Machine Learning (SS 23)

Assignment 03: Classification (Solution)

Mojtaba Nayyeri
Mojtaba.Nayyeri@ipvs.uni-stuttgart.de

Akram Sadat Hosseini
Akram.Hosseini@ipvs.uni-stuttgart.de

Nadeen Fathallah
Nadeen.Fathallah@ipvs.uni-stuttgart.de

Rodrigo Lopez Portillo Alcocer
rodrigo.lopez-portillo-alcocer@ipvs.uni-stuttgart.de

Tim Schneider
timphillip.schneider@ipvs.uni-stuttgart.de

Osama Mohammed
osama.mohammed@ipvs.uni-stuttgart.de

Daniel Frank
daniel.frank@ipvs.uni-stuttgart.de

This assignment sheet consists of 7 pages with 4 Questions and 1 Task:

Submit your solution in ILIAS as a single PDF file.[1] Make sure to list full names of all participants, matriculation number, study program and B.Sc. or M.Sc on the first page. Optionally, you can *additionally* upload source files (e.g. PPTX files). If you have any questions, feel free to ask them in the exercise forum in ILIAS.

Submission is open until Monday, 8th May 2023, 12:00 noon.

[1] Your drawing software probably allows to export as PDF. An alternative option is to use a PDF printer. If you create multiple PDF files, use a merging tool (like pdfarranger) to combine the PDFs into a single file.

Question 1: kNN for Data Classification

Consider a classification problem with one input variable (attribute) x, and the following training data points:

attribute	x	-3	-1	1	7.2
target	y	0	1	1	0

- Find the number of miss-classified points in the training data for each of the following classifiers (use the Euclidean distance function): (1) K-nearest Neighbor with K = 1 and (2) K-nearest Neighbor with K = 3. State the reasons in each case.

Solution:

1. - The one-dimensional dataset could be visualized as shown in Fig.1. The blue points represent the points with target value of (-1) and the red points represent the points with target value of (+1).

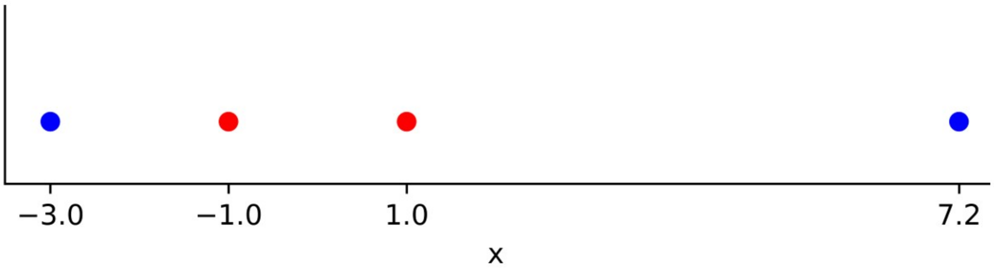


Figure 1 One-dimensional dataset

   - We will check the nearest K neighbors for each point using the Euclidean distance to find the miss-classified points.

Point (x_1)	Euclidean distance with other points	Nearest Point	Predicted Class	Actual Class
P_1: -3	$d(P_1, P_1) = \sqrt{((-3) - (-3))^2} = 0$ $d(P_1, P_2) = \sqrt{((-3) - (-1))^2} = 2$ $d(P_1, P_3) = \sqrt{((-3) - (1))^2} = 4$ $d(P_1, P_4) = \sqrt{((-3) - (7.2))^2} = 10.2$	P_1	-1	-1
P_2: -1	$d(P_2, P_1) = \sqrt{((-1) - (-3))^2} = 2$ $d(P_2, P_2) = \sqrt{((-1) - (-1))^2} = 0$ $d(P_2, P_3) = \sqrt{((-1) - (1))^2} = 2$ $d(P_2, P_4) = \sqrt{((-1) - (7.2))^2} = 8.2$	P_2	$+1$	$+1$
P_3: 1	$d(P_3, P_1) = \sqrt{((1) - (-3))^2} = 4$ $d(P_3, P_2) = \sqrt{((1) - (-1))^2} = 2$ $d(P_3, P_3) = \sqrt{((1) - (1))^2} = 0$ $d(P_3, P_4) = \sqrt{((1) - (7.2))^2} = 6.2$	P_3	$+1$	$+1$
P_4: 7.2	$d(P_4, P_1) = \sqrt{((7.2) - (-3))^2} = 10.2$ $d(P_4, P_2) = \sqrt{((7.2) - (-1))^2} = 8.2$ $d(P_4, P_3) = \sqrt{((7.2) - (1))^2} = 6.2$ $d(P_4, P_4) = \sqrt{((7.2) - (7.2))^2} = 0$	P_4	-1	-1

Therefore, there are 0 miss-classified points in the case of KNN classifier at K $= 1$.

However, at K $= 3$. In this case, the prediction is decided based on the majority voting.

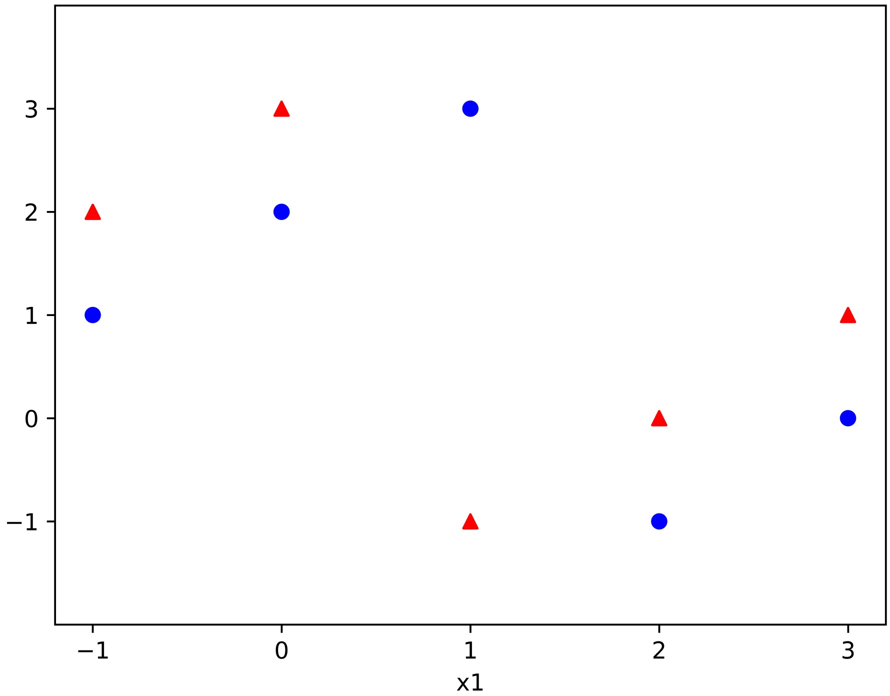
Point (x_1)	Euclidean distance with other points	Nearest 3 Points	Predicted Class	Actual Class
P_1: -3	$d(P_1, P_1) = \sqrt{((-3) - (-3))^2} = 0$ $d(P_1, P_2) = \sqrt{((-3) - (-1))^2} = 2$ $d(P_1, P_3) = \sqrt{((-3) - (1))^2} = 4$ $d(P_1, P_4) = \sqrt{((-3) - (7.2))^2} = 10.2$	P_1, P_2 and P_3	$+1$	-1
P_2: -1	$d(P_2, P_1) = \sqrt{((-1) - (-3))^2} = 2$ $d(P_2, P_2) = \sqrt{((-1) - (-1))^2} = 0$ $d(P_2, P_3) = \sqrt{((-1) - (1))^2} = 2$ $d(P_2, P_4) = \sqrt{((-1) - (7.2))^2} = 8.2$	P_1, P_2 and P_3	$+1$	$+1$
P_3: 1	$d(P_3, P_1) = \sqrt{((1) - (-3))^2} = 4$ $d(P_3, P_2) = \sqrt{((1) - (-1))^2} = 2$ $d(P_3, P_3) = \sqrt{((1) - (1))^2} = 0$ $d(P_3, P_4) = \sqrt{((1) - (7.2))^2} = 6.2$	P_1, P_2 and P_3	$+1$	$+1$
P_4: 7.2	$d(P_4, P_1) = \sqrt{((7.2) - (-3))^2} = 10.2$ $d(P_4, P_2) = \sqrt{((7.2) - (-1))^2} = 8.2$ $d(P_4, P_3) = \sqrt{((7.2) - (1))^2} = 6.2$ $d(P_4, P_4) = \sqrt{((7.2) - (7.2))^2} = 0$	P_2, P_3 and P_4	$+1$	-1

Therefore, there are 2 miss-classified points in the case of KNN classifier at K $= 3$.

Question 2: kNN and cross-validation

For the data given below with 10 points and 2 classes, answer the following questions:

1. What is the leave-one-out cross-validation error when using K-nearest neighbor classifier with K = 1?
2. Which of the following values of K leads to the minimum number of leave-one-out validation errors: 3, 5 or 9?



Solution:

1. The question asks what the leave-one-out cross-validation error is when using a K-nearest neighbor (KNN) classifier with K=1. KNN is a classification algorithm that works by finding the K nearest neighbors to a given data point in the training set, and then predicting the class of the data point based on the majority class of its K nearest neighbors.

   In this case, we have 10 data points, with two classes. We need to find the leave-one-out cross-validation error for K=1. This means we will train the KNN classifier on all the data except one point, and then use the classifier to predict the class of the remaining point. We repeat this process for all 10 data points, leaving out a different point each time. Use average performance.

   For K=1, the classifier simply assigns the class of the nearest neighbor to the test point. In this case, since we have two classes, the closest neighbor of each point will always be from the opposite class. This means that the KNN classifier will always misclassify each point, resulting in a total leave-one-out cross-validation error of 100%.

2. All given values of K lead to the same result: 10 errors similar to question 1.

Question 3: kNN for Image Classification

Research and discuss how you could use a k-nearest neighbor classifier for image classification. You should at least answer these questions:

- How do you represent the images?
- What distance function do you use?
- What decision rule do you use?

Provide an example for your representation of the images and how your classification decision is made based on the distance function and decision rule. Explain the advantages and disadvantages of your approach. Solution:

- How do you represent the images?

  Solution: To represent the images, we can extract a set of features that capture the important characteristics of the image. One popular approach is to use a bag-of-visual-words representation, which involves extracting local features (such as SIFT, SURF, or ORB) from the image, clustering them into a set of visual words using k-means, and counting the frequency of each visual word in the image. This results in a fixed-length feature vector for each image, which can be used for classification.

- What distance function do you use?

  Solution: The choice of distance function depends on the nature of the features. For example, if we use the bag-of-visual-words representation, we can use the Euclidean distance or cosine similarity to compare feature vectors.

- What decision rule do you use?

  Solution: The decision rule in KNN is to assign a new instance to the class that is most common among its k nearest neighbors. The value of k is a hyperparameter that needs to be tuned using a validation set.

  Here's an example to illustrate how image classification can be done using KNN:

  Suppose we have a dataset of images of fruits (apples, bananas, and oranges), each represented as a bag-of-visual-words feature vector of length 1000. We randomly split the dataset into training and test sets (80% for training and 20% for testing), and use KNN with k=5 to classify the test images.

  To classify a test image, we compute its feature vector and find its 5 nearest neighbors in the training set based on the Euclidean distance. We then assign the test image to the class that is most common among the 5 nearest neighbors. For example, if 3 of the neighbors are apples and 2 are bananas, we classify the test image as an apple.

  Advantages and disadvantages: The advantages of KNN for image classification are:

  - It can work well for high-dimensional feature spaces.
  - It can be used with any distance function and decision rule.
  - It can be easily extended to handle multi-class classification.

  The disadvantages of KNN for image classification are:

  - It can be computationally expensive, especially for large datasets.
  - It can be sensitive to the choice of hyperparameters such as k and the distance function.

- It can suffer from the curse of dimensionality when the feature space is very large.
- It can be affected by the presence of irrelevant or noisy features in the data.

Task 1: kNN and Classification

Please download the Jupyter notebook *assignment2.ipynb*. Follow the instructions in the Jupyter notebook.