# Machine Learning Lab Assignment: Classification of physical activities with Logistic Regression

## Objectives

* Feature selection
* Cost function and gradient function construction
* Finding optimized λ value
* Evaluation of the linear/ logistic regression performance
* Solving the bias or variance problems

## Preparation

To be able to perform the lab, you need to go through the free coursera course on Machine Learning.

<https://www.coursera.org/learn/machine-learning/home/welcome>

You need to follow week 1, week 3 and week 6. You should also look at the programming exercises in these weeks as you can reuse this code for the Assignment.

You also need to install MATLAB on your laptop.

## Dataset

The dataset is a public collection of labelled accelerometer data recordings to be used for the creation and validation of acceleration models of simple physical activities [1]. The dataset is composed of the recordings of 6 activities (walking, walking-upstairs, walking-downstairs, sitting, standing and lying), performed by 30 volunteers, with the ID of 1~30, which is saved in the file named subject.mat.

1. Activity label information:

In this dataset, there are 6 activities, which are labeled separately as number:

1 – Walking

2 – Walking upstairs

3 – Walking downstairs

4 – Sitting

5 – Standing

6 – Laying

The information is saved in the file named Label.mat.

2. Device specifications

|  |  |
| --- | --- |
| 1) Type: | tri-axial accelerometer integrated in Samsung Galaxy S2 |
| 2) Output sample rate: | 50 Hz |
| 3) Location: | Attached to the waist of the user |

3. Data format

The data are stored in the text format.

1. Original dataset folder: The raw acceleration data is saved in the folder named Original dataset, with three documents in it: Acc\_x.mat, Acc\_y.mat, Acc\_z.mat, referring to the raw acceleration data at x, y, z-axis, respectively. Each row of the data in these files represents for the raw acceleration data of one axis corresponding to one specific activity of one subject. As the data has already been preprocessed using the fixed width length window (2.56 seconds) with 50% of overlapping, the size of each row is the same, with 128 data points.

2. After the signal segmentation, features are extracted, firstly, using a median filter and a 3rd order low pass Butterworth filter with passing frequency of 20 Hz to remove noise. Then the body and gravity acceleration signals are separated using a low pass Butterworth filter with a passing frequency of 0.3Hz.). There are in total 561 features extracted from the body or gravity acceleration signal, saved in the file named ‘totalfeatures.mat’, with the illustration saved in ‘featureinfo.txt’ and ‘features.txt’. These three files are stored in the folder named ‘561 features’.

In this exercise, 8 of them are selected, saved in ‘Features.mat’, and those features refer to:

1. Max value of the acceleration at the x-axis of body acceleration.
2. Max value of the acceleration at the x-axis of gravity acceleration.
3. Max value of the acceleration at the y-axis of gravity acceleration.
4. Max value of the acceleration at the z-axis of gravity acceleration.
5. Min value of the acceleration at the x-axis of gravity acceleration.
6. Min value of the acceleration at the y-axis of gravity acceleration.
7. Min value of the acceleration at the z-axis of gravity acceleration.
8. Signal magnitude area of the 3-axis of gravity acceleration.

(1)

sma- signal magnitude area, ax-acceleration at x-axis, ay-acceleration at y-axis, az – acceleration at z-axis.

3. File named ‘Subject’ saves the label of each subject.

4. File named ‘Label’ saves the label of each activity.

For example, in the Features.mat file, the data at the first row and second column refers to the feature for the max value of the acceleration at x-axis of gravity acceleration collected from the standing activity of subject 1. For further information, please read README.txt. For this exercise, you can only use the three files: ‘Features. mat’, ‘Label. mat’, ‘subject. mat’.

**MATLAB functions recommend** *load ()*, for importing file.

## Exercise 1: Feature selection

After knowing about all the features, you need to select the features that perform best. Firstly, visualize these features in a 2D space and choose two features. The whole dataset should be separated into **training, validation** and **testing** **dataset**, with the ratio of **0.4:0.3:0.3**.

Human activity is typically classified by a binary classifier (a classifier that only detects 2 classes), also called “one vs rest”. In this case, the activity of interest is class 1 and all other activities are class 0. In this assignment, each group can select only one “interested” activity for classification. The rest is assigned to class 0.

This can be done as follows:

>>y1 = y==3;

Here 3 indicates the 3rd class, walking downstairs.

Choose two features and visualize these in a 2D space.

**MATLAB function recommend**: *gplotmatrix(), double().*

Check whether you see a clear difference between your activity and the other activities. If there is a clear difference, these features will be suitable for creating a model. If not, choose 2 different features. An ideal separation between classes would look like the example in Figure 1.  It could well be that a 2D feature space does not yield perfect separation between the two classes.

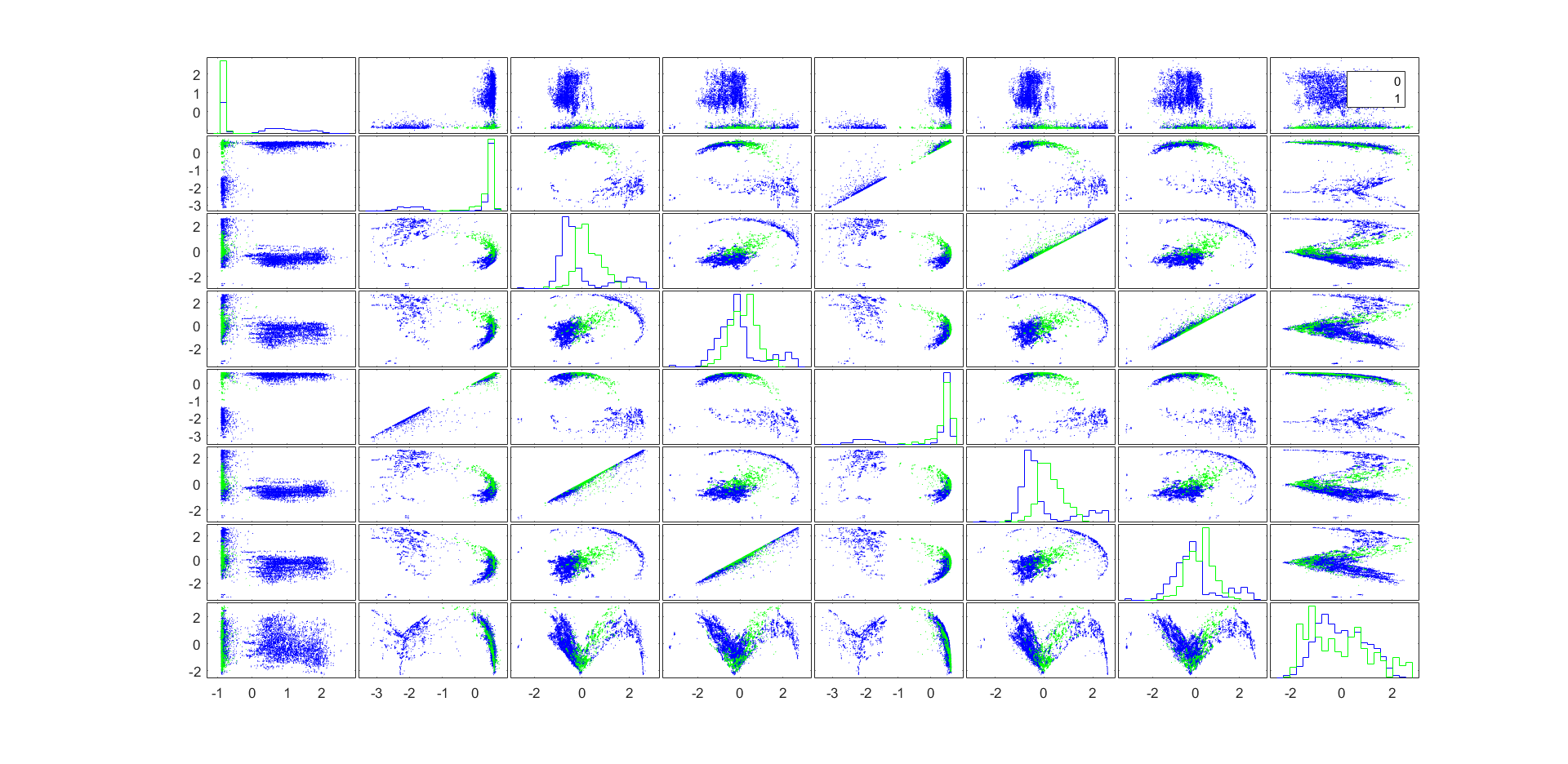


Figure 1. Visualization feature selection

Choose binary classification sitting vs the rest. Also use features 4 and feature 6.

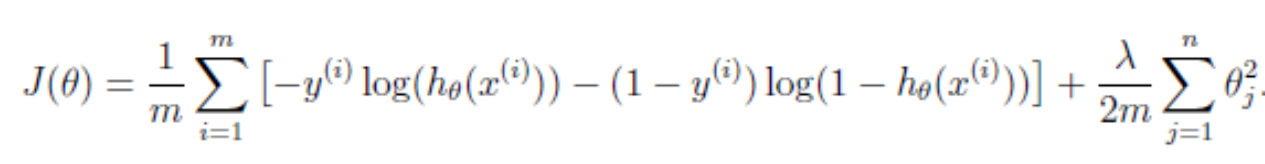
Note that for calculation of the F1 score sitting is the positive (minority) class. Be careful using confusion.m.

## Exercise 2: Classification: Logistic regression

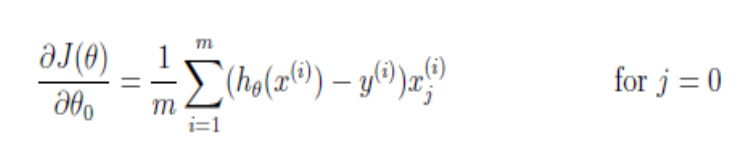
## 2.1 Cost function and gradient

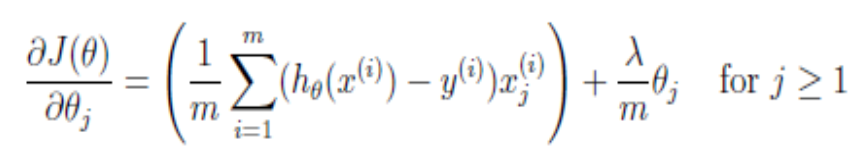
1. Accomplish the sigmoid function.(see sigmoid.m)

2. Finish the Cost function and gradient function. Recall the cost function in logistic regression:



Recall the gradient of the cost function:





This is executed in costFunctionReg.m.

Use MATLAB functions from Machine Learning Coursera course programming exercise 2. For example costFunctionReg.m, sigmoid.m, plotDecisionBoundary.m, predict.m, mapFeatures.m.

## 2.2 Linear model with 2 features

After selecting 2 features, train and test a classifier with these 2 features and calculate the F1-score.

Before classification, each feature needs to be **normalized**. Use the **training** dataset to train the model and the **validation** dataset to test the model. Use λ = 0. You must reuse the code developed in programming exercise 2 of the Coursera Machine Learning course. Show a figure with the training dataset and the linear decision boundary. You can use the instruction plotDecisionBondary (from the Coursera course) for this. Figure 2 illustrates a possible result.

**Learn Parameters-** θ

For logistic regression, you want to optimize the cost function J with parameters θ. Function *fminunc()* is used to find the best parameters θ, given a fixed dataset.

The inputs of *fminunc()* are as follows:

* The initial values of the parameters trying to optimize - initial θ.
* A function that, when given the training set and a particular θ, computes the logistic regression cost and gradient with respect to θ for the dataset (x, y).

**MATLAB functions recommend**: *fminunc(), optimset(),* ***confusionmat().*** *Also see ex2\_reg.m in Coursera Machine Learning programming assignment.*

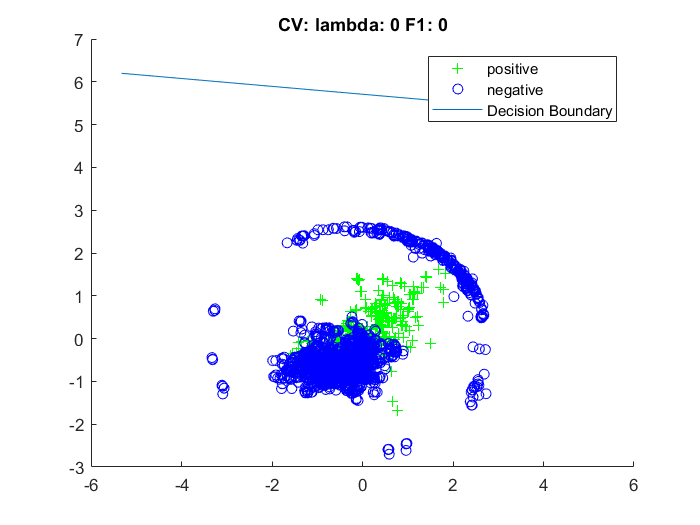


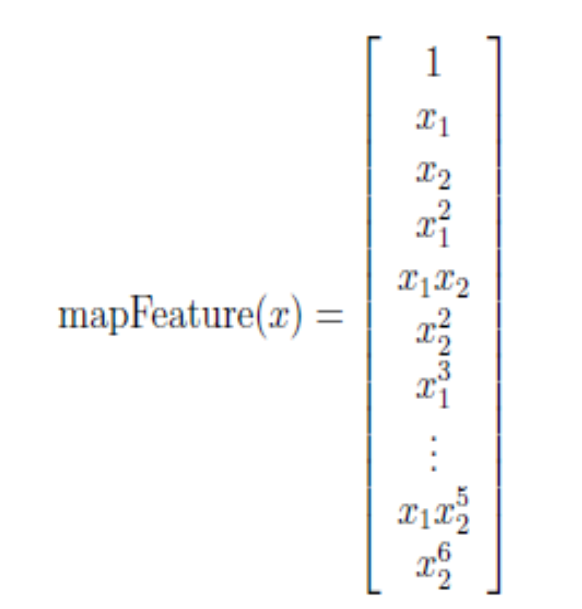
Figure 2: A linear decision boundary for two features. The ‘+’ indicates the positive interested event while the ‘o’ indicates the other events.

Calculate the f1-score obtained on the training and validation set. Discuss your results.

## 2.3 Polynomial features from 2 features

We want to improve the F1-score of the classifier. First, we will look at using a polynomial features to increase the number of features.

For example, map the features into all polynomial terms of x1 and x2 up to the 6th power. As a result of this mapping, our vector of two features has been transformed into a 28-dimensional vector. A logistic regression classifier trained on this higher-dimensional feature vector will have a more complex decision boundary and will appear nonlinear when plotted in the 2-dimensional plot. Construct the polynomial features with mapFeatures.



**Optimizing λ**

Evaluate the impact of the λ value on the performance. Vary λ in the range of [3^(-10) : 3^(10)] and train the model with this λ value. Test the model using the **validation** dataset. Plot the F1-score against the λ for the training and validation dataset. What can you conclude regarding variance and bias? A possible result can be found in Figure 3. The nonlinear decision boundary is illustrated in Figure 4.

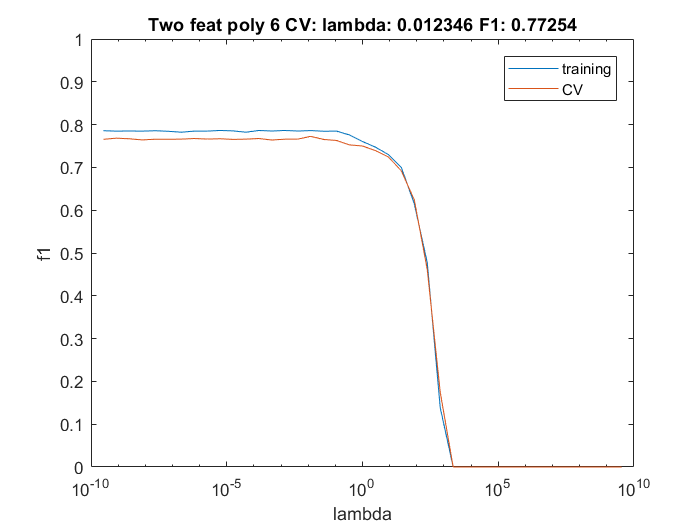


Figure 3: Illustration of the F1-score as a function of the regularization parameter lambda.

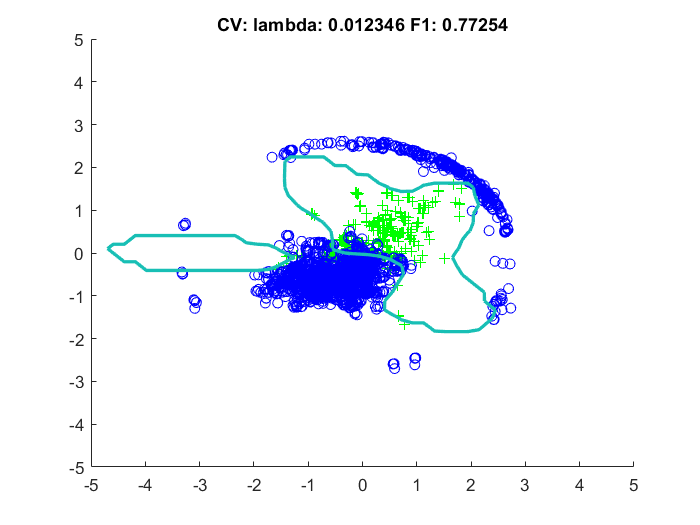


Figure 4: A non-linear decision boundary with an F1-score on the training set of 0.60.

**2.4 Construct a linear classifier with all eight features**

Now use all 8 features for classification. Again, plot the F1-score as a function of lambda as illustrated in Figure 5. We cannot visualize the decision boundary anymore as the dimensions are larger than 3. Do you notice any improvement in F1-score?

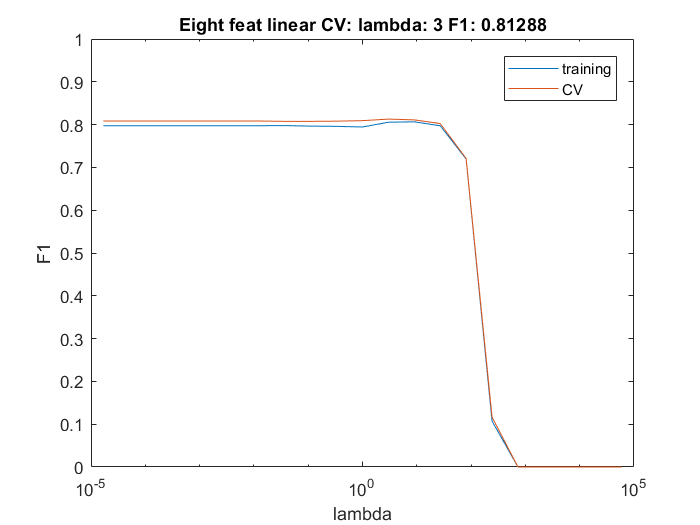


Figure 5: the F1-score as a function of the lambda value.

## 2.4 Construct a non-linear classifier with the 8 features

Finish the mapFeature function to transform the 8 features in a quadratic polynomial feature set of 45 dimensions. Again, plot the F1-score as a function of the regularization parameter lambda as illustrated in Figure 6. What can you observe regarding variance and bias?

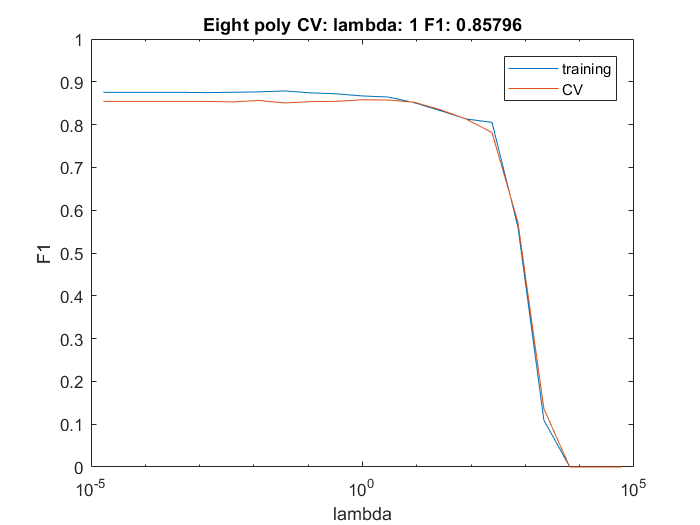


Figure 6: The F1-score as a function of the regularization parameter lambda.

Furthermore, we can add training examples and inspect the F1-score as a function of the number of training examples. This is illustrated in Figure 7. We can add more training example by concatenating examples from the test dataset.

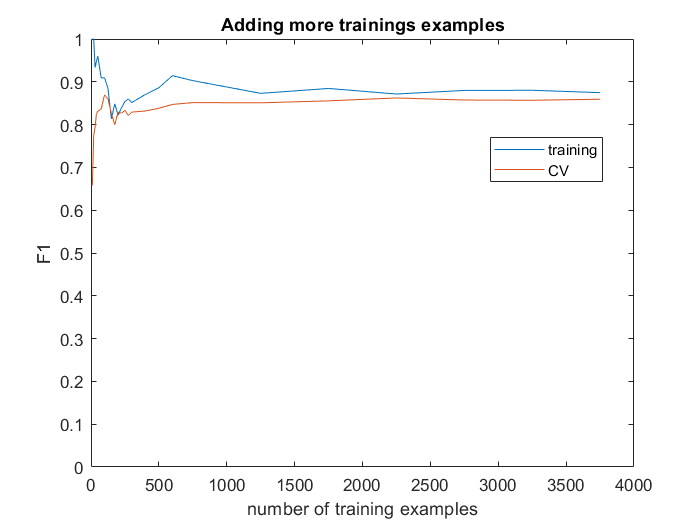


Figure 7 illustrates the F1-score as a function of the number of training examples.

## 3. What to submit

One report for 2 students. Each group choose one activity for classification. Include all m-files. Pay attention to reflect on the obtained results. Also, discuss issues for future improvement and also plot the result.

## Reference

1. Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones. 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.