Machine Learning and Artificial Intelligence

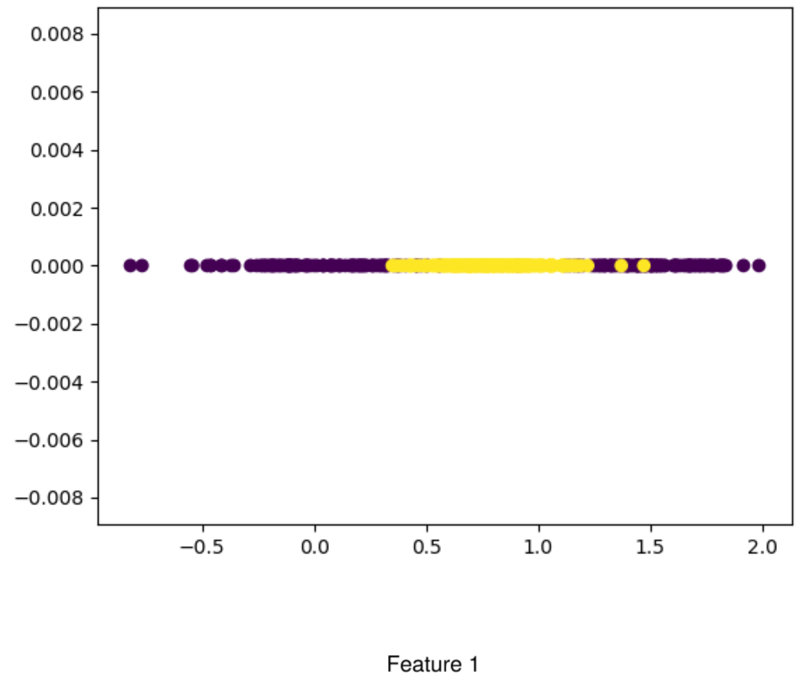
Lab 03 – K-Nearest Neighbours & Parzen Windows

23/03/2021

The problem

Starting point:

- Train data set (400 objects, 1 feature)
- Test data set (100 objects, 1 feature)
- Train data labels
- Test data labels (will never be considered except in final validation)



Non-parametric methods

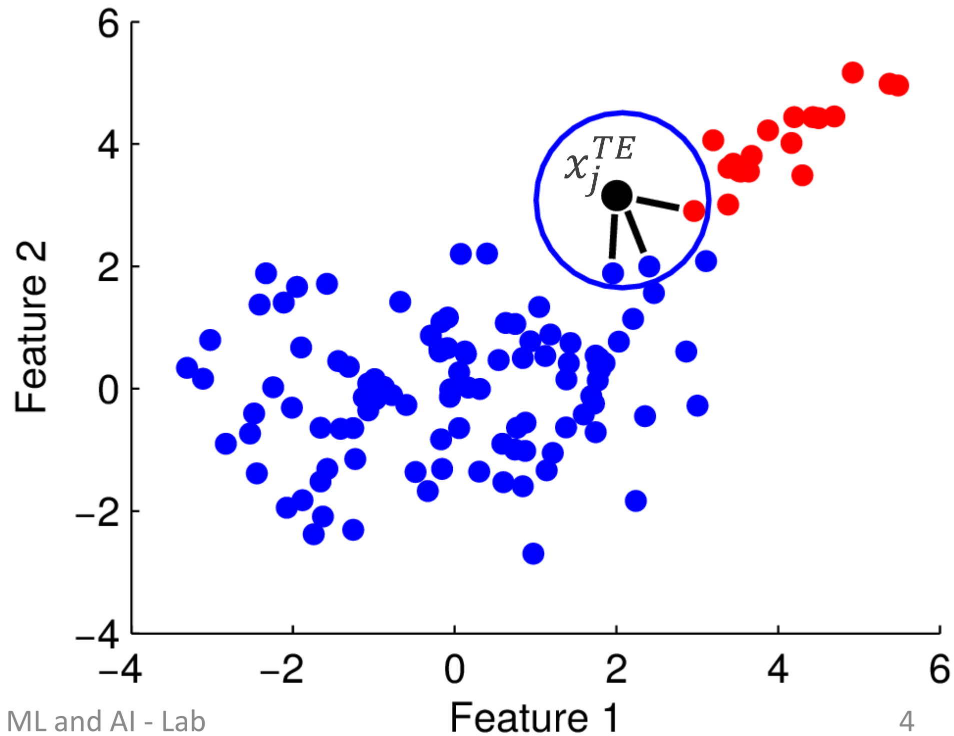
- Parametric methods make a *strong* assumption, which is that the shape of the probability densities is known.

- Idea of non-parametric methods: estimate the probability density function from the samples directly:
    1. K-Nearest Neighbours
    2. Parzen Windows

K-Nearest Neighbours

- Idea: given a test point x_j^{TE} to be classified, I consider the K train points closest to it, according to a certain metric.

- I assign to x_j^{TE} the most frequent class among these K points.



K-NN: In practice

Starting point:

- Train data set
- Train data labels
- Test data set
- Test data labels (will never be considered until final validation)

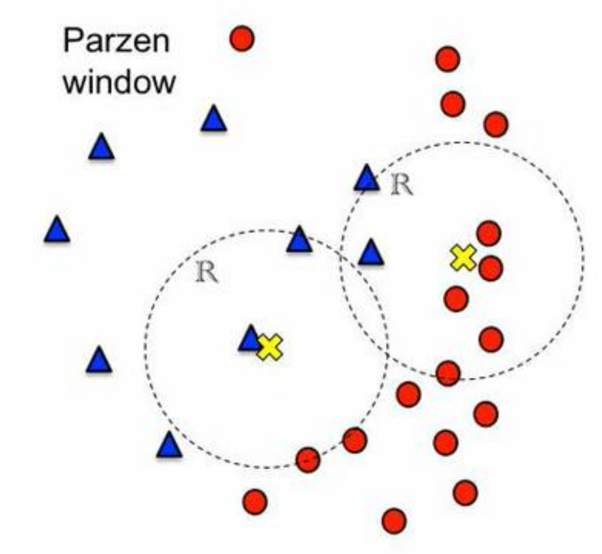
K-NN: In practice

- A priori, I decide how many nearest neighbours K to consider.
- Given a test point x_j^{TE}:
  - I calculate the distance between x_j^{TE} and all train points x_j^{TR}.
  - I order the distances and find the indices of the neighbouring K train points.
  - I check the labels of these K points and assign the most frequent class label as y_j^{TE}.

Exercises

Parzen Windows

- Idea: estimate the underlying probability distribution by looking at individual regions in space.
- If you need to estimate p(x=x0), you look at the region in a window centred on x0 and estimate from that region.
- This can be repeated for all the points to be classified.



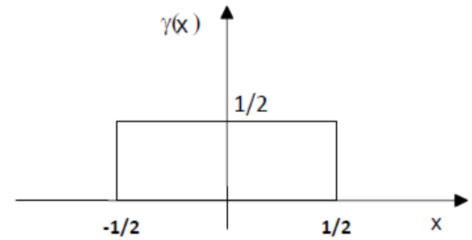
Parzen Windows: In practice

- A priori, I decide the width of the window h (e.g., h=0.2)

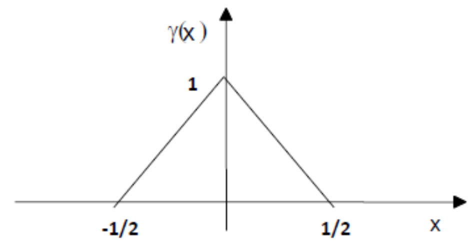
- I divide the training dataset in two, based on the classes (the labels of the test are unknown)
- Given a test point x_j^{TE}:
  - For every train point x_i^{TR} of class c:
    - Calculate the function $\gamma\left(\frac{x_j^{TE} - x_i^{TR}}{h}\right)$
  - Compute the probability $p\left(x_j^{TE}|\omega_c\right) = \frac{1}{N_c}\sum_{i=1}^{N_c} \frac{1}{h}\gamma\left(\frac{x_j^{TE} - x_i^{TR}}{h}\right)$
    - N_c : Number of points belonging to class c
  - Assign x_j^{TE} to the class with max probability

Viable functions for γ

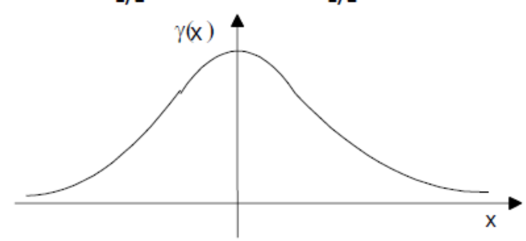
1) $$\gamma(\mathbf{x}) = \begin{cases} 0{,}5 & |\mathbf{x}| \le 1 \\ 0 & |\mathbf{x}| > 1 \end{cases}$$ Rectangle



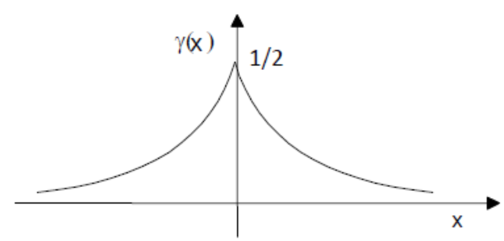
2) $$\gamma(\mathbf{x}) = \begin{cases} 1 - |\mathbf{x}| & |\mathbf{x}| \le 1 \\ 0 & |\mathbf{x}| > 1 \end{cases}$$ Triangle



3) $$\gamma(\mathbf{x}) = (2\pi)^{-\frac{1}{2}} e^{-\left(\frac{\mathbf{x}^2}{2}\right)}$$ Gaussian



4) $$\gamma(\mathbf{x}) = \frac{1}{2} e^{-|\mathbf{x}|}$$ Exponential (decreasing)



Exercises