Machine Learning and Pattern Recognition Practice Session I

Martin Palazzo

Universite de Technologie de Troyes
Universidad Tecnologica Nacional Buenos Aires
Biomedicine Research Institute of Buenos Aires - Max Planck Partner

martin.palazzo@utt.fr

October 22, 2020

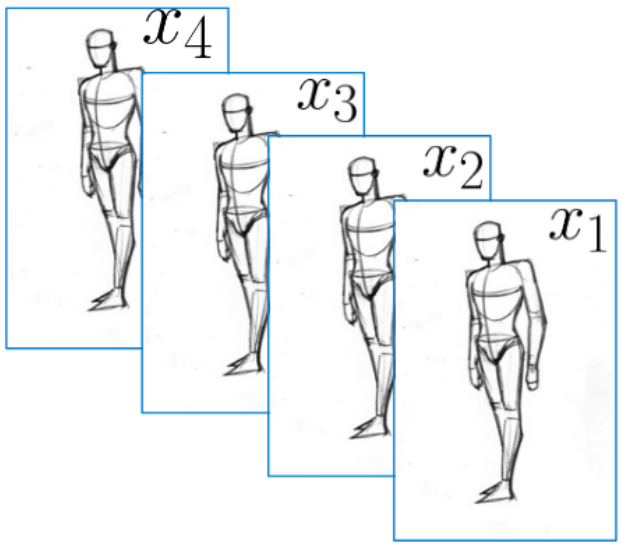
Overview

1. Supervised Learning
2. Bayes Decision Theory
   - Univariate cases
   - Multi-variate cases
3. Maximum Likelihood parameter estimation
4. Non-parametric methods
   - Parzen Window
   - K Nearest Neighbors

Disclaimer

The following document has been created as supporting and guiding material during the practical lessons during the Pattern Recognition course of the Master OSS. The official bibliography and theory materials have been distributed previously by the organization committee of the Master OSS.

Data



$$X_i = [x_{i1}, x_{i2}, x_{i3}, \ldots, x_{id}]$$

X1**1** (feat. 1 in sample 1)
X1**2** (feat. 2 in sample 1)
X1**3** (feat. 3 in sample 1)
X1**d** (feat. d in sample 1)

$$\mathcal{X} \in \mathbb{R}^d$$

Figure: Samples and features.

Learning approaches

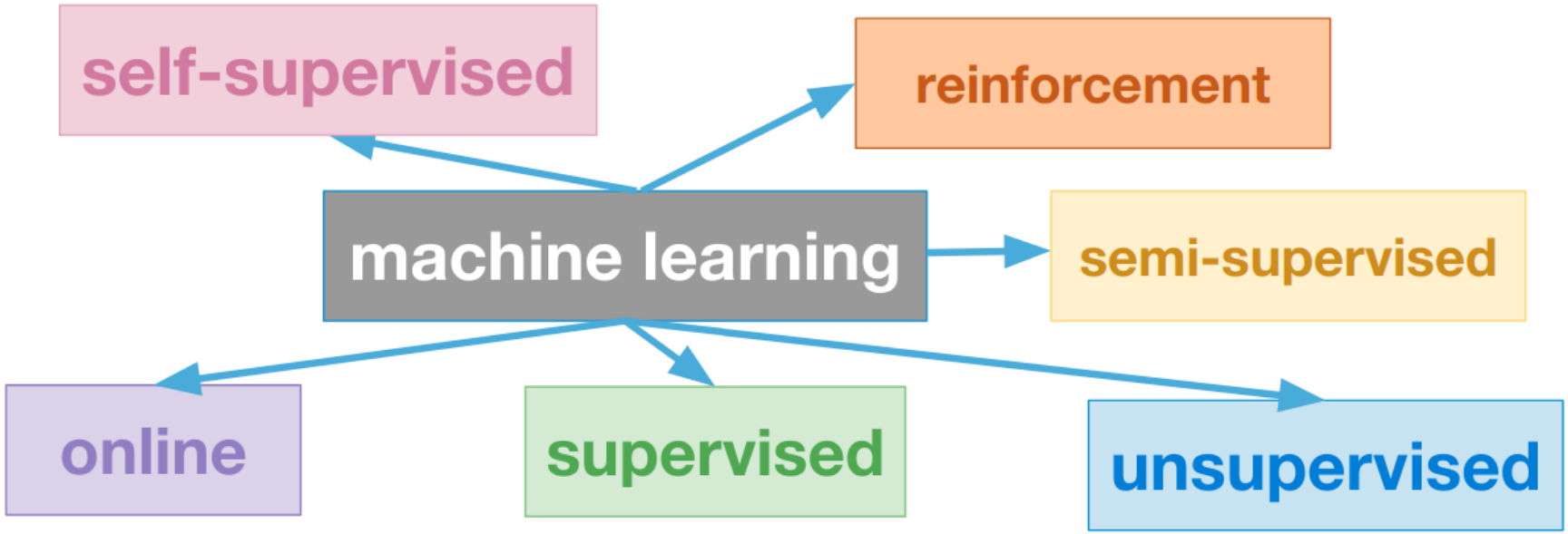


Figure: Learning approaches.

Supervised Learning

Supervised Learning

Our data is expressed as vectors x of dimension d.

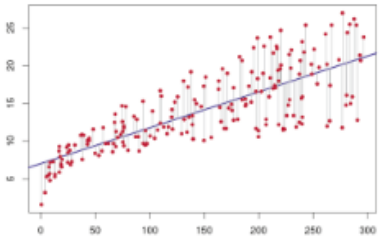
$$\begin{array}{rcl} X \in \mathcal{X} \subset \mathbb{R}^d & \rightarrow & \text{our data} \\ x_i \in X & \rightarrow & \text{our samples} \\ Y \in \mathcal{Y} \subset \mathbb{R} & \rightarrow & \text{labels} \\ y_i \in Y & \rightarrow & \text{sample labels} \end{array} \tag{1}$$

Sample set with labels

In a supervised learning approach, each sample x_i belongs to a class with label y_i and the sample set S is defined as $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$

Supervised Learning: regression and classification

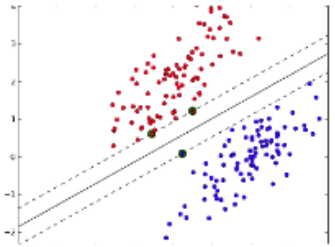
regression →



Y is a real number

$$y \subseteq \mathbb{R}$$

classification →



Y is categorical

$$y \in \{-1, 1\}$$

Figure: A labeled dataset.

Supervised Learning: classification

A supervised learning problem can present different types of labels and number of classes c. Particularly when the classes are categorical there are three approaches to consider

Binary Classification problem

There are two classes and the label vector Y has two values

$$Y = \begin{cases} +1 & \text{if } y_i = c_0 \\ -1 & \text{if } y_i = c_1 \end{cases} \tag{2}$$

Multi-class Classification problem

There are m classes and the labels vector Y is represented as

$$Y \in 1, 2, 3, ..., m \quad \text{if} \quad y_i = c_k \tag{3}$$

One-class classification problem

There are only one class c_0.

Supervised Learning

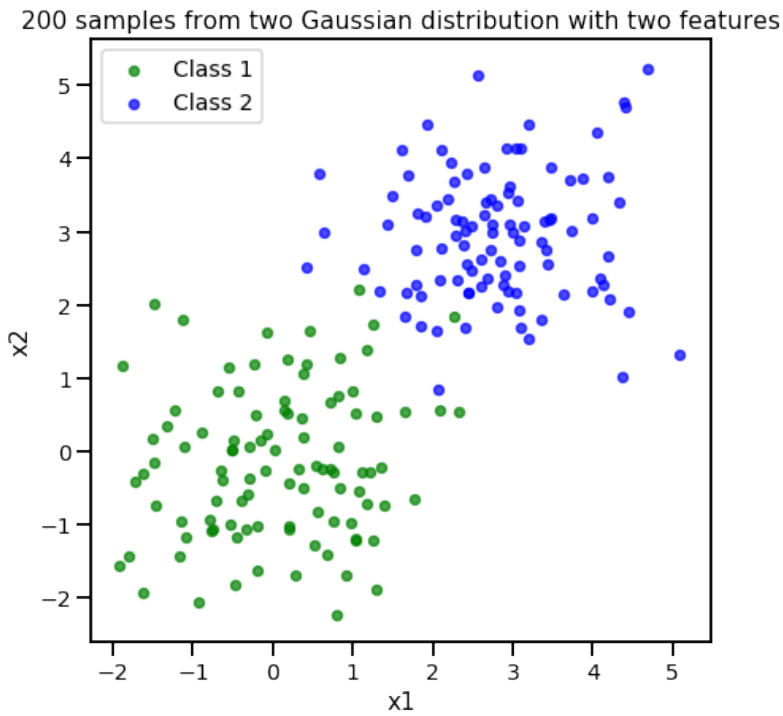


Figure: A labeled dataset.

Supervised approaches have the condition that every sample is associated to a label and the problem generally is to find the rule that explain the relation between each sample x and its label y. An unsupervised approach only consider the set of samples S.

Supervised Learning

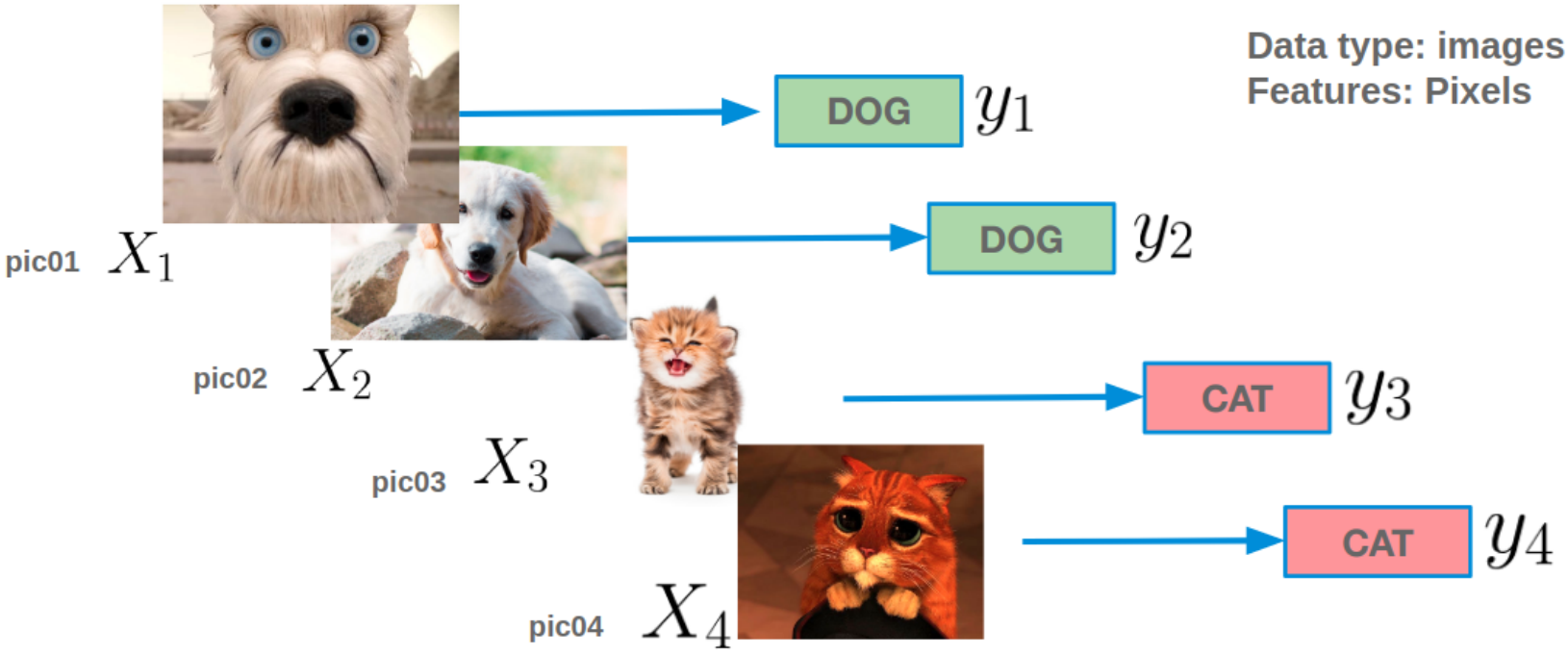


Figure: A labeled dataset.

Supervised Learning

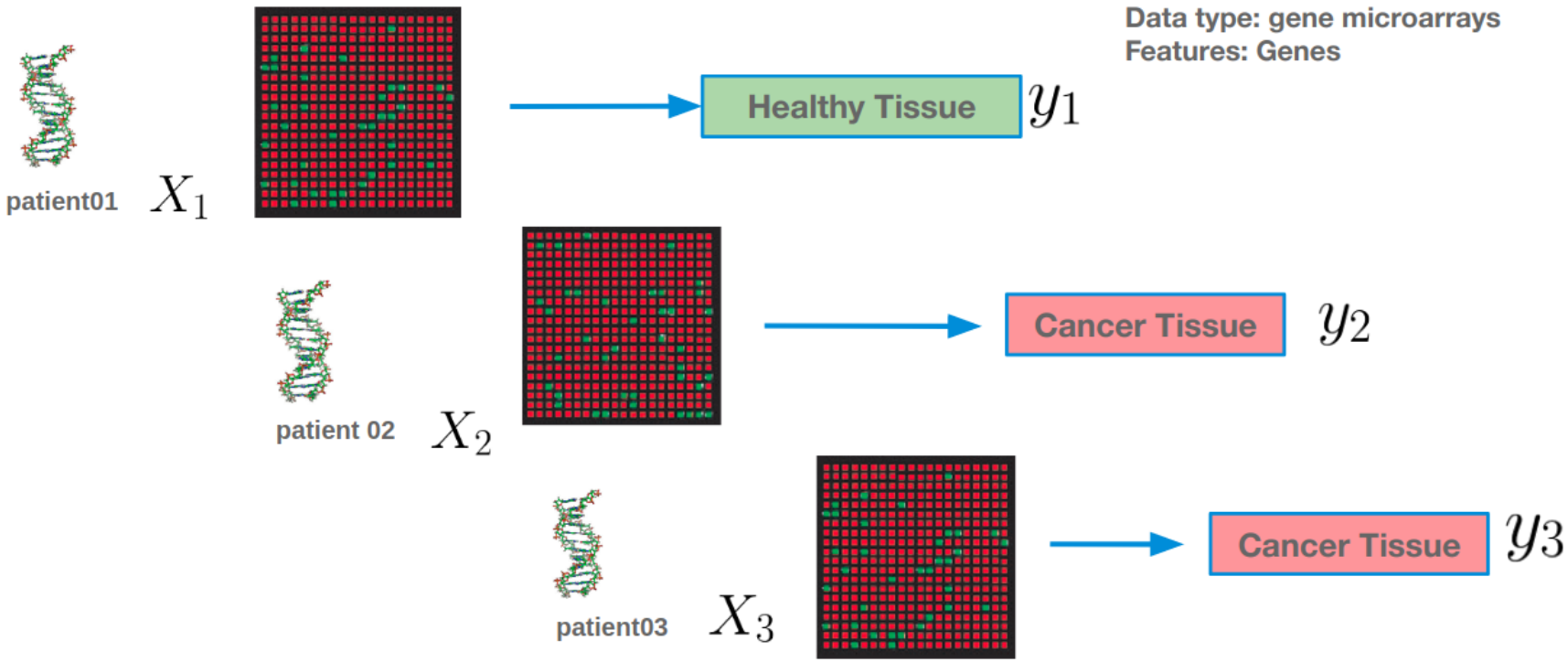


Figure: A labeled dataset.

Supervised Learning

A supervised problem invites to propose a rule $f(x)$ that explain the relation between x and y. The rule can be determined by the following approaches depending of the availability of the data.

- Parametric Learning (learn parameters of known density distributions)
- learn a rule from a set of data S
- a human to define the rule (like pathologists)

Initially we will explore rules defined by parametric learning assuming we know the probability distribution of our data. In the following chapters we will estimate the rules from a set of data without having any knowledge about the underlying distribution.

Supervised Learning

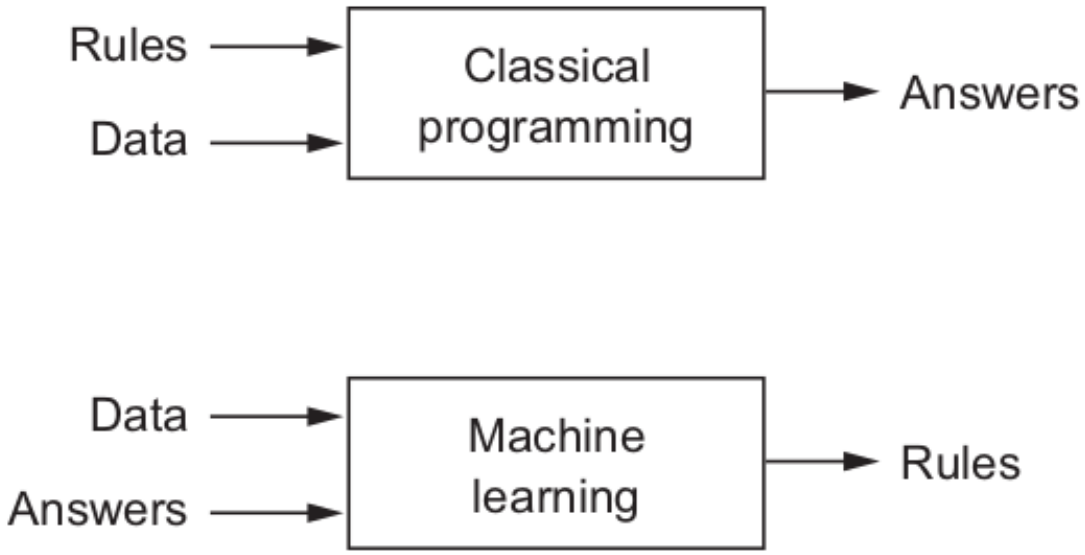


Figure: Machine Learning programming paradigm (Deep Learning with Python, Francois Chollet, 2018) [1]

The learning algorithms build rules based on data.

Classification: decision function

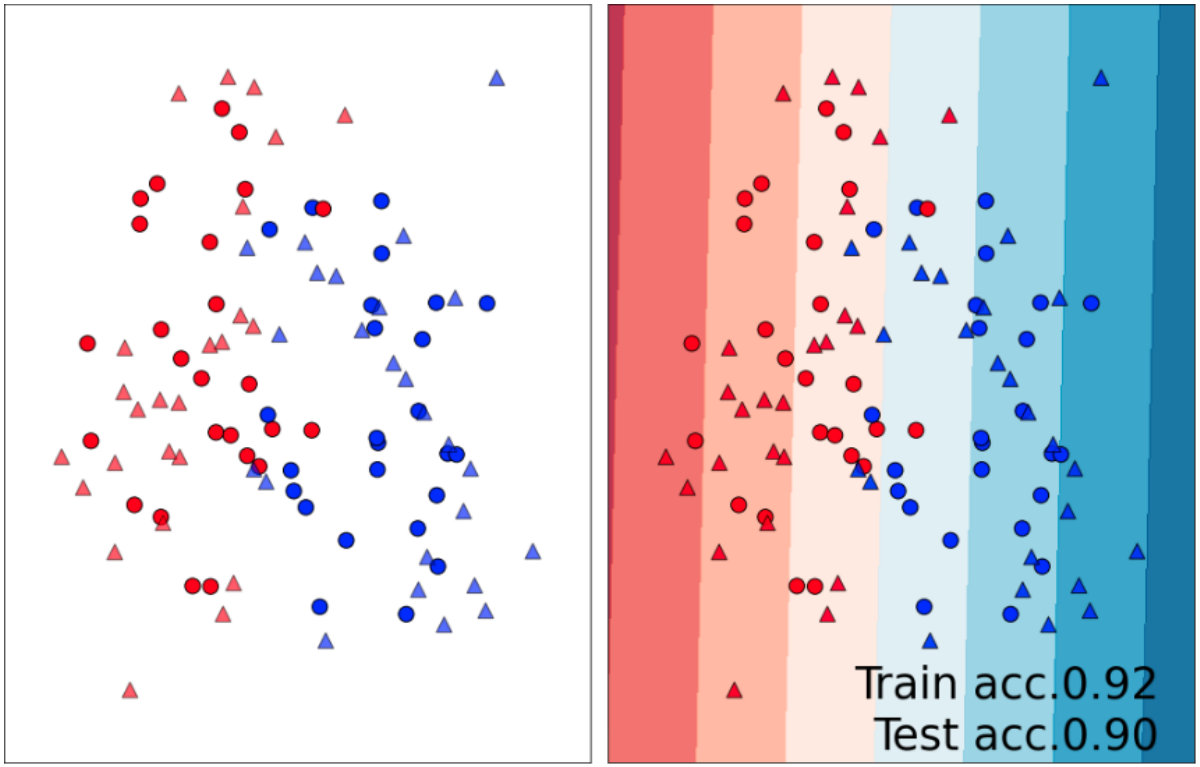


Figure: Linear decision boundary) [1]

The learning algorithms build rules based on data.

Supervised Learning: decision function

Learning f(x)

From the dataset S a function is learned such as $f(x) = y$ where $f(x)$ is the decision boundary (in classification).

Empirical error rate

Expected Loss

$$\mathcal{E}(f) = \mathbb{E}\left[L(y, f(x))\right]$$

Empirical error

$$\mathcal{E}(f) = \mathbb{E}\left[L(y, f(x))\right] = \frac{1}{n}\sum_{n=1}^{N} \delta(f(x_i) - y_i)$$

where

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

Bayes Decision Theory

Bayes Rule

Bayes Rule

$$P(c_j \mid x) = \frac{p(x \mid c_j)P(c_j)}{p(x)}$$

Where $P(c_j)$ is the prior probability, $p(x \mid c_j)$ is the likelihood, $P(c_j \mid x)$ is the posterior and $p(x)$ is an scaling factor.

Decision boundary by Bayes Rule

Suppose that we want to classify samples generated by a random variable of known distribution and parameters. We want to build a decision boundary given a set of classes $C = \{C_0, C_1\}$ using Bayes rule. We have a set of data generated by a Gaussian distribution for each class with same variance and different mean which are known parameters:

$$p(x|c_j) \sim N(\mu, \sigma)$$

with

$$p(x|c_j) \sim \frac{1}{\sqrt{2\pi\sigma^2}} \mathbf{exp}\left[\frac{-1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

and the prior probabilities $P(c_0) = P(c_1) = 0.5$, the variance of the distribution as $\sigma_0^2 = \sigma_1^2 = 1$ and the means $\mu_0 = 6$ and $\mu_1 = 12$.

Decision boundary by Bayes Rule

From the Bayes expression we want to decide c_0 if $P(c_0 \mid x) > P(c_1 \mid x)$, else we decide c_1. Then

$$P(c_j \mid x) = \frac{p(x \mid c_j)P(c_j)}{p(x)} \tag{4}$$

$$\frac{p(x \mid c_0)P(c_0)}{p(x)} > \frac{p(x \mid c_1)P(c_1)}{p(x)} \tag{5}$$

$$p(x \mid c_0)P(c_0) > p(x \mid c_1)P(c_1) \tag{6}$$

Decision boundary by Bayes Rule

$$\frac{p(x \mid c_0)}{p(x \mid c_1)} > \frac{P(c_1)}{P(c_0)} \tag{7}$$

$$\frac{p(x \mid c_0)}{p(x \mid c_1)} > \frac{0.5}{0.5} \tag{8}$$

$$\frac{p(x \mid c_0)}{p(x \mid c_1)} > 1 \tag{9}$$

$$p(x \mid c_0) > p(x \mid w_1) \tag{10}$$

We can compute the equation 10 using the input values of the distribution given above :)

Decision boundary by Bayes Rule

$$\frac{1}{\sqrt{2\pi\sigma^2}}\mathbf{exp}\left[\frac{-1}{2}\left(\frac{x-\mu_0}{\sigma}\right)^2\right] > \frac{1}{\sqrt{2\pi\sigma^2}}\mathbf{exp}\left[\frac{-1}{2}\left(\frac{x-\mu_1}{\sigma}\right)^2\right] \quad (11)$$

Remember we have different mean and the same variance, then the first term is not necessary

$$\mathbf{exp}\left[\frac{-1}{2}\left(\frac{x-\mu_0}{\sigma}\right)^2\right] > \mathbf{exp}\left[\frac{-1}{2}\left(\frac{x-\mu_1}{\sigma}\right)^2\right] \quad (12)$$

Then compute the decision function $f(x)$ from the input parameters μ and σ of each class.

Decision boundary by Bayes Rule

Find the Bayes decision rule using the known parameters of the probability distributions and the prior probabilities.

Exercise 01

Same variance, different mean, same prior and 1 feature:

- $P(c_0) = P(c_1) = 1/2$
- $\sigma_0^2 = \sigma_1^2 = 1$
- $\mu_0 = 6$ and $\mu_1 = 12$

How is the obtained decision function?

Decision boundary by Bayes Rule

Find the Bayes decision rule using the known parameters of the probability distributions and the prior probabilities.

Exercise 02

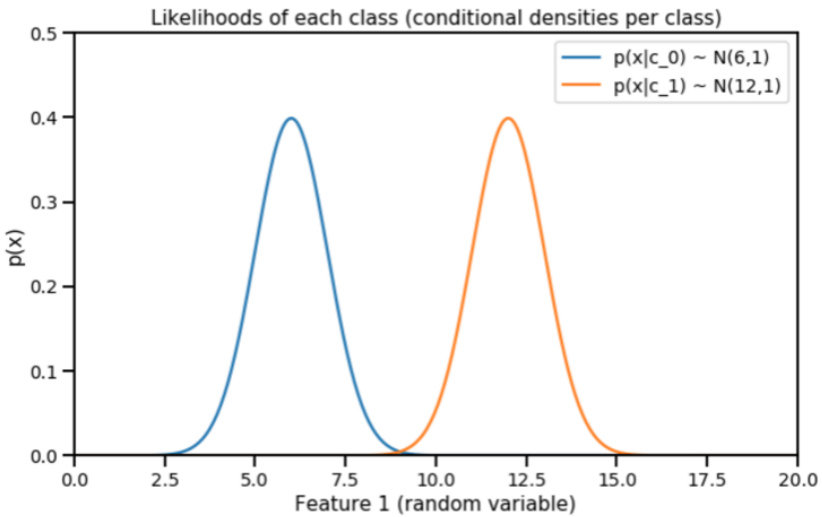
Same variance, different mean, different prior and 1 feature:

- $P(c_0) = 3/4$ and $P(c_1) = 1/4$
- $\sigma_0^2 = \sigma_1^2 = 1$
- $\mu_0 = 6$ and $\mu_1 = 12$

How is the obtained decision function?

Decision boundary by Bayes Rule

$$P(C_0) = P(C_1) = 0.5$$



$$P(C_0) = 0.75,\ P(C_1) = 0.25$$

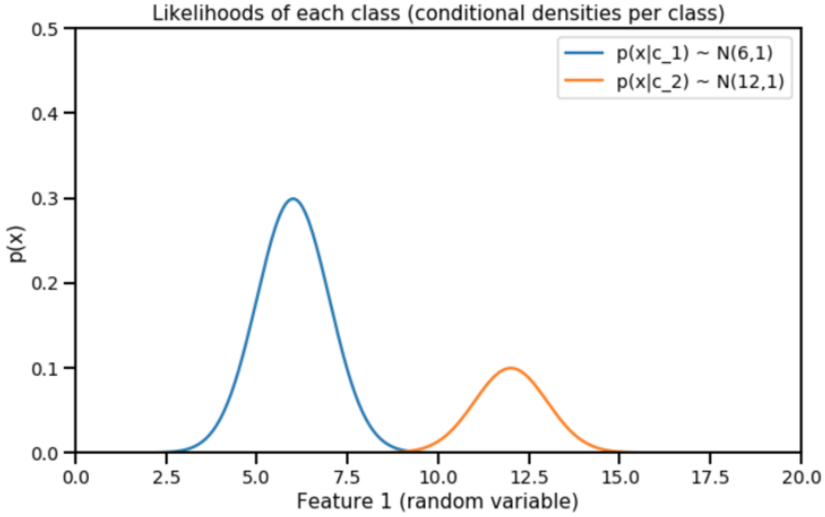


Figure: Exercise 01 and 02 probability density functions of each class.

If we know the family of PDF, its parameters and its prior probabilities then we can determine analytically the decision function.

Decision boundary by Bayes Rule

Exercise 03

For each of the two previous exercises use the probability density function of each class to generate 100 new samples. Then by using the previously obtained decision boundaries compute the empirical error rate for the 100 new samples per clas per exercise.

Decision boundary by Bayes Rule

Find the Bayes decision rule using the known parameters of the probability distributions and the prior probabilities.

Exercise 04

Same variance, different mean, same prior and 1 feature:

- $P(c_0) = P(c_1) = 1/2$
- $\sigma_0^2 = \sigma_1^2 = 2$
- $\mu_0 = 6$ and $\mu_1 = 12$

Exercise 05

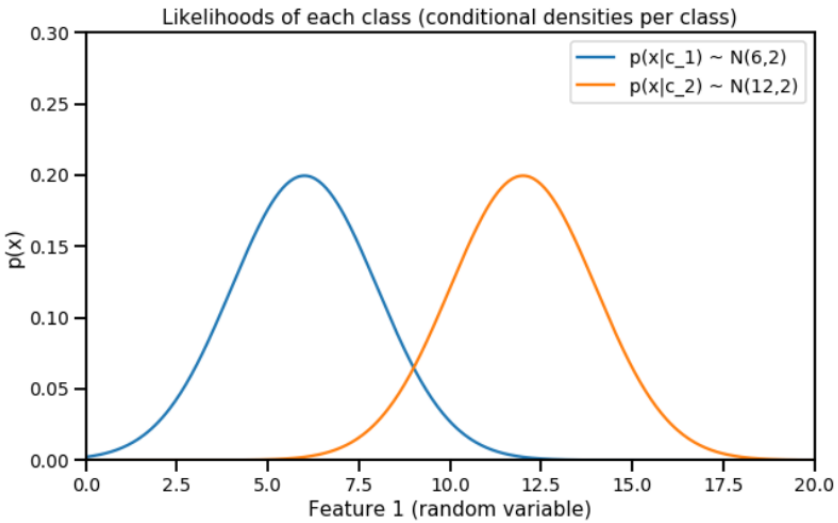
Same variance, different mean, different prior and 1 feature:

- $P(c_0) = 3/4$ and $P(c_1) = 1/4$
- $\sigma_0^2 = \sigma_1^2 = 2$
- $\mu_0 = 6$ and $\mu_1 = 12$

How is the obtained decision function?

Decision boundary by Bayes Rule

$$P(C_0) = P(C_1) = 0.5$$



$$P(C_0) = 0.75,\ P(C_1) = 0.25$$

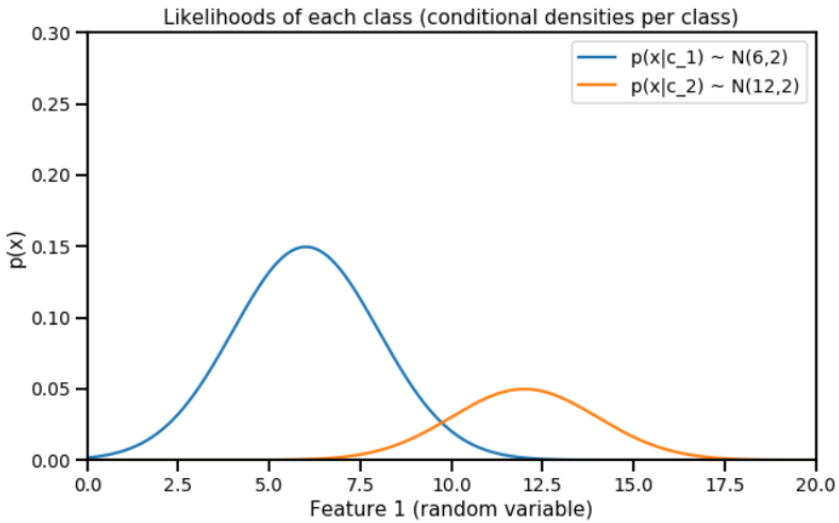


Figure: Exercise 03 and 04 probability density functions of each class.

If we know the family of PDF, its parameters and its prior probabilities then we can determine analytically the decision function.

Decision boundary by Bayes Rule

Exercise 06

For each of the two previous exercises use the probability density function of each class and generate 100 new samples. Then compute the empirical error rate considering the obtained decision rule.

Decision boundary by Bayes Rule

Find the Bayes decision rule using the known parameters of the probability distributions and the prior probabilities.

Exercise 07

Different variance, different mean, same prior and 1 feature:

- $P(c_0) = P(c_1) = 1/2$
- $\sigma_0^2 = 3$
- $\sigma_1^2 = 1$
- $\mu_0 = 6$ and $\mu_1 = 12$

How is the obtained decision function?

Exercise 08

For the exercises 07 use the probability density function of each class and generate 100 new samples. Then compute the empirical error rate considering the obtained decision rule.

Decision boundary by Bayes Rule

We can extend the Bayes decision rule to a multi-variate context. This time using two features x_1 and x_2.

$$p(X|c_j) \sim (2\pi)^{-d/2}|\Sigma^{-1/2}|\mathbf{exp}\left[-\frac{1}{2}(X-M)^t\,\Sigma^{-1}\,(X-M)\right]$$

Then the problem is defined as

$$X = [x_1, x_2]; M_1 = [0,0]; M_2 = [2,2]; P(C_1) = P(C_2) = 0.5$$

with a co-variance matrix

$$\Sigma_1 = \Sigma_2 = \begin{bmatrix} \sigma_{00}^2 & \sigma_{01}^2 \\ \sigma_{10}^2 & \sigma_{11}^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

So the problem presents two classes generated by normal distributions of equal variance, equal prior probabilities and different means.

Decision boundary by Bayes Rule

We want to decide c_0 if $P(c_0 \mid x) > P(c_1 \mid x)$, else we decide c_1. Using the equation 10 obtained from the problem with equal variance, equal priors and different mean we can express the problem as

$$q(x) = \frac{p(x \mid c_0)}{p(x \mid w_1)} > 0$$

$$log(q(x)) = [-1/2(X-M_0)^T\Sigma^{-1}(X-M_0)]-[-1/2(X-M_1)^T\Sigma^{-1}(X-M_1)]$$

$$(M_0 - M_1)^T\Sigma^{-1}X + 1/2M_1^T\Sigma^{-1}M_1 - 1/2M_0^T\Sigma^{-1}M_0 > 0$$

Decision boundary by Bayes Rule

Exercise 09

Given the multi-variate context, find the decision function. Once it is defined, generate 200 random samples from the probability density function of each class and compute the empirical error of the obtained boundary.

About the model

What assumption taken into account in the model do you think is naive?

About the decision boundary

How is the obtained decision function?

Maximum Likelihood parameter estimation

Maximum Likelihood parameter estimation

Figure: We assume that the conditional probability density function of each class is a Gaussian distribution.

There are cases where we know the family of the probability density function of each class but the parameters are unknown. Via MLE we can estimate the parameters using data coming from these distributions. Finally with the obtained parameters we can define a decision function via Bayes rule.

Maximum Likelihood parameter estimation

The conditional gaussian distribution for a given class c is

$$p(X|M,\Sigma) \sim (2\pi)^{-d/2}|\Sigma^{-1/2}|\mathbf{exp}\left[-\frac{1}{2}(X-M)^t\,\Sigma^{-1}\,(X-M)\right]$$

given a set of data $X = (x_1, x_2, x_3, ..., x_n)^t$ drawn from a gaussian distribution, its log-likelihood is [2]

$$\ln[p(X|M,\sigma)] = -\frac{Nd}{2}ln(2\pi) - \frac{N}{2}|\Sigma| - \frac{1}{2}\sum_{n=1}^{N}(X_n - M)^t\Sigma^{-1}(X_n - M)$$

Then applying the derivative of the log likelihood with respect to M

$$\frac{\partial}{\partial M}ln[p(x|M,\Sigma)] = \sum_{n=1}^{N}\Sigma^{-1}(X_n - M)$$

Maximum Likelihood parameter estimation

Finally, making the derivative equal to zero let us estimate the value of $\hat{\boldsymbol{M}}$ as

$$\hat{\boldsymbol{M}} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

and we obtain the maximum likelihood estimate of the mean. For the covariance parameter

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{n=1}^{n}(\boldsymbol{x}_n - \hat{M})(\boldsymbol{x}_n - \hat{M})^t$$

Maximum Likelihood parameter estimation

Exercise 10

Given 100 random samples of each class generated by the conditional density functions of exercise 09 estimate the mean vector $\hat{M}$ and its co-variance matrix $\hat{\Sigma}$ for both distributions. Compare these estimations with the true parameters. Finally plug the new parameters in the Bayes framework and find an estimated decision boundary between the two classes.

Conclusions

Decision functions by parametric approaches to density modelling

We have considered Gaussian PDF with different mean, equal or different variance, equal or different prior probabilities and with known and unknown parameters. What can we conclude about the obtained classification boundaries?

Non-parametric methods

Non-parametric methods

The first part of this workshop put focus on the use of known probability density functions determined by parameters whose values are to be determined from a data set. This is called the parametric approach to density modelling. An important limitation of this approach is that the chosen density might be a poor model of the distribution that generates the data, which can result in poor predictive performance. Reality presents probability density functions more complex than a Gaussian one. In the following sections non-parametric density estimation via frequentist approaches are presented. There are two strategies to consider:

- count the number of samples of c_0 and c_1 on a fixed volume
- count the number of samples of c_0 and c_1 on a variable volume

Non-parametric density estimation

If we can not assume the PDF where our data samples lies, then we can try other density estimations. The most easy one is the histogram. It simply partition x into several bins of width Δ_i and then count the number n_i of samples within the bin i. In order to turn this count into a normalized probability density, divide by the total number N of observations and by the width i of the bins to obtain probability values for each bin [2].

$$\hat{p}(X) = \frac{n_i}{N\Delta_i}$$

The resultant probability density is constant over the width of each bin.

Parzen window for density estimation

We want to estimate the class-conditional densities (likelihoods) without any prior information about them. Instead of knowing the full set of parameters that define the likelihood we can estimate the probability of each sample to classify and then make the decision (likelihood ratio). The Parzen Windows method proposes a certain region or volume v around each sample to make the estimate.

$$\hat{p}(X) = \frac{n(X)}{vN} = \frac{1}{N} \sum_{i=1}^{N} K(x - x_i)$$

The probability will be determined as the ratio between the number of samples n within the volume V and the total number of samples N.

$$\hat{p}(X) = \frac{nV}{N}$$

There are many types of volumes V. In this practice we will focus on Gaussian and Uniform volumes or "Kernels".

Parzen window for density estimation

The kernel function is defined as

$$k(u) = \begin{cases} 1 & |u_i| < 1/2 \\ 0 & otherwise \end{cases}$$

and its value will be 1 if the sample x_n lies inside of a cube of side h centered on x, then the kernel final value will be

$$K = \sum_{n=1}^{N} k(\frac{x - x_n}{h})$$

and for the full density estimation

$$\hat{p}(x) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{h^d} k(\frac{x - x_n}{h})$$

where h^d is the volume V of a hypercube of side h.

Parzen window for density estimation

Supose $d = 1$, $h = 1$, $K = 1_{(-\frac{1}{2},\frac{1}{2})}$, compute $\hat{p}(x)$ graphically on the following data samples:

$$x_1 = 1 \quad x_2 = 5 \quad x_3 = 5.25 \quad x_4 = 7$$

by using a uniform kernel

$$\hat{p}(x) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{h^d} k(\frac{x - x_n}{h})$$

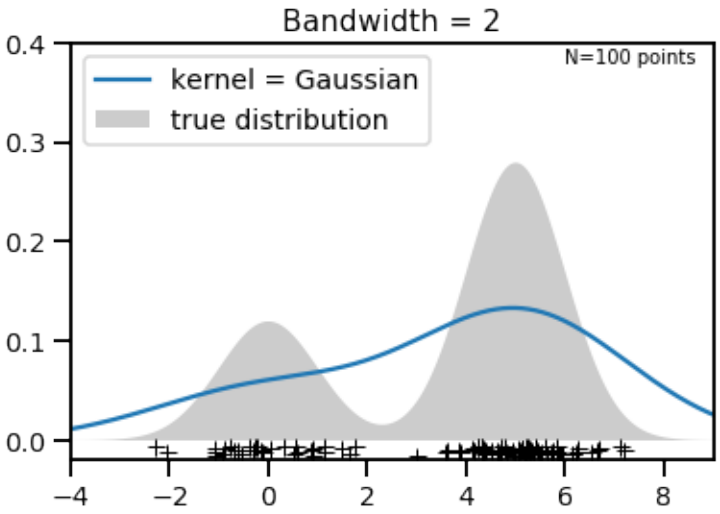
Parzen window for density estimation

Since the boundaries of the hypercube can present discontinuities, a Gaussian Kernel can be used to make the estimation smoother

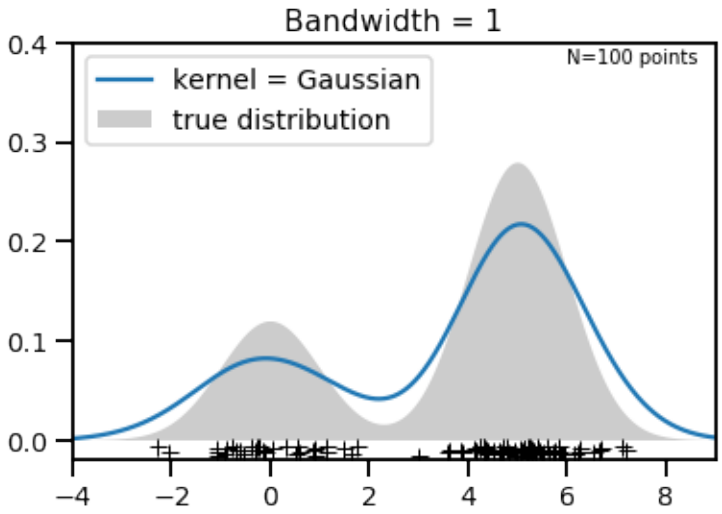
$$\hat{p}(x) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{\sqrt{2\pi h^2}}\textbf{exp}\left[\frac{||x - x_n||^2}{2h^2}\right]$$

where h corresponds in this case to the standard deviation as $h = \sigma$. Finally, with Gaussian or Hypercube kernel we can estimate $P(x|c_0)$ and $P(x|c_1)$. If we want to perform classification for a new sample x_i then the likelihood ratio obtained from Parzen method has to be computed.

Parzen window for density estimation



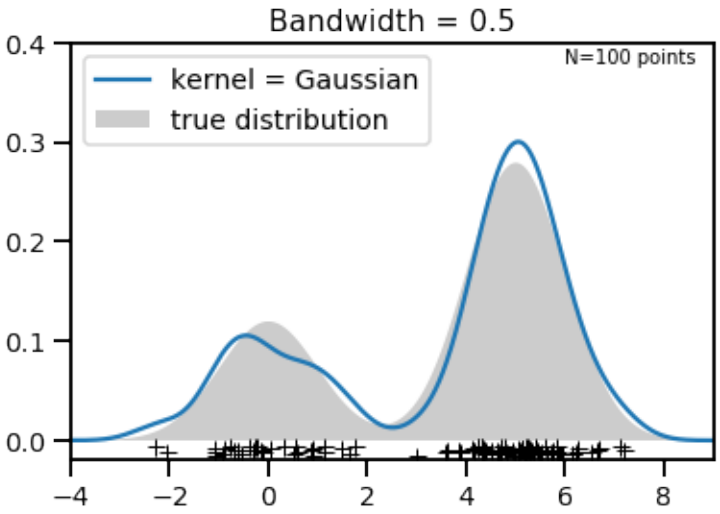
(a) K=1



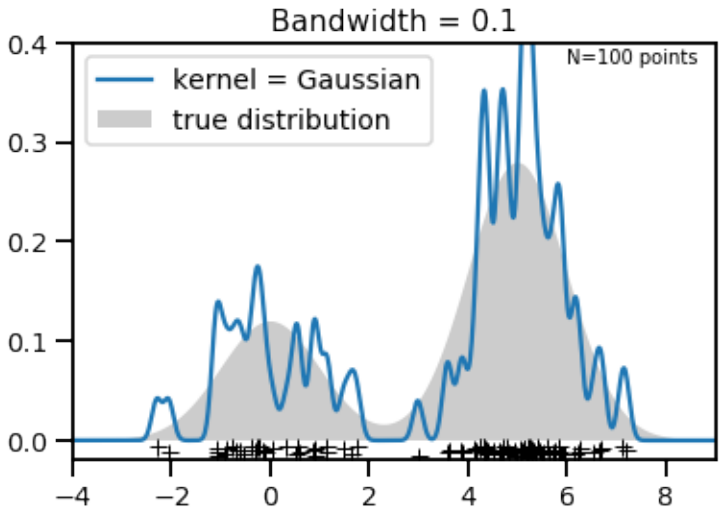
(b) K=5

Figure: Parzen window using gaussian kernel for different values of sigma

Parzen window for density estimation



(a) K=1



(b) K=5

Figure: Parzen window using gaussian kernel for different values of sigma

Question

Which value of sigma has the lowest empirical error rate?

Parzen window for density estimation

Exercise 11

Generate 200 samples of class C_0 using a gaussian PDF of $\mu = [0, 0]$ and $\Sigma = [[1, 0], [0, 1]]$. Then estimate in the origin the density using an hypercube and a gaussian kernel. Compute the differences between the Parzen method and the grownd truth values. Try different values of h.

Exercise 12

Generate 200 samples of class C_1 using a gaussian PDF of $\mu = [2, 2]$ and $\Sigma = [[1, 0], [0, 1]]$. Generate 10 random samples using uniform distributions. Determine to which class belong using a Gaussian and Hypercube kernel. Try different values of h.

K Nearest Neighbors

We have seen with the Parzen estimation windows that defining a fixed volume V by an hypercube or a Gaussian Kernel let to estimate the likelihood $P(x|C_i)$ by counting the number of samples within V and its relation to the total number of samples. One of the limitations of Parzen Windows is that we have to select a parameter h to define the size of the fixed volume. This leads a problem when the density of the data samples changes significantly. A high value of h will not detect details. On the other hand a low value of h can lead to noisy estimates.

K Nearest Neighbors

KNN density estimation

KNN considers the closests K neighbors of each sample to compute the likelihood $P(x|C_i)$. The parameter to tune is not anymore h but K now. By this way the volume to consider is variable and the number of samples fixed.

$$P(x) = \frac{k_i}{N}$$

The value k_i is the total amount of samples of class i selected in the k nearest neighbors.

K Nearest Neighbors

Then, how to make classification with KNN?

$$p(x|C_i) = \frac{K_i}{N_i} \qquad \text{The likelihood}$$

$$P(x) = \frac{K}{N} \qquad \text{The unconditional density}$$

$$P(C_{ik}) = \frac{N_{ik}}{N} \qquad \text{Prior for class } C_i$$

With this we can use bayes and compute the posterior probability

$$P(C_i|x) = \frac{p(x|C_i)P(C_{ik})}{P(x)} = \frac{K_i}{K}$$

where K_i is the total samples of class C_i in the subset of K nearest neighbors.

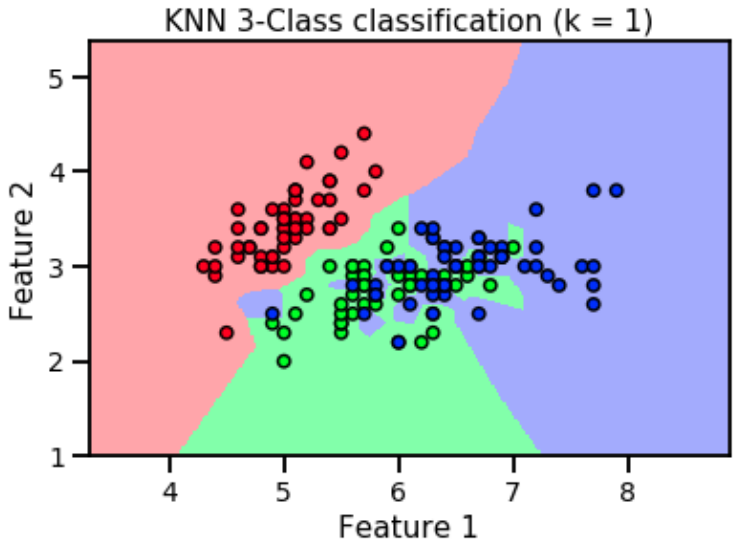
K Nearest Neighbors: Classification

Classification with KNN

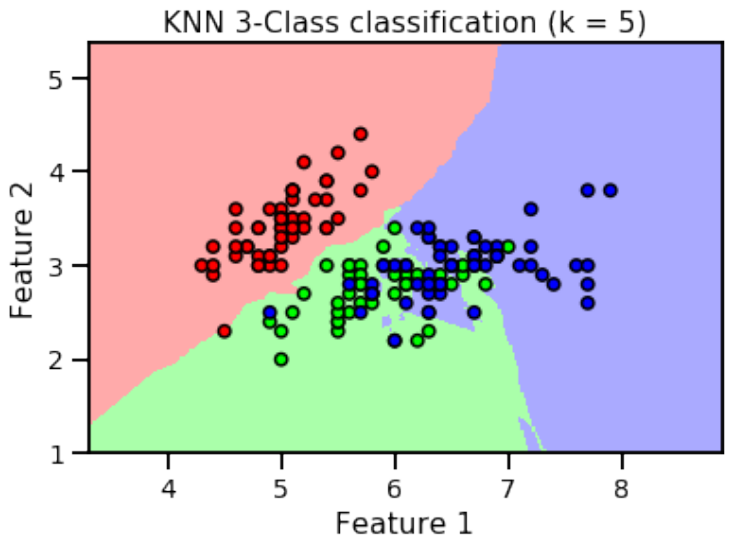
If we want to make classification with KNN, what we have to do is to asssign to a new sample X_p the highest value of the posterior probability $P(C_i|x)$ found. The same principle is used with Parzen window method.

$$\max P(C_i|x) \forall i$$

K Nearest Neighbors: Classification



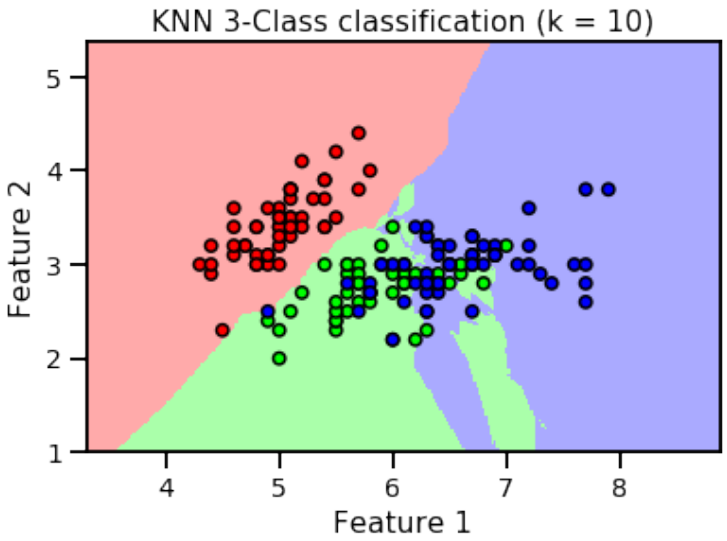
(a) K=1



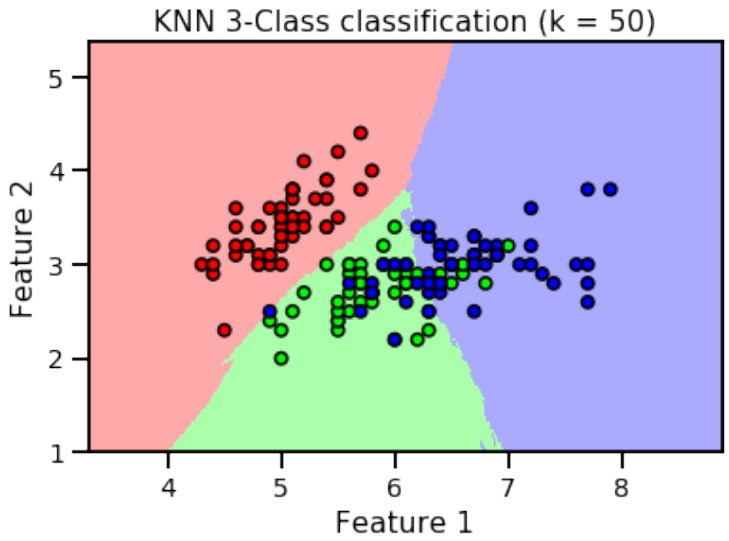
(b) K=5

Figure: KNN classification for different values of K

K Nearest Neighbors: Classification



(a) K=10



(b) K=50

Figure: KNN classification for different values of K

Question

How is the behaviour of the classifier when we increase K? How is the evolution of the empirical error when we change K?

K Nearest Neighbors: Classification

Exercise 13

Generate 20 samples for each of the two classes C_0 and C_1 with gaussian distributions. The first has $\mu_0 = [0, 0]$ and $\Sigma_0 = [[1, 0], [0, 1]]$. The second has $\mu_1 = [1, 1]$ and $\Sigma_1 = [[1, 0], [0, 1]]$. Develop an KNN algorithm from scratch. Suppose you know the labels of all of the 40 samples. Then use the KNN model to assign to each sample a new label. Use K = 1,3,5. Compute the empirical error and confusion matrix for each value of K.

Exercise 14

Generate 10 new samples using the PDF distribution of class C_1. Using the KNN algorithm trained before with the K related to the lowest empirical error value classify the new 10 samples. Compute again the empirical error and confusion matrix.

Chollet, F. (2018). Keras: The python deep learning library. Astrophysics Source Code Library.

Bishop, C. M. (2006). Pattern recognition and machine learning. springer.