Machine Learning and apps in AI: practice session 00

Martin Palazzo

Universite de Technologie de Troyes
Universidad Tecnologica Nacional Buenos Aires

martin.palazzo@frba.utn.edu.ar

November 29, 2022

Overview

1. Bayes Decision Theory
   - Univariate cases

2. Non parametric density estimation
   - Parzen Window
   - K Nearest Neighbors

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 and

$$x \in \mathbb{R}^d$$

with d as the dimensionality of the random variable $\mathcal{X}$

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

$$P(c_j \mid x) = \frac{p(x \mid c_j)P(c_j)}{p(x)} \tag{1}$$

$$\frac{p(x \mid c_0)P(c_0)}{p(x)} > \frac{p(x \mid c_1)P(c_1)}{p(x)} \tag{2}$$

$$p(x \mid c_0)P(c_0) > p(x \mid c_1)P(c_1) \tag{3}$$

Decision boundary by Bayes Rule

$$\frac{p(x \mid c_0)}{p(x \mid c_1)} > \frac{P(c_1)}{P(c_0)} \tag{4}$$

$$\frac{p(x \mid c_0)}{p(x \mid c_1)} > \frac{0.5}{0.5} \tag{5}$$

$$\frac{p(x \mid c_0)}{p(x \mid c_1)} > 1 \tag{6}$$

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

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

Non parametric density estimation

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

$$k(u) = \begin{cases} \frac{1}{h} \cdot \frac{n}{N} & \textit{if } x \textit{ falls within the } h \textit{ bin} \\ 0 & \textit{otherwise} \end{cases}$$

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: exercise

Supose $d = 1$, $h = 1$, $K = \mathbf{1}_{(-\frac{1}{2},\frac{1}{2})}$, 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: exercise

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

$$x_1 = 1 \quad x_2 = 1.5 \quad x_3 = 3 \quad x_4 = 5$$

by using a uniform kernel

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

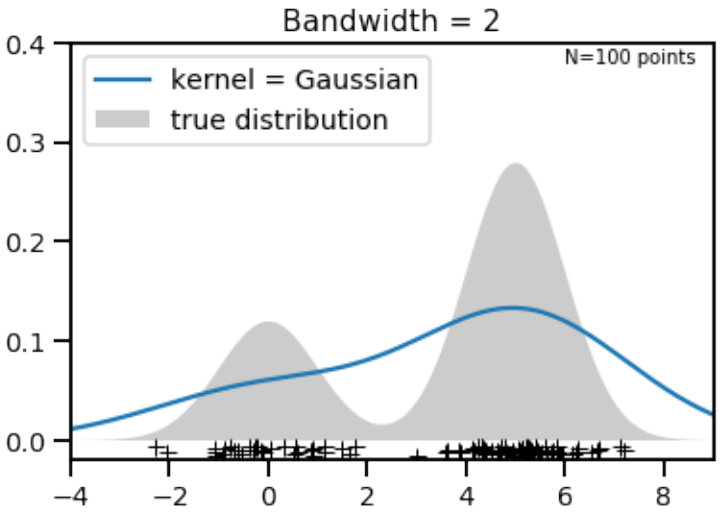
Parzen window for density estimation

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

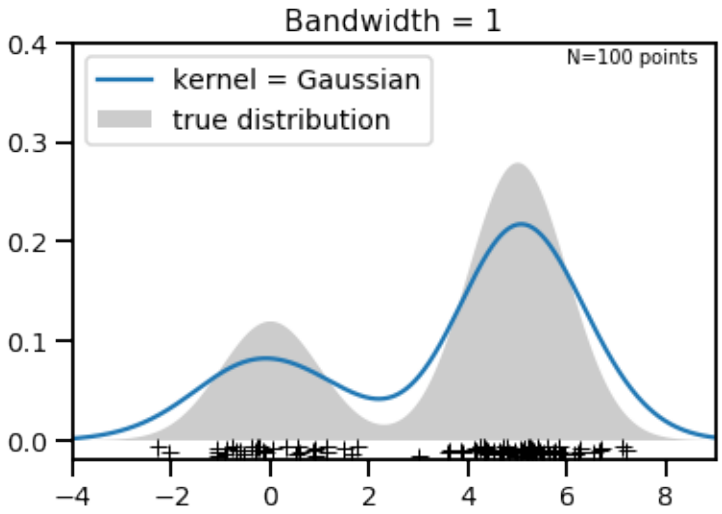
$$\hat{p}(x) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{\sqrt{2\pi h^2}}\textbf{exp}\left[\frac{||x - x_n||^2}{2h^2}\right]$$

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



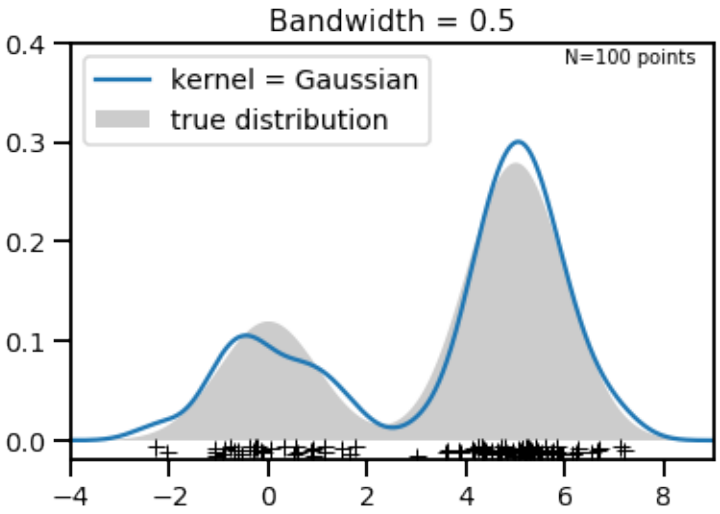
(a) h = 2



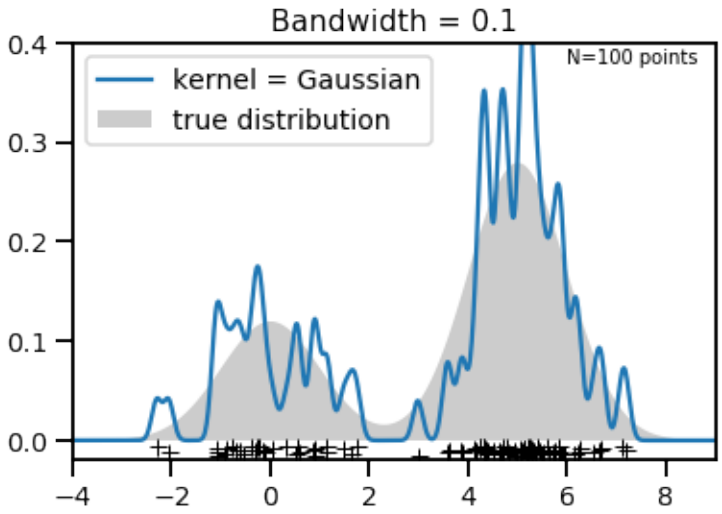
(b) h=5

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

Parzen window for density estimation



(a) h=0.5



(b) h=0.1

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.

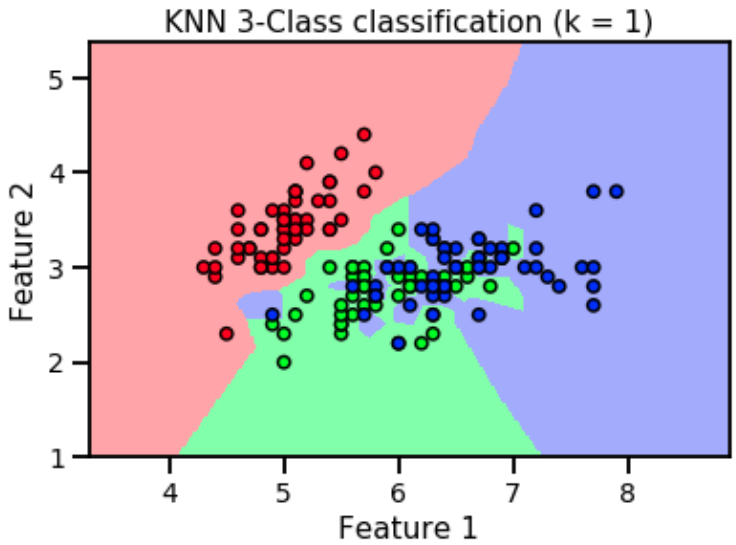
K Nearest Neighbors: Classification

Classification with KNN

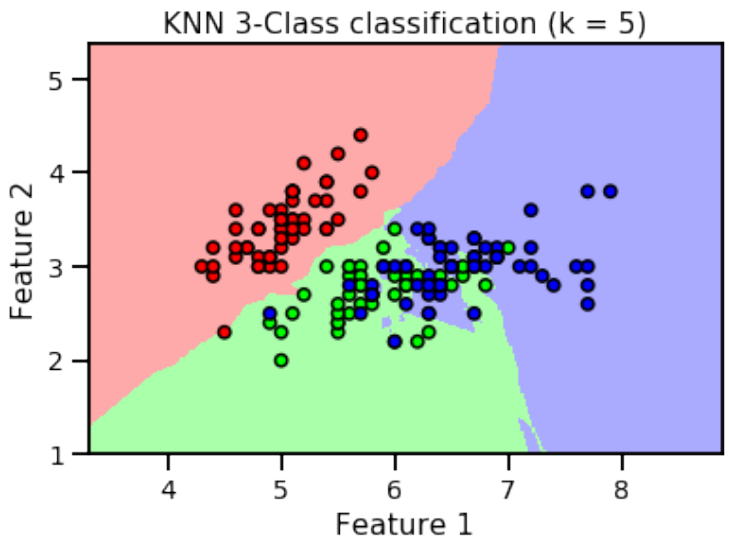
If we want to make classification with KNN, what we have to do is to asssign to a new sample X_p the highest value of the posterior probability $P(C_i|x)$ found. The same principle is used with Parzen window method.

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

K Nearest Neighbors: Classification



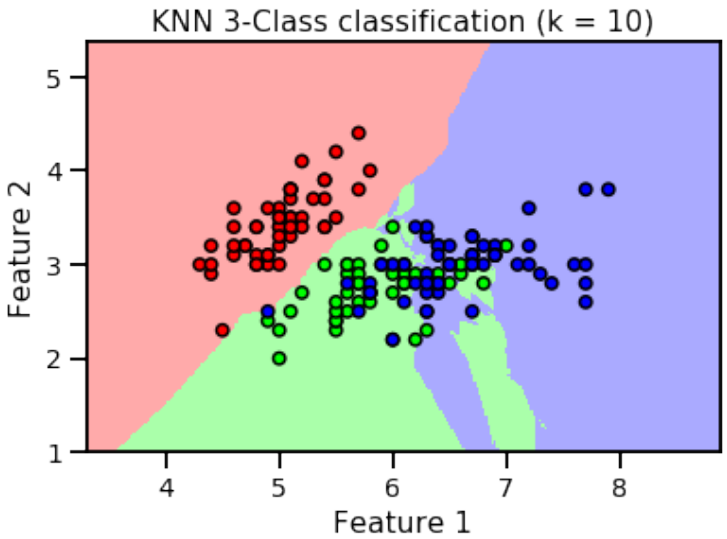
(a) K=1



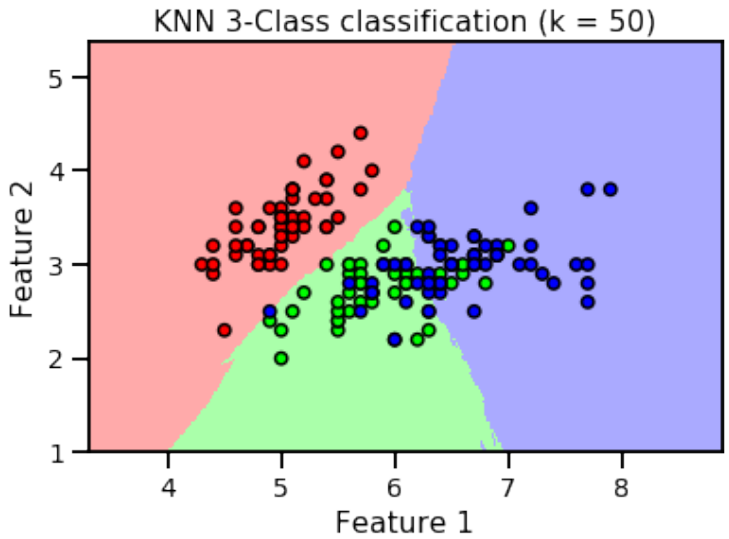
(b) K=5

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.