Pattern Recognition Winter Term 2019

Institute of Neural Information Processing
PD Dr F. Schwenker

Assignment 4 (Submission until November 26, 2019)

Exercise 1 (3 points): Maximum Likelihood Parameter Estimation

In this exercise, we want to estimate the parameters of a probability function by using the maximum likelihood method. We can use this method if we know (or assume) the type of a distribution but don't have a clue about its parameters. Then, the general idea is to take a random sample and choose the parameters with the highest likelihood of producing the data.

As an example, suppose you participate in a lottery where each ticket is either a winning or a blank. You are, of course, interested in the event $A =$ *The lottery ticket rewards you with a price.* You don't know anything about the distribution of the lottery tickets, but you are interested in figuring out how long it takes to get a price. For this, you define a random variable $X =$ *Number n of lottery tickets to buy until A happens*, i.e. what is the probability that the n-th lottery ticket is a winner?

This an ideal problem for the geometric distribution where the probability mass function is defined as

$$P(X = n) = p(1 - p)^{n-1}.$$

Note that this is a discrete distribution with only one degree of freedom (the success probability $p = P(A) \in]0; 1[$). It returns the probability that the first success happens in the n-th trial ($n = 1, 2, \ldots$). Therefore, it suits perfectly to the random variable X.

Aglow with enthusiasm, you take three runs buying lottery tickets. In the first run, you get a price for the second ticket. In the second run, you are lucky and already get a price for the first ticket. But luck leaves you in the last run where you need to buy five tickets before getting a price. Summarizing, we have

$$n = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 5 \end{pmatrix}$$

with n_i denoting the number of tickets required to get a price in the i-th run. Assume that the runs are independent of each other and that the n_i arise from identically distributed random variables X_i.

1. [Pen and Paper] The goal of the likelihood function $L(p)$ is to find the most probable parameter p which suits the sample vector n. For this, we need to consider the probability of the joint occurrence of all runs X_i

   $$L(p) = P(\{X_1 = n_1\} \cap \{X_2 = n_2\} \cap \{X_3 = n_3\}).$$

   Make this formula explicit and simplify as much as possible. Hint: remember that we assumed independent and identically distributed variables.

2. [Pen and Paper] Find the estimate of the success parameter $\hat{p}$ by maximizing $L(p)$. You can assume that the estimate $\hat{p}$ is a maximum and not a minimum or saddle point (this is also verified visually in the next step). Hint: $p = 0$ and $p = 1$ are excluded by definition.
3. [Python] Plot $L(p)$ in the range $[0; 1]$ and label the estimated parameter $\hat{p}$.

Exercise 2 (3 points): Parzen Window (1D) [Pen and Paper]

Sometimes, we don't even know the type of probability density function (PDF) which produced the observed data. In these cases, we can use non-parametric methods like the Parzen window estimator (kernel density estimation). Let $x_i \in \mathbb{R}^d$ $(i = 1, \ldots, N)$ denote our samples from the dataset. Then, the estimate of the PDF is given by

$$\hat{p}(x) = \frac{1}{h^d \cdot N} \sum_{i=1}^{N} K\left(\frac{x_i - x}{h}\right) \tag{1}$$

with the side length h, N total number of data points from the d-dimensional space and K as the kernel function. In this exercise, we want to get used to this method by analysing a one-dimensional example.

1. Take a look at this animation[1] which uses standard normal Gaussian functions as kernels on a small dataset $X = (x_1, x_2, x_3, x_4) = (1, 1.5, 3, 5)$ which consists only of $N = 4$ values. Describe the effect of increasing side length h on
   a) the kernel functions $\tilde{K}_G(x)$ shown in the bottom of the graph,
   b) the histogram and
   c) the resulting probability estimation $\hat{p}(x)$.
2. What do you think is a suitable value of h for the dataset X?
3. Now, suppose that instead of Gaussians uniform distributed kernels centred at the origin are used

$$K_U(x) = \begin{cases} 1, & -0.5 \leq x \leq 0.5 \\ 0, & \text{else} \end{cases}$$

   Draw the probability estimate $\hat{p}(x)$ for the following three cases:
   a) $h = 0.5$
   b) $h = 1$
   c) $h = 2.5$

[1] https://milania.de/blog/Introduction_to_kernel_density_estimation_%28Parzen_window_method%29#fig:ParzenWindow_AnimationEstimateGaussian

Exercise 3 (4 points): Parzen Window (2D) [Python]

In this part, we want to apply the Parzen window method to a more complicated example. We want to find a probability estimate for the two-dimensional dataset shown in Figure 1. As kernels, we are using multivariate standard normal Gaussian functions parameterized by

$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2}$$

Note: this exercise is best implemented in a Jupyter Notebook to leverage some of the interactive functionalities.

1. The data is stored in the attached `data.pickle` file. You can load it in your Python program via

   ```
   import pickle
   data = pickle.load(open('data.pickle', 'rb'))
   ```

   The file contains 200 data points organized in a matrix of size 200×2, i.e. each row corresponds to one observation point.

2. Define a standard normal kernel with the parameters of Equation 2. You can create a distribution object with the function `multivariate_normal` from the `scipy.stats` package. Make sure you know how to retrieve multiple density values at once, e.g. for the points $x_1 = (0, 0)$ and $x_2 = (1, 1)$ at the same time.

3. Write a function `estimateProbability(x, h)` which computes the density $\hat{p}(x)$ for an arbitrary point $x \in \mathbb{R}^2$ based on the side length h, i.e. Equation 1 with $d = 2$.

4. Calculate and plot $\hat{p}(x)$ in the range $(x, y) \in [-400, 400] \times [-400, 400]$ with step sizes of $\Delta x = \Delta y = 10$. Play around with different values of the side length h. Some general hints for this part:

   - The function `Axes3D.plot_surface` from the `mpl_toolkits.mplot3d` package creates a surface plot which can be used to visualize the resulting PDF.
   - Add the magic command `%matplotlib notebook` to interact with the 3D plot (e.g. rotate with the left mouse button, zoom with the right mouse button).
   - It is possible to enrich the user experience of Jupyter Notebooks with interactive widgets. In our case, a simple slider (`interact` function) may be helpful to test different values of the side length h.

5. Based on your results, which side length h would you choose as a good estimate for the data?

6. You are listening to a discussion between colleagues about the Parzen window estimator and especially about the side length h. You hear how one of them is arguing "If you have no idea which side length to choose, just use $h = 1$. This will always lead to a good approximation.". What do you think about this statement? Consider what you chose for h in this and the previous exercise.

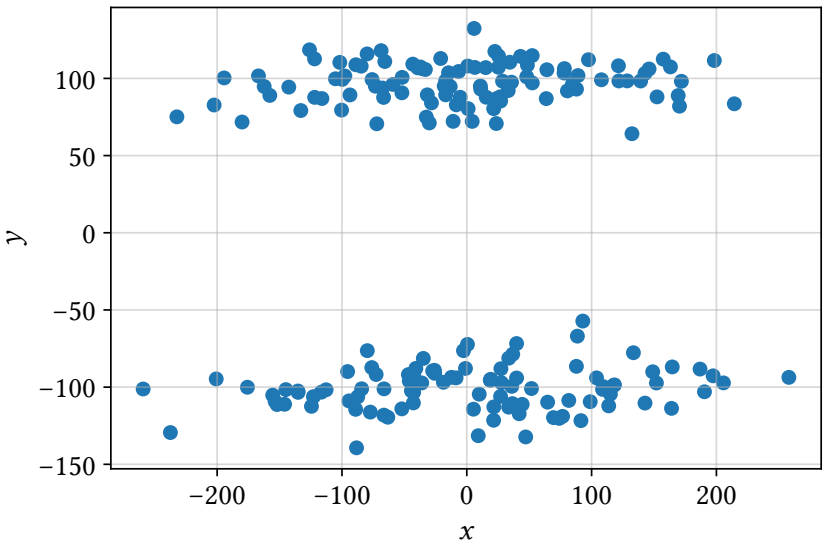

Figure 1: Example data used in Exercise 3.