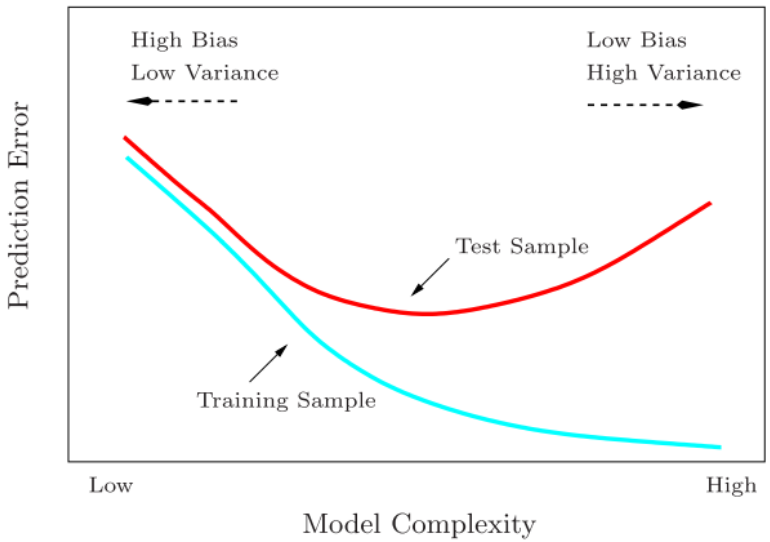
Bias-variance trade-off and overfitting

For k-nearest neighbours we had seen the bias-variance trade-off

$$\mathrm{MSE}(\hat{f}(x); f(x)) = \left(\frac{1}{k} \sum_{i \in N_k(x)} f(x_i) - f(x) \right)^2 + \frac{\sigma^2}{k}.$$

- bias-variance trade-off is a very general phenomenon.
- complex model – non-smooth fit – possible overfitting – larger variance
- simple model – smooth fit – possible underfitting – larger bias

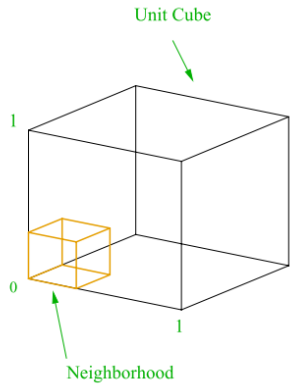
- prevent overfitting by assessing performance on a test sample.



Curse of dimensionality

See [HTF09, Section 2.5]

- a local method (e.g., nearest neighbours) works well if any new input x has many observations $x_1, \ldots, x_n$ in its vicinity.



- suppose inputs $x_1, \ldots, x_n$ have uniform distribution in the hypercube $[0, 1]^p$.
- how many points will lie in the sub-hypercube $[0, 0.1]^p$?
- answer: approximately $n \times (0.1)^p$.
- in order to maintain a fixed ratio of points in any sub-hypercube for growing p, we require n to grow exponentially in p.

Learning objectives lecture 3.1

- Key distinctions: supervised vs unsupervised learning, regression vs classification (G)
- k-nearest neighbours as a simple example of supervised learning
- Probabilistic setting of supervised learning (G)
- Mean squared error, bias-variance trade-off (G)
- The use of training- and test-set to estimate risk (G)
- Curse of dimensionality (G)

Reading and exercises

Background reading

- draft lecture notes, Chapter 1
- Murphy, Sections 1.1-1.3, 1.6

Recommended exercises

- Exercises in lecture notes, Chapter 1
- Exercises on slides