Kristianstad University
www.hkr.se

Fredrik Frisk
Computer Science Department

Content and purpose of this lab

The main focus of the lab is to do classification.

Save the code you are writing in this lab for future use. To pass the lab you need to solve/program the different bullet points and be able to explain your results. If you are not finished with the all the bullet points the remaining ones are a part of the required preparations for part 3 of the labs. The lab report in the end is an individual report, but you are allowed to work two and two with one exception all of you have to record your own sensor data.

Preparations

You need to finish part one of the lab and be able to show the result in the beginning of the lab.

It is important that you can show the pairplots with all classes included and with the two different sets of features.

We will use the K-nearest neighbor algorithm in this lab. You need to prepare a bit by reading selected parts of the text in the following links

Read the subchapters Algorithms and parameter selection in the Wiki page below.

https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Read the subchapter 1.6.2 Nearest Neighbor Classification on the scikit-learn page below.

https://scikit-learn.org/stable/modules/neighbors.html#classification

Run the following example.

https://scikit-learn.org/stable/auto_examples/neighbors/plot_classification.html#sphx-glr-auto-examples-neighbors-plot-classification-py

Below is a simple explanation of the KNN algorithm.

https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

Classify using KNN – Iris data set

During the preparation you were working with the Iris example using the KNeighborsClassifier from scikit-learn. In the following bullet points you should use the Iris dataset. It is enough to look at the validation set. For example use cross validation for validate the performance. You are free to use any performance measure you want, but a good starting point is the accuracy.

- What is the difference between the two plots in the Iris example above?
- Calculate the confusion matrix and explain the result. Why do you get the errors you get?
- Change the hyperparameter *k*. Which is the optimal value? What is the drawback of choosing a to large *k* or a to small *k*?
- One hyperparameter is weights. What is the difference between ‘uniform’ and ‘distance’?
- In the Iris data set there is four different features. Which are these?
- Do you need all features? What happens performance wise if you only use two features? Which to do you choose and why?

There is a preprocessing module in sklearn. One very useful method is the standardscaler

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html?highlight=standard%20scaler#sklearn.preprocessing.StandardScaler

- Use the standarscaler on the iris data set. Do you get a different result now? Can you explain the difference or that there is no difference?

KNN once again with your own recorded data

Now it is time to work with your own data. You should use the KNN algorithm as previously. To start with you should work with the data stored as below

Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6
a_x	a_y	a_z	m_x	m_y	m_z

That is 6 features. You have stored the data in three classes in the previous lab. These are your target classes in this part of the lab.

1. Should you use the standardscaler or not when you work with the data? You need to motivate your answer and explain why

2. Train the KNN classifier with your training data (cross validation!) Choose the *k* value that is optimal. You should present the result accuracy as a function of *k*. Present the confusion matrix and explain the result.

3. Test the KNN classifier with the optimal *k* using the test data set. Present the accuracy and confusion matrix. Explain the result and compare it with the validation set.

4. Redo point 2 and 3 but only use feature 1-3. Compare with the result in 3 with point 3 above.

5. Redo point 2 and 3 but only use feature 4-6. Compare with the result in 3 with point 3 above.

KNN– with transformed recorded data.

You should use the KNN algorithm as previously. To start with you should work with the data stored as below. That is the length of the vectors and the angles defined in part 1.

Feature 1	Feature 2	Feature 3	Feature 4
|a|	a_θ	|m|	m_θ

Work through the points 1 – 6 again with the transformed data. The points 4 and 5 the features at hand will be 1 – 2 respectively 3-4.

KNN – with four classes

Previously you have worked with three classes:

- Stand
- Sit down
- Ly down

As you have maybe noted you have “corrupted” data in the beginning and end of all files. These data points correspond to some movement, you are starting or stopping the recording. Previously we have removed this data. Instead keep all the data and put the “corrupted” data as a new class. Maybe you call this class corrupted or movement.

Use KNN once again to classify these four classes.

- Choose one of the feature sets (x, y, z) or (magnitude, angle). Motivate your choice.l
- Work through the point 1-3 as previously.