Statistical Learning, MESIO UPC-UB

Chapter 1. Introduction.

Introduction

Pedro Delicado

Deptartament d'Estadística i Investigació Operativa, UPC

1. Instructors

2. Machine learning and statistical learning

3. Supervised and unsupervised learning

3.1 Supervised Learning (the prediction problem)

3.1.1 The regression problem

3.1.2 The classification problem

1. Instructors

2. Machine learning and statistical learning

3. Supervised and unsupervised learning
3.1 Supervised Learning (the prediction problem)
3.1.1 The regression problem
3.1.2 The classification problem

Instructors

Pedro Delicado

Departament d'Estadística i Investigació Operativa, UPC.

Research topics: Nonparametric estimation, Functional Data Analysis, dimensionality reduction, distance based prediction, bridging the gap between Statistics and Data Science.

Ferran Reverter

Departament de Genètica, Microbiologia i Estadística, UB.

Research topics: Interpretability of Machinel Learning Kernel based models, and their applications in Biosciences. He participates in the Genomics and Bioinformatics program at CRG.

Esteban Vegas

Departament de Genètica, Microbiologia i Estadística, UB.

Research topics: Visualization, analysis and integration of different sources of information (e.g., omic data) with machine learning techniques, and particularly Kernel based methods.

1. Instructors

2. Machine learning and statistical learning

3. Supervised and unsupervised learning
3.1 Supervised Learning (the prediction problem)
3.1.1 The regression problem
3.1.2 The classification problem

Statistics, Machine Learning, Data Science

- **Statistics** *is the science of learning from experience* (Efron and Hastie 2016).
- *We define* **Machine Learning** *as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data* (Murphy 2012).
- From Baumer, Kaplan, and Horton (2017):
  - **Statistics** *can be defined as the science of learning from data.*
  - *We see [***Data Science***] as the science of extracting meaningful information from data.*
  - *Many statisticians will say:* **"But we already have a field for that: it's called Statistics!"**
- A common goal: **Learning from data**.

[1]

- *Much of statistical technique was originally developed in an environment where data were scarce and difficult or expensive to collect, so statisticians focused on creating methods that would maximize the strength of inference one is able to make, given the least amount of data* (Baumer, Kaplan, and Horton 2017).
- *Much of the development of statistical theory was to find mathematical approximations for things that we couldn't yet compute* (Baumer, Kaplan, and Horton 2017).
- Mathematics was the best computer (Efron and Hastie 2016).
- *[F]rom the 1950s to the present is the "computer age" of [Statistics], the time when computation, the traditional bottleneck of statistical applications, became faster and easier by a factor of a million* (Efron and Hastie 2016).

[1] Slides with colored background are optional material.

- *The center of the field [Statistics] has in fact moved in the past sixty years, from its traditional home in mathematics and logic toward a more computational focus* (Efron and Hastie 2016).
- *Today, the manner in which we extract meaning from data is different in two ways, both due primarily to advances in computing:*
  - *we are able to compute many more things than we could before, and;*
  - *we have a lot more data than we had before.*

  (Baumer, Kaplan, and Horton 2017)

- Michael Jordan (UC Berkeley) has described **Data Science** as **the marriage of computational thinking and inferential thinking. [...] These styles of thinking support each other**.
- **Data Science** is a science, a rigorous discipline combining elements of **Statistics** and **Computer Science**, with roots in **Mathematics**.

From Efron and Hastie (2016):

- *A particularly energetic brand of the statistical enterprise has flourished in the new century, data science, emphasizing algorithmic thinking rather than its inferential justification.*
- *Data Science [...] seems to represent a statistics discipline without parametric probability models or formal inference.*
- *[Machine Learning] large-scale prediction algorithms (neural nets, deep learning, boosting, random forests, and support-vector machines) are the media stars of the big-data era.*
- *Why have they taken center stage?*
  - **Prediction is commercially valuable**.
  - *Prediction is the simplest use of regression theory.*
  - *It can be carried out successfully without probability models, perhaps with the assistance of cross-validation, permutations, bootstrap.*

Different focus of Statistics and Machine Learning:

	Asymptotics, optimality	Interpretability	Accurate prediction	Scalability
Statistics	XXXXX	XXXXX	XX	X
Machine Learning	X	XX	XXXXX	XXXXX

Statistical Learning, Machine Learning (I)

- This course is about learning from data, and specifically focused on the prediction problem.
- We are studying models, tools and techniques originally developed by statisticians (sparse estimation of linear regression models and generalized linear models, nonparametric versions of them, classification and regression trees, ...)
- Nowadays they are part of the toolkit of Machine Learning practitioners (usually coming from Computer Science) and Data Scientists in general.
- We also will learn algorithms and procedures that has been proposed by researchers in Machine Learning (neural networks, support vector machines, boosting, random forests, ...) that now are part of the statisticians' prediction toolkit.
- *Since computation plays a key role [in learning from data], it is not surprising that much of the new development has been done by researchers in other fields such as computer science and engineering* (Hastie, Tibshirani, and Friedman 2009).

Statistical Learning, Machine Learning (II)

- Which would have been the right name for this course?
  - Learning from data.
  - Machine Learning.
  - Statistical Learning.
- We the instructors are statisticians. So this course will necessarily be different from a Machine Learning course in a Computer Science MSc program, even if the contents of both could have a large intersection.
- We feel comfortable with **Statistical Learning**, a term that started to be popular from the publication in 2001 of the first edition of *The Elements of Statistical Learning*, with a second edition in 2009 (Hastie, Tibshirani, and Friedman 2009).
- See also James, Witten, Hastie, and Tibshirani (2013) and Hastie, Tibshirani, and Wainwright (2015).

1. Instructors

2. Machine learning and statistical learning

3. Supervised and unsupervised learning
3.1 Supervised Learning (the prediction problem)
3.1.1 The regression problem
3.1.2 The classification problem

Supervised and unsupervised learning

References:

Section 14.1 in Hastie, Tibshirani, and Friedman (2009)

Supervised and unsupervised learning (I)

- **Supervised learning** (prediction problem):
  - Regression: Predicting of a quantitative response.
  - Classification (or discriminant analysis): Predicting a qualitative variable.
- Probabilistic model:
  - Response variable Y.
    Explanatory variables (features) $\boldsymbol{X} = (X_1, \ldots, X_p)$.
  - Data $(\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip}), y_i)$, $i = 1, \ldots, n$ i.i.d. from the random variable
    $$(\boldsymbol{X} = (X_1, \ldots, X_p), Y) \sim \Pr(\boldsymbol{X}, Y).$$
  - $\Pr(\boldsymbol{X}, Y)$ denotes the joint distribution of $\boldsymbol{X}$ and Y.
  - When this joint distribution is continuous, $\Pr(\boldsymbol{X}, Y)$ is the joint probability density function.

- Main interest in the conditional distribution $\Pr(Y|\boldsymbol{X})$ and specifically in a conditional location parameter,

$$\mu(\boldsymbol{x}) = \underset{\mu}{\operatorname{argmin}}\, \mathbb{E}(L(Y,\mu)|\boldsymbol{X}=\boldsymbol{x}),$$

  where $L(y,\hat{y})$ is a loss function, measuring the error of predicting y with $\hat{y}$.
- For quadratic loss, $L(y,\hat{y}) = (y-\hat{y})^2$, $\mu(\boldsymbol{x})$ is the regression function:

$$\mu(\boldsymbol{x}) = \mathbb{E}(Y|\boldsymbol{X}=\boldsymbol{x}).$$

Supervised and unsupervised learning (II)

- **Unsupervised learning:** to learn relationships and structure from the observed data.
- Probabilistic model:
  - Variables of interest: $\boldsymbol{X} = (X_1, \ldots, X_p)$.
  - Data $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})$, $i = 1, \ldots, n$ i.i.d. from the random variable

$$\boldsymbol{X} = (X_1, \ldots, X_p) \sim \Pr(\boldsymbol{X}).$$

  - $\Pr(\boldsymbol{X})$ denotes the probability distribution of $\boldsymbol{X}$.
  - When this distribution is continuous, $\Pr(\boldsymbol{X})$ is the probability density function of $\boldsymbol{X}$.
- Main interest: To infer properties of $\Pr(\boldsymbol{X})$.

- Specific problems in unsupervised learning:
  - Estimating directly the density function $\Pr(\boldsymbol{x})$:
    Density estimation (histogram, kernel density estimation, Gaussian mixture models, ...)
  - Detecting homogeneous subpopulations:

$$\Pr(\boldsymbol{x}) = \sum_{j=1}^{k} \alpha_j \Pr(\boldsymbol{x}|C_j),\ \alpha_j \geq 0,\ \sum_j \alpha_j = 1.$$

    Clustering (hierarchical clustering, k-means, ...)
  - Finding low-dimensional hyper-planes or hyper-surfaces (manifolds) in $\mathbb{R}^p$ around which the probability $\Pr(\boldsymbol{x})$ is concentrated.
    Dimensionality reduction (PCA, MDS, principal curves, ISOMAP, manifold learning, ...)
  - Proposing generative probabilistic models for $\boldsymbol{X}$, depending on low-dimensional unobservable random variables $\boldsymbol{F}$.
    Extraction of latent variables (Factor Analysis, ...)
  - ...

1. Instructors

2. Machine learning and statistical learning

3. Supervised and unsupervised learning
3.1 Supervised Learning (the prediction problem)
3.1.1 The regression problem
3.1.2 The classification problem

Supervised Learning (the prediction problem)

- Let $(\boldsymbol{X}, Y)$ be a r.v. with support $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^p \times \mathbb{R}$.
- General supervised learning or prediction problem:
  - Training sample: $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, i.i.d. from $(\boldsymbol{X}, Y)$.
  - The goal is to define a function (possibly depending on the sample) $h_S : \mathcal{X} \mapsto \mathcal{Y}$ such that for a new independent observation $(\boldsymbol{x}_{n+1}, y_{n+1})$, from which we only know $\boldsymbol{x}_{n+1}$, it happens that

$$\hat{y}_{n+1} = h_S(\boldsymbol{x}_{n+1}) \text{ is close to } y_{n+1} \text{ (in some sense).}$$

  - Function h_S is called generically prediction function (or classification function or regression function, depending on the case).
- We say that we have a problem of binary classification (or discrimination) when $\mathcal{Y} = \{0, 1\}$ (you can also use $\mathcal{Y} = \{-1, 1\}$).
- The problem of classification in K classes arises when $\mathcal{Y} = \{1, \ldots, K\}$ (or $\mathcal{Y} = \left\{\boldsymbol{y} \in \{0,1\}^K : \sum_{k=1}^K y_k = 1\right\}$).
- When $\mathcal{Y} \subseteq \mathbb{R}$ (or $\mathcal{Y}$ is an interval) we have a standard regression problem.

Statistical Decision Theory

- We will write the prediction problem as a decision problem.
- Let $(\boldsymbol{X}, Y)$ be a r.v. with support $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^p \times \mathbb{R}$.
- **Prediction problem:** To look for a prediction function $h : \mathcal{X} \mapsto \mathcal{Y}$ such that $h(\boldsymbol{X})$ is close to Y in some sense.
- The (lack of) closeness between $h(\boldsymbol{X})$ and Y is usually measured by a **loss function** $L(Y, h(\boldsymbol{X}))$.
- For instance, the *squared error loss* is $L(Y, h(\boldsymbol{X})) = (Y - h(\boldsymbol{X}))^2$.
- $L(Y, h(\boldsymbol{X}))$ is a r.v., with expected value $\mathrm{EL}(h) = \mathbb{E}(L(Y, h(\boldsymbol{X})))$, called **expected loss**, that only depends on h.
- **Decision problem:** To find the prediction function $h : \mathcal{X} \mapsto \mathcal{Y}$ that minimizes the expected loss.

Bayes rule

Denote by $\mathrm{Pr}_{(\boldsymbol{X},Y)}(\boldsymbol{x},y)$ the joint probability distribution of $(\boldsymbol{X},Y)$. Observe that, for any $h:\mathcal{X}\mapsto\mathcal{Y}$,

$$\mathrm{EL}(h)=\mathbb{E}(L(Y,h(\boldsymbol{X})))=\int_{\mathcal{X}\times\mathcal{Y}}L(y,h(\boldsymbol{x}))d\mathrm{Pr}_{(\boldsymbol{X},Y)}(\boldsymbol{x},y)$$

$$=\int_{\mathcal{X}}\left(\int_{\mathcal{Y}}L(y,h(\boldsymbol{x}))d\mathrm{Pr}_{Y|\boldsymbol{X}=\boldsymbol{x}}(y)\right)d\mathrm{Pr}_{\boldsymbol{X}}(\boldsymbol{x})$$

$$=\int_{\mathcal{X}}\mathbb{E}(L(Y,h(\boldsymbol{x}))|\boldsymbol{X}=\boldsymbol{x})\,d\mathrm{Pr}_{\boldsymbol{X}}(\boldsymbol{x})$$

$$\geq\int_{\mathcal{X}}\min_{y\in\mathcal{Y}}\mathbb{E}(L(Y,y)|\boldsymbol{X}=\boldsymbol{x})\,d\mathrm{Pr}_{\boldsymbol{X}}(\boldsymbol{x})=\mathrm{EL}(h_B).$$

It follows that the optimal prediction function is **the Bayes rule**

$$h_B(\boldsymbol{x})=\arg\min_{y\in\mathcal{Y}}\mathbb{E}(L(Y,y)|\boldsymbol{X}=\boldsymbol{x}).$$

1. Instructors

2. Machine learning and statistical learning

3. Supervised and unsupervised learning
3.1 Supervised Learning (the prediction problem)
3.1.1 The regression problem
3.1.2 The classification problem

The regression problem

- Let $(\boldsymbol{X}, Y)$ be a $(p+1)$-dimensional random variable, with $Y \in \mathbb{R}$ (or in a subset of $\mathbb{R}$).
- The regression problem: To predict Y from known values of $\boldsymbol{X}$.
- The most common and convenient loss function is the squared error loss: $L(Y, h(\boldsymbol{X})) = (Y - h(\boldsymbol{X}))^2$.
- The expected loss is known as Prediction Mean Squared Error, (PMSE):

$$\text{PMSE}(h) = \mathbb{E}\left((Y - h(\boldsymbol{X}))^2\right).$$

- The Bayes rule in this case is

$$h_B(\boldsymbol{x}) = \arg\min_{y \in \mathcal{Y}} \mathbb{E}\left((Y - y)^2 | \boldsymbol{X} = \boldsymbol{x}\right).$$

- Bayes rule: $h_B(\boldsymbol{x}) = \arg\min_{y\in\mathcal{Y}} \mathbb{E}\left((Y-y)^2|\boldsymbol{X}=\boldsymbol{x}\right)$.
- Observe that, for any $y \in \mathcal{Y}$, $\mathbb{E}\left((Y-y)^2|\boldsymbol{X}=\boldsymbol{x}\right)$

$$= \mathbb{E}\left(((Y-\mathbb{E}(Y|\boldsymbol{X}=\boldsymbol{x}))+(\mathbb{E}(Y|\boldsymbol{X}=\boldsymbol{x})-y))^2|\boldsymbol{X}=\boldsymbol{x}\right)$$

$$= \mathbb{E}\left((Y-\mathbb{E}(Y|\boldsymbol{X}=\boldsymbol{x}))^2|\boldsymbol{X}=\boldsymbol{x}\right) + \underbrace{(\mathbb{E}(Y|\boldsymbol{X}=\boldsymbol{x})-y))^2}_{\geq 0}$$

$$+2(\mathbb{E}(Y|\boldsymbol{X}=\boldsymbol{x})-y)\underbrace{\mathbb{E}\left(Y-\mathbb{E}(Y|\boldsymbol{X}=\boldsymbol{x})|\boldsymbol{X}=\boldsymbol{x}\right)}_{=0}$$

$$\geq \mathbb{E}\left((Y-\mathbb{E}(Y|\boldsymbol{X}=\boldsymbol{x}))^2|\boldsymbol{X}=\boldsymbol{x}\right)$$

- The Bayes rule here is the conditional expectation of Y given $\boldsymbol{X}=\boldsymbol{x}$,

$$h_B(\boldsymbol{x}) = \mathbb{E}(Y|\boldsymbol{X}=\boldsymbol{x}),$$

also known as regression function of Y over $\boldsymbol{x}$ and usually denoted by

$$m(\boldsymbol{x}) = \mathbb{E}(Y|\boldsymbol{X}=\boldsymbol{x}).$$

- Parametric regression models assume that $m(\boldsymbol{x})$ is known except for a finite number of unknown parameters,
$$m(\boldsymbol{x}) \equiv m(\boldsymbol{x};\theta),\ \theta \in \Theta \subseteq \mathbb{R}^q,$$
- For instance, the multiple linear regression model postulates that $m(\boldsymbol{x}) = \beta_0 + \boldsymbol{x}^\mathsf{T}\boldsymbol{\beta}_1$, with unknown parameters $\beta_0 \in \mathbb{R}$, $\boldsymbol{\beta}_1 \in \mathbb{R}^p$.
- The training sample, $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, i.i.d. from $(\boldsymbol{X}, Y)$, is used to estimate the parameter θ.
- In this case $h_S(\boldsymbol{x}) = m(\boldsymbol{x};\hat{\theta})$, where $\hat{\theta} = \hat{\theta}(S)$ is the estimation of θ from sample S.

Least squares estimation

- In this context the usual way to estimate θ is by least squares (LS):

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \sum_{i=1}^{n} (y_i - m(\boldsymbol{x}_i; \theta))^2.$$

- This is equivalent to the maximum likelihood estimation of θ if $(\boldsymbol{X}, Y)$ is assumed to have a joint normal distribution. In this case:
  - The regression function $m(\boldsymbol{x})$ is linear in $\boldsymbol{x}$.
  - It is equivalent to state the model as

$$Y = m(\boldsymbol{X}) + \varepsilon,$$

where ε is an additive noise normally distributed with zero mean and independent from $\boldsymbol{X}$, also normally distributed.

Least squares estimation (II)

- Observe that the LS estimator $\hat{\theta}$ minimize the prediction error in the training sample (Residual Sum of Squares: $\text{RSS}(\theta) = \sum_{i=1}^{n}(y_i - m(\boldsymbol{x}_i;\theta))^2$), which takes its minimum value

$$\overline{\text{err}} = \text{RSS}(\hat{\theta}) = \sum_{i=1}^{n}(y_i - m(\boldsymbol{x}_i;\hat{\theta}))^2$$

- This quantity $\overline{\text{err}}$ is known as the training error or the apparent error.
- Usually it is an optimistic estimation of the Prediction Mean Squared Error (PMSE) in an observation of $(\boldsymbol{X}_{n+1}, Y_{n+1})$ independent from the training sample,

$$\text{PMSE}(\theta) = \mathbb{E}((Y_{n+1} - m(\boldsymbol{x}_i;\theta))^2),$$

mainly when the parametric family $m(\boldsymbol{x};\theta)$, $\theta \in \Theta \subseteq \mathbb{R}^q$, is too flexible:

$$\overline{\text{err}} < \text{PMSE}(\hat{\theta}) \neq \min_{\theta\in\mathbb{R}^q} \text{PMSE}(\theta).$$

- This is the case in non-parametric regression and in many machine learning algorithms. (Example: k-nearest neighbors regression, where the *tuning parameter* is k).
- We will talk later in the course about cross-validation and tuning parameters.

Example: k nearest-neighbors

- The k nearest-neighbor estimator of $m(\boldsymbol{t}) = E(Y|\boldsymbol{X} = \boldsymbol{t})$ is defined as

$$\hat{m}(\boldsymbol{t}) = \frac{1}{|N_k(\boldsymbol{t})|} \sum_{i \in N_k(\boldsymbol{t})} y_i,$$

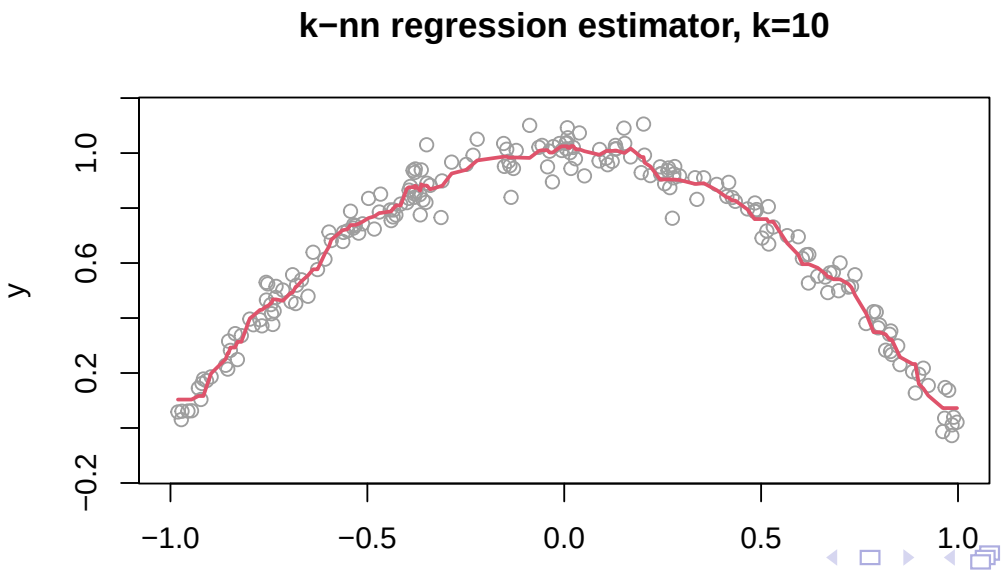
  where $N_k(\boldsymbol{t})$ is the neighborhood of $\boldsymbol{t}$ defined by the k closest points $\boldsymbol{x}_i$ in the training sample.
- *Closeness* is defined according to a previously chosen distance measure $d(\boldsymbol{t}, \boldsymbol{x})$, for instance, the Euclidean distance.

k-nn regression, in R

```
knn_regr<- function(x, y, t=NULL, k=3,
dist.method = "euclidean"){
  nx <- length(y)
  if (is.null(t)){
    t<- as.matrix(x)
  } else {
    t<-as.matrix(t)
  }
  nt <- dim(t)[1]
  Dtx <- as.matrix( dist(rbind(t,as.matrix(x)),
  method = dist.method) )
  Dtx <- Dtx[1:nt,nt+(1:nx)]
  mt <- numeric(nt)
  for (i in 1:nt){
    d_t_x <- Dtx[i,]
    d_t_x_k <- sort(d_t_x,partial=k)[k]
    N_t_k <- unname( which(d_t_x <= d_t_x_k) )
    mt[i]=mean(y[N_t_k])
  }
  return(mt)
}
```

Example of *k*-nn regression

```
n <- 200; sde <- .05
x <- sort(2*runif(n)-1)
mx <- 1-x^2;  e <- rnorm(n,0,sde)
y <- mx + e
plot(x,y,xlim=c(-1,1),ylim=c(-3*sde,1+3*sde),col=8)
lines(x,mx,col=4)
k <- n/20
hat_mx <- knn_regr(x,y, k= k )
lines(x,hat_mx,col=2,lwd=2)
title(main=paste0("k-nn regression estimator, k=",k))
```



Practice:

Follow the Rmd file `task_knn.Rmd`.

1. Instructors

2. Machine learning and statistical learning

3. Supervised and unsupervised learning
3.1 Supervised Learning (the prediction problem)
3.1.1 The regression problem
3.1.2 The classification problem

The classification problem

- Let $(\boldsymbol{X}, Y)$ be a r.v. with support $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^p \times \{1, \ldots, K\}$. We want to predict Y from observed values of $\boldsymbol{X}$.
- The loss function in this case can be represented by a $K \times K$ matrix $\boldsymbol{L}$, that will be zero on the diagonal and nonnegative elsewhere.
- The element (j, k) of $\boldsymbol{L}$ is $L(j, k)$: the price paid for classifying in class k an observation belonging to class j.

The zero-one loss function

- Most often the zero-one loss function is used, where all misclassifications are charged a single unit.
- With the 0-1 loss function the Bayes rule is

$$h_B(\boldsymbol{x}) = \arg\min_{y \in \mathcal{Y}} \mathbb{E}\left(L_{0\text{-}1}(Y, y) | \boldsymbol{X} = \boldsymbol{x}\right)$$

$$= \arg\min_{k \in \{1,\ldots,K\}} \sum_{j=1}^{K} L_{0\text{-}1}(j, k) \Pr(Y = j | \boldsymbol{X} = \boldsymbol{x})$$

$$= \arg\min_{k \in \{1,\ldots,K\}} (1 - \Pr(Y = k | \boldsymbol{X} = \boldsymbol{x})) = \arg\max_{k \in \{1,\ldots,K\}} \Pr(Y = k | \boldsymbol{X} = \boldsymbol{x}).$$

- In this context the Bayes rule is known as the Bayes classifier, and says that we classify to the most probable class, conditional to the observed value $\boldsymbol{x}$ of $\boldsymbol{X}$.

The problem of binary classification

- Consider the binary classification problem: $\mathcal{Y} = \{0, 1\}$. Then

$$(Y \mid \boldsymbol{X} = \boldsymbol{x}) \sim \text{Bernoulli}(p = p(\boldsymbol{x}) = \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \mathbb{E}(Y \mid \boldsymbol{X} = \boldsymbol{x})).$$

- The Bayes classifier is

$$h_B(\boldsymbol{x}) = \begin{cases} 1 & \text{if} \quad p(\boldsymbol{x}) \geq 1/2 \\ 0 & \text{if} \quad p(\boldsymbol{x}) < 1/2 \end{cases}$$

- As $p(\boldsymbol{x})$ is unknown, we use a training sample to estimate it.
- Let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ be n independent realizations of $(\boldsymbol{X}, Y)$.
- Given an estimation $\hat{p}(\boldsymbol{x})$ of the regression function $p(\boldsymbol{x})$, the estimated version of the Bayes classifier is

$$h_S(\boldsymbol{x}_{n+1}) = \begin{cases} 1 & \text{if} \quad \hat{p}(\boldsymbol{x}_{n+1}) \geq 1/2 \\ 0 & \text{if} \quad \hat{p}(\boldsymbol{x}_{n+1}) < 1/2 \end{cases}$$

- In practice, cut points different from $1/2$ can be used.

The problem of binary classification. Parametric estimation

- Parametric modeling: it is assumed that $p(\boldsymbol{x}) = \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$ is known except for a finite number of unknown parameters,
$$p(\boldsymbol{x}) \equiv p(\boldsymbol{x}; \theta),\ \theta \in \Theta \subseteq \mathbb{R}^q.$$
- The likelihood function is
$$L(\theta) = \prod_{i=1}^{n} \Pr(Y_i = y_i \mid \boldsymbol{X}_i = \boldsymbol{x}_i) = \prod_{i=1}^{n} p(\boldsymbol{x}_i; \theta)^{y_i} (1 - p(\boldsymbol{x}_i; \theta))^{1-y_i},$$
with logarithm
$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{n} \left(y_i \log p(\boldsymbol{x}_i; \theta) + (1 - y_i) \log(1 - p(\boldsymbol{x}_i; \theta)) \right).$$
- Let $\hat{\theta} = \arg\max_{\theta \in \Theta} \ell(\theta)$ be the maximum likelihood estimator of θ.
- Then $\hat{p}(\boldsymbol{x}) = p(\boldsymbol{x}; \hat{\theta})$ is used to define the classification rule.

The problem of binary classification. Other optimization criteria

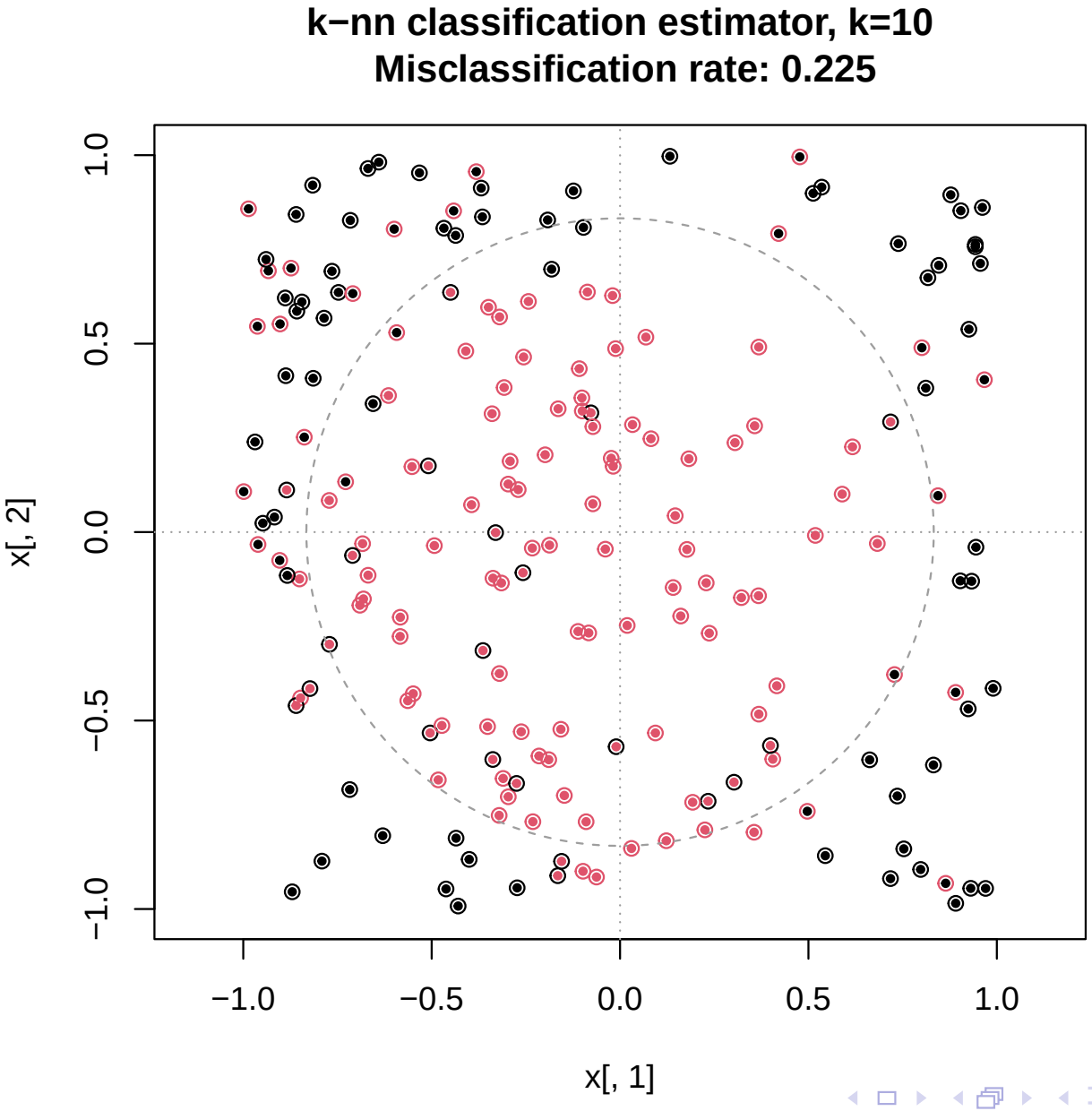
- Maximum likelihood is not the only possibility for estimating θ in $p(\boldsymbol{x};\theta)$.
- Alternatives:
  - Minimization of the misclassification error: $\hat{\theta}_{Miss} = \arg\min_{\theta\in\Theta} \sum_{i=1}^{n} (y_i - \mathbb{I}\{p(\boldsymbol{x}_i;\theta) \geq 0.5\})^2$.
  - Least squares estimation: $\hat{\theta}_{LS} = \arg\min_{\theta\in\Theta} \sum_{i=1}^{n} (y_i - p(\boldsymbol{x}_i;\theta))^2$.
  - Least absolute deviation: $\hat{\theta}_{LAD} = \arg\min_{\theta\in\Theta} \sum_{i=1}^{n} |y_i - p(\boldsymbol{x}_i;\theta)|$.
  - Penalized version of these criteria, when the statistical model $p(\boldsymbol{x};\theta)$, $\theta \in \mathbb{R}^q$, is too flexible.

k-nn classification, in R

```
knn_class<-function(x,y,t=NULL,k=3,dist.method="euclidean"){
  nx <- length(y)
  classes <- sort(unique(y))
  nclasses <- length(classes)
  if (is.null(t)){t<-as.matrix(x)}else{t<-as.matrix(t) }
  nt <- dim(t)[1]
  Dtx <- as.matrix(dist(rbind(t,as.matrix(x)), method=dist.
method))
  Dtx <- Dtx[1:nt,nt+(1:nx)]
  hat_probs_t <- matrix(0, nrow = nt, ncol=nclasses)
  hat_y_t <- numeric(nt)
  for (i in 1:nt){
    d_t_x <- Dtx[i,]
    d_t_x_k <- sort(d_t_x,partial=k)[k]
    Ntk <- unname( which(d_t_x <= d_t_x_k) )
    for (j in 1:nclasses){
      hat_probs_t[i,j]<-sum(y[Ntk]==classes[j])/length(Ntk)
    }
    hat_y_t[i] <- classes[which.max(hat_probs_t[i,])]
  }
  return(list(hat_y_t=hat_y_t, hat_probs_t=hat_probs_t))
}
```

Example of k-nn classification

```
n <- 200; sd_eps <- .05
x <- matrix(2*runif(2*n)-1, ncol=2)
px <- exp(-x[,1]^2-x[,2]^2)
y <- rbinom(n,size=1,prob = px)
plot(x[,1],x[,2],xlim=c(-1,1),ylim=c(-1,1), col=y+1, asp=1)
abline(h=0,v=0,col=8,lty=3)
lines(sqrt(log(2))*cos(seq(0,2*pi,length=201)),
sqrt(log(2))*sin(seq(0,2*pi,length=201)),col=8,lty=2)
k <- n/20
hat_y <- knn_class(x,y, k= k)
points(x[,1],x[,2], pch=19, cex=.5, col=hat_y$hat_y_t+1)
title(main=paste0("k-nn classification estimator, k=",k,
"\n Misclassification rate: ", mean(y!=hat_y$hat_y_t)))
```


k−nn classification estimator, k=10
Misclassification rate: 0.225
x[, 1]
x[, 2]
−1.0
−0.5
0.0
0.5
1.0

Practice:

Follow the Rmd files

`SimMixtNorm.Rmd` and `knn_class.Rmd`.

Baumer, B. S., D. T. Kaplan, and N. J. Horton (2017).
Modern Data Science with R.
CRC Press.

Efron, B. and T. Hastie (2016).
Computer Age Statistical Inference.
Cambridge University Press.

Hastie, T., R. Tibshirani, and J. Friedman (2009).
The Elements of Statistical Learning (2nd ed.).
Springer.

Hastie, T., R. Tibshirani, and M. Wainwright (2015).
Statistical learning with sparsity: the lasso and generalizations.
CRC Press.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013).
An Introduction to Statistical Learning with Applications in R.
Springer.

Murphy, K. P. (2012).
Machine Learning: A Probabilistic Perspective.
The MIT Press.