Cognitive Algorithms Assignment 4

Electromyographic Decoding of Hand-Position

Due on Tuesday, June 13, 2017 ,10 am via ISIS

In this assignment you will implement a linear regression and predict two dimensional hand positions from electromyographic (EMG) recordings obtained with high-density electrode arrays on the lower arm. Download the python template `assignment4.py`, and the data set `myo_data.mat` from the ISIS web site.

1. **(12 points)** Implement ordinary least squares regression (OLS) with an optional ridge parameter by completing the function stubs `train_ols` and `apply_ols`. In `train_ols`, you estimate a linear mapping W,

$$W = (X_{\text{train}} X_{\text{train}}^{\top} + \lambda I)^{-1} X_{\text{train}} Y_{\text{train}}^{\top} \tag{1}$$

that optimally predicts the training labels from the training data, $X_{\text{train}} \in \mathbb{R}^{D_X \times N_{tr}}$, $Y_{\text{train}} \in \mathbb{R}^{D_Y \times N_{tr}}$. Here, $\lambda \in \mathbb{R}$ is the (optional) Ridge regularization parameter. The function `apply_ols` than uses the weight vector to predict the (unknown) hand positions of new test data $X_{\text{test}} \in \mathbb{R}^{D_X \times N_{te}}$

$$Y_{\text{test}} = W^{\top} X_{\text{test}} \tag{2}$$

The function `test_assignment4` helps you to debug your code.

2. **(1 point)** The data set `myo_data.mat` consists of preprocessed EMG data X and 2-dimensional stimulus labels Y. Labels are x/y positions of the hand during different hand movements. The function `load_myo_data` loads the data and splits it into train and test data. Familiarize yourself with the data by answering the following questions:
How many time points N_{tr} does the train set contain? How many time points N_{te} does the test set contain? At each time point, at how many electrodes D_X was the EMG collected?

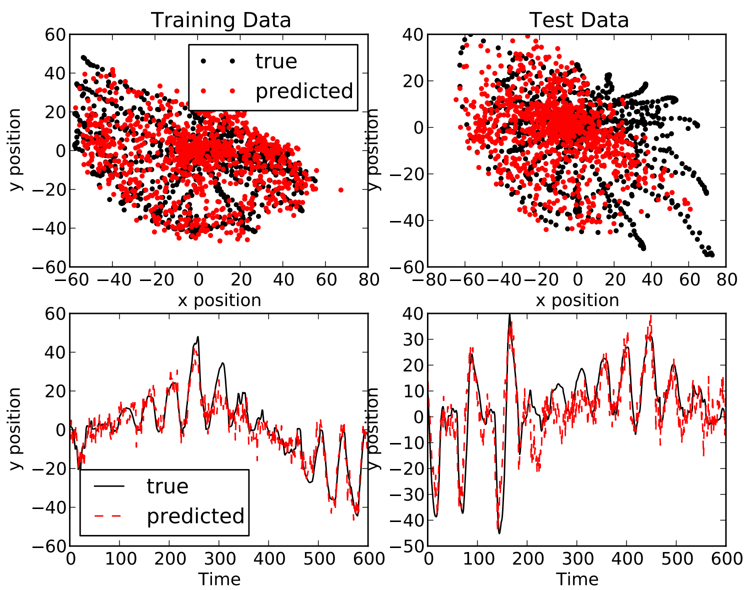

Figure 1: True versus predicted hand positions. *Upper Row*: Predicted hand position as x/y coordinates for the training and the test data set. *Lower row*: Time series of y-coordinates for the training and the test data set.

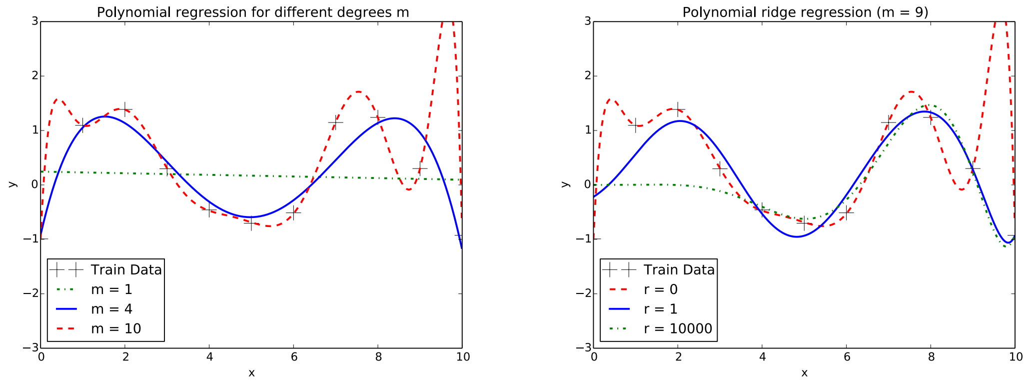

Figure 2: Polynomial ridge regression on toy data.

3. **(2 points)** Predict two dimensional hand positions by calling the function `predict_handpositions`. It plots, for the train and the test data, the true hand position versus the estimated hand position, as in Figure 1. Do you notice a performance difference between train and test data set?

4. **(3 points)** In question 3, we have used the logarithmized muscle activiations to predict the hand positions. Comment the line where we logarithmize the EMG features in the function `load_myo_data` and call `predict_handpositions` again. Do you notice a performance difference compared to the logarithmized version?

5. **(2 points)** If we cannot predict the labels Y perfectly by a linear regression on X, does this imply that the relationship between X and Y is non-linear?

6. **(10 points)** Write a function `test_polynomial_regression` which generates toy data and visualizes the results from a polynomial regression. The goal is to create two plots as in Figure 2 (Note that your figure will look slightly different, because the data is generated randomly.)

   To do so, first create toy data from a sine function as follows:

   $$x_i \in \{0, 1, 2, \ldots, 10\}, y_i = \sin(x_i) + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, 0.5)$$

   where $\mathcal{N}(\text{mean, standard deviation})$ denotes the Gaussian distribution and $i \in \{1, 2, \ldots, 11\}$ is an index. Then implement polynomial regression, which models the relationship between y and x as an mth order polynomial, i.e.

   这里要注意的是要做一个这样的矩阵phy_x:
   [1,   1,   1, ...]
   [x1,   x2,   x3   ...]
   [x1^2, x2^2, x3^3, ...]
   然后用它来带入train_ols(phy_x,y)得到w,
   在求y_pre = sp.sum(w*phy_x,axis=0)

   $$\hat{y} = w_0 + w_1 x + w_2 x^2 + \ldots + w_m x^m.$$

   The parameters $w_0, w_1, \ldots, w_m \in \mathbb{R}$ are estimated by Ridge Regression. You can use your functions `train_ols` and `apply_ols`.

   Apply and visualize polynomial ridge regression for different parameters:

   a) $m = 1, 4, 9$ with $\lambda = 0$

   b) $\lambda = 0, 1, 10000$ with $m = 9$

Please hand in your completed `assignment4.py` via ISIS. Please write your name and your Matrikel Number as the first line of the code. Also hand in a pdf file that contains your name, the answers to the questions and the generated plots, as well as your code of the functions `train_ols`, `apply_ols` and `test_polynomial_regression`.