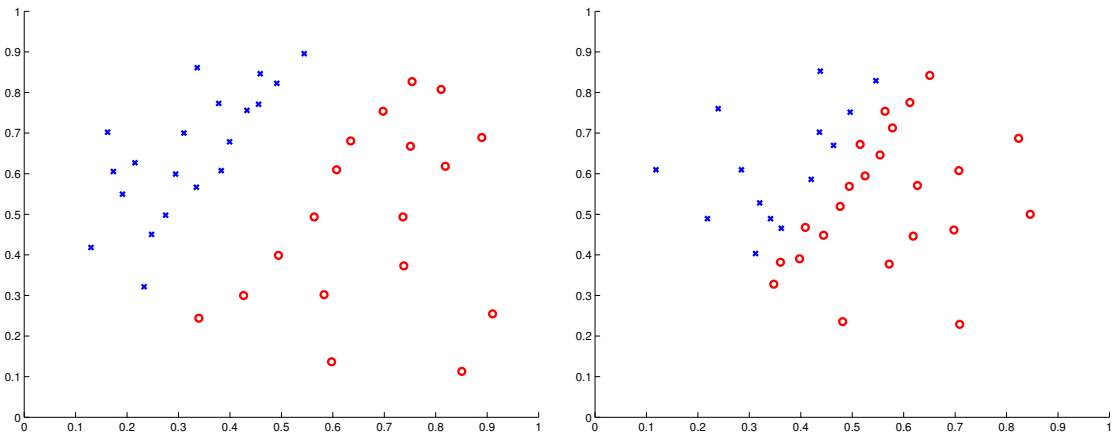
George Lydakis <lydakis@vision.rwth-aachen.de>
Stephanie Kaes <kaes@vision.rwth-aachen.de>

Exercise 2: Least Square Linear Classifiers

due **before** 2023-20-11

Important information regarding the exercises:

- The exercises are not mandatory. Still, we strongly encourage you to solve them! All submissions will be corrected. If you submit your solution, please read on:
- Use the Moodle system to submit your solution. You will also find your corrections there.
- Due to the large number of participants, we require you to submit your solution to Moodle **in groups of 3 to 4 students**. You can use the **Discussion Forum** on Moodle to organize groups.
- If applicable submit your code solution as a zip/tar.gz file named `mn1_mn2_mn3.{zip/tar.gz}` with your **matriculation numbers** (`mn`).
- Please do **not** include the data files in your submission!
- Please upload your pen & paper problems as PDF. Alternatively, you can also take pictures (.png or .jpeg) of your hand written solutions. Please make sure your handwriting is legible, the pictures are not blurred and taken under appropriate lighting conditions. All non-readable submissions will be discarded immediately.



(a) Training data (b) Test data

Figure 1: Datasets for linear classifier

Question 1: Least Square Linear Classifier ..... $(\Sigma = 5)$

Train a least-squares linear classifier on the 2D training data (c.f. Figure 1a) and test it on the training and test set (c.f. Figure 1b). The data for this exercise are stored in the files `lc_train_data.dat`, `lc_train_label.dat`, `lc_test_data.dat` and `lc_test_label.dat`. To this end write two functions:

(a) The function **(2 pts)**

```
1 def leastSquares(data, label):
2     # Sum of squared error shoud be minimized
```

```
    #
    # INPUT:
    # data         : Training inputs  (num_samples x dim)
    # label        : Training targets (num_samples x 1)
    #
    # OUTPUT:
    # weights      : weights   (dim x 1)
    # bias         : bias term (scalar)
    return weight, bias
```

that trains a least-squares classifier based on a data matrix `data` and its class label vector `label`. It provides as output the linear classifier weight vector `weight` and bias `bias`.

(b) The function **(2 pts)**

```
def linclass(weight, bias, data):
    # Linear Classifier
    #
    # INPUT:
    # weight       : weights                (dim x 1)
    # bias         : bias term              (scalar)
    # data         : Input to be classified (num_samples x dim)
    #
    # OUTPUT:
    # class_pred   : Predicted class (+-1) values  (num_samples x
        1)
    return class_pred
```

that classifies a data matrix `data` based on a trained linear classifier `weight, bias`.

(c) Run the script `apply`. This function loads the train and the test datasets. It first trains the linear classifier on the training data and then applies it on both the training and the test datasets. Analyze the classification plots for both the datasets. Are the sets optimally classified? Explain! **(1 pt)**