Artificial Intelligence and Machine Learning

Exercises – Non-parametric density estimation and EM

Question 1 ⊛

You use a k-nearest neighbors classifier and set $k := N$, where N is the total number of data points in the dataset. Which class is predicted by the classifier?

Question 2 ⊛

The decision boundary shown in figure 1 was generated by a k-nearest neighbors classifier.

1. How do you rate the performance of the classifier?
2. What might be problems and how could they be mitigated?
3. Can you guess the value of k which was used? *Please explain your answer!*

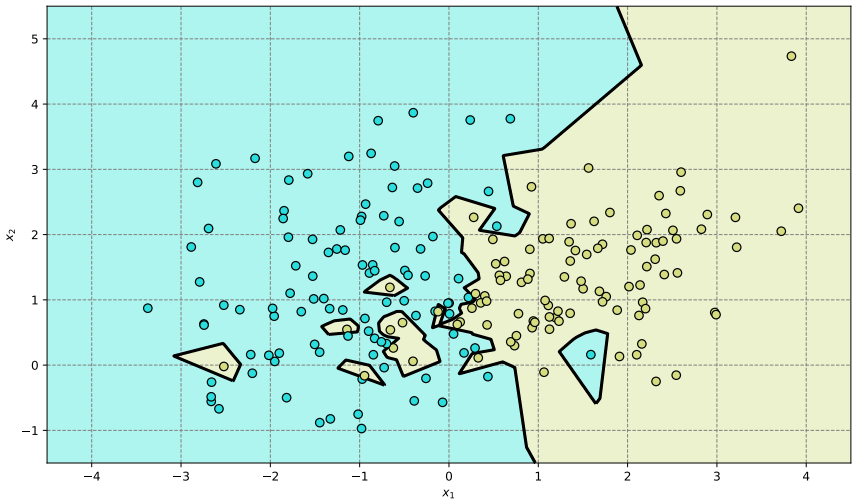


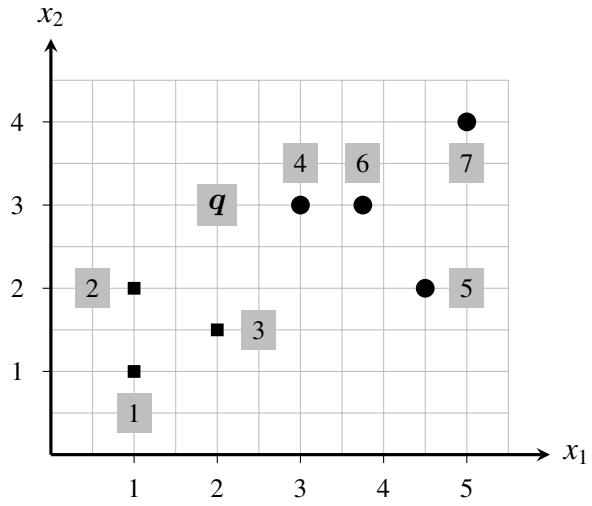
Figure 1: Decision boundary of a k-nearest neighbors classifier.

Question 3 (Non-parametric methods) ⊛

You have a dataset consisting of 500,000 data points. Your boss suggests to use a non-parametric method for classification (e. g. a k-nearest neighbors classifier). What does the term *non-parametric* mean? Do you agree with your boss? *Please explain your answer!*

Question 4 (k-nearest neighbors algorithm) ⊛

The examples in the training dataset as shown in figure 2 below belong to either of the two classes ■ or ●. Your goal is to classify the unknown data point $\boldsymbol{q} := \begin{pmatrix} 2 & 3 \end{pmatrix}^\top$ using the k-nearest neighbors algorithm. You choose $k := 3$.



Row	x_1	x_2	Class
1	1.00	1.00	■
2	1.00	2.00	■
3	2.00	1.50	■
4	3.00	3.00	●
5	4.50	2.00	●
6	3.75	3.00	●
7	5.00	4.00	●

Figure 2: Illustration of the training dataset.

1. Calculate the prediction

   (a) using the *Manhattan distance*, and

   (b) using the *Euclidean distance*.

   Do both distance metrics lead to the same result?

2. Suppose you had chosen $k := 7$. Which class would have been predicted? What problem do you see?

3. Illustrate two possible *tie breaking strategies* in case that both classes appear equally often in the neighborhood of $\boldsymbol{q}$.

Question 5 ⊛

Tick the correct statements concerning the k-nearest neighbors algorithm!

- □ The k-nearest neighbors algorithm is model-based.
- □ k can be determined using the validation set.
- □ The choice of k does not have a noteworthy effect on the predictions.
- □ Too large of a k leads to overfitting.
- □ The algorithm is an instance of lazy learning.

- ☐ The training phase is computationally expensive and time consuming.
- ☐ The prediction of unseen data points is computationally expensive and time consuming.
- ☐ k should be termined on the training set.

Question 6 (Implementing a kernel density estimator)

Implement kernel density estimation for the one-dimensional dataset produced by the code snippet below. Specifically, implement

1. the Parzen window (try different window sizes),
2. the Gaussian kernel (try different kernel widths), and
3. the k-nearest neighbors approach (try different values for k).

Finally, plot the results!

```
import numpy as np

np.random.seed(20)

# Gaussian 1
mu = 2.0
sigma = 0.1
X1 = np.random.normal(mu, sigma, 50)

# Gaussian 2
mu = 4.0
sigma = 0.5
X2 = np.random.normal(mu, sigma, 50)

X = np.concatenate((X1, X2), axis=0)
```

Question 7 (Implementing EM for Gaussian mixture models)

Implement the Expectation-Maximization (EM) algorithm for Gaussian mixture models and apply it to the dataset generated by the below code snippet. To evaluate the multivariate Gaussian distribution, you may want to use the library function `scipy.stats.multivariate_normal`.

```
from sklearn.datasets import make_classification

np.random.seed(42)

# number of data points
N = 400
```

```
# number of features
M = 2

X, y = make_classification(
    n_samples=N,
    n_features=M,
    n_redundant=0,
    n_informative=2,
    n_clusters_per_class=1,
    n_classes=3,
    class_sep=1.25
)
```

Finally, plot the results!