Linnaeus University
Introduction to Machine learning, 2DV516
Jonas Nordqvist
`jonas.nordqvist@lnu.se`

Assignment 1: k-NN-classifier

Introduction

In this Assignment you will use MATLAB to handle a number of exercises related to k-nearest neighbors. The datasets used in the assignment can be downloaded in Moodle. For this first assignment you will also get three m-files in which there is a suggested structure for your m-files.

Submission Instructions

All exercises are individual. We expect you to submit one m-file for each exercise and provide all functions (as local functions, scripts or classes) used in the exercise[1].

Most exercises can be handled with a single m-file. Certain quantitative questions such as: *What is the expected number of stories in a 900 ft building?*, can simply be handled as a print statement in the m-file. More qualitative questions such as: *Motivate your choice of model.*, should be handled in a text file. (All such answers can be grouped into a single text-file)

Finally, keep all your m-, mat-, csv-, and text-files in a single folder named as `username_A1` and submit a zipped version of this folder.

k-NN classification and regression (Lecture 2)

Exercise 1: k-NN classification

In the following subexercises the dataset(s) referred to are found in data1.mat. In this exercise you will implement a k-NN classifier and perform your first classification task.

1. Plot the training data in a scatter plot using different colors or symbols for the two classes.

2. Implement the function `kNNclassify`, which takes as input an integer k, data matrix X, labels y a point z and outputs the classification of z by computing the mode of its k closest neighbors in X. Use the Euclidean metric as your distance.

3. Classify the point $(-17, 14)$ using the provided data file X for all $k \in \{1, 3, 5\}$. Note that points in $\mathbb{R}^2$ in Matlab can be written as `[ a b ]`. The classifications ought to be 1, 1, and 0 respectively.

4. Implement the function `kNNdrawBoundary`, which takes as input an integer k, data matrix X, labels y, which draws the decision boundary of the model `kNNclassify`. Use it to draw the decision boundary for all $k \in \{1, 3, 5\}$.

[1] If you are using local functions, this might lead to multiple copies of certain functions scattered across several m-files. That is okay.

5. Which of the of the values for $k \in \{1, 3, 5\}$ gives the smallest training error. Do you think this is the best choice of k in terms of generalization of the model to unseen points? Motivate your answer.

6. Change the metric used in your `kNNclassify` to construct `kNNclassify_taxi` to use the Taxi cab-distance or try some other distance metric, and redo exercise 3 and 4. Is the classifications for $(-17, 14)$ the same? Preferably try to plot both decision boundaries for the same k in the same plot.

Exercise 2: Multi-class k-NN

In the following subexercises the dataset(s) referred to is found in data2.mat. Note that any local functions used in Exercise 1 can simply be copy-pasted if needed. This exercise is similar to Exercise 1 in that it is classification of two dimensional data, but now there are four different classes, with labels in $\{0, 1, 2, 3\}$. In this exercise you are also provided a test set to use for testing your model.

1. Plot the training data in a scatter plot using different colors or symbols for the four classes.

2. Draw the decision boundary for all $k \in \{1, 3, 5, 7\}$. You might have to adjust your method from Exercise 1 to work for a multi-class problem.

3. Find the accuracy and error rate of your k-NN model using the provided test set. Plot the test error of different models versus k. Which model gives the smallest test error?

Exercise 3: k-NN regression

In the following subexercises the dataset(s) referred to are found in data3.mat. The datasets in data3.mat describes the price of 160 cars in terms of 13 continuous-valued features. Description of the data, its features and its source are found in the import-85.names-file in the data3.zip. However, note that we are only using a subset of the original data.

1. Implement the function `kNNregression`, which takes as input an integer k, data matrix X, labels y a point z and outputs the prediction of z by computing the mean of its k closest neighbors in X. Use the Euclidean distance as metric.

2. To test your regression model by predicting the price of

   `z = [100,180,70,50,3000,130,6.5,3,7.5,160,5000,20,20]`,

   using $k = 12$. The point is included in data3.mat. If your implementation is correct it should output `1.6156e+04` $\approx$ \$16000.

3. Use your `kNNregression.m` to measure the mean squared error against the test set. Use all $k \in [1, 10]$ and plot the MSE against k. Which k minimizes the test error?

Exercise 4: Visualize k-NN regression (VG exercise)

This exercise is optional for passing the assignment, but required in order to obtain higher grades (A-B).

For this exercise you will have to find (or create) your own dataset. The restriction is that there should be precisely two features (preferably real-valued), and corresponding real-valued labels.

The exercise is to implement a function which takes the your data and an integer k as input and returns some visualization of the regression surface. This could for instance be a heatmap, as illustrated in Lecture 2, or any other way. Feel free to be creative! Motivate your choice in text.