Cognitive Algorithms Assignment 5

Improved Electromyographic Decoding of Hand-Position

Due on Wednesday, January 8th, 2014 ,10 am via ISIS

The application in this assignment is the same as in assignment 4. You will predict two dimensional hand positions $y \in \mathbb{R}^2$ from electromyographic (EMG) recordings $x \in \mathbb{R}^{192}$ obtained with high-density electrode arrays on the lower arm. Labels are 2D positions of the hand during different hand movements.
Remember that even after 'linearizing' the EMG-hand position relationship by computing the log of the EMG features, the relationship is not exactly linear. Also we do not know the exact non-linearity; it might not be the same for all regions in EMG space and for all electrodes. So we can hope to gain something from using a non-parametric and non-linear method like kernel ridge regression.
The criterion to evaluate the model and select optimal parameters is the so called coefficient of determination, or r^2 index

$$r^2 = 1 - \frac{\sum_{d=1}^{D} \mathrm{Var}(\hat{y}_d - y_d)}{\sum_{d=1}^{D} \mathrm{Var}(y_d)}$$

where D is the dimensionality of the data labels, y are the true labels and $\hat{y}$ the estimated labels. This score is 1 for perfect predictions and smaller otherwise.

Download the python template `assignment5.py` from the ISIS web site, and use the data set `myo_data.mat` from the last assignment.

1. **(18 points)** Implement Kernel Ridge Regression (KRR) by completing the function stubs `krr_train` and `krr_apply`. We use the notation from assignment 4,

$$X_{\text{train}} \in \mathbb{R}^{D_X \times N_{tr}}, Y_{\text{train}} \in \mathbb{R}^{D_Y \times N_{tr}}, X_{\text{test}} \in \mathbb{R}^{D_X \times N_{te}}$$

   In `krr_train`, you estimate a linear combination of the input vectors α,

$$\alpha = (K + \lambda I)^{-1} Y_{\text{train}}^{\top}$$

   where λ is the regularization parameter and K is the $N_{tr} \times N_{tr}$ Gaussian Kernel matrix with Kernel width σ, $K_{ij} = \exp\left(-\frac{\|X_{\text{train}}^i - X_{\text{train}}^j\|^2}{\sigma^2}\right)$. You can compute K with the provided function `GaussianKernel`.

   The function `krr_apply` than uses the weights α to predict the (unknown) hand positions of new test data X_{test}

$$Y_{\text{test}} = (\mathbf{k}\alpha)^{\top}.$$

   where $\mathbf{k}$ is the $N_{\text{test}} \times N_{\text{train}}$ matrix $\mathbf{k}_{ij} = \exp\left(-\frac{\|X_{\text{test}}^i - X_{\text{train}}^j\|^2}{\sigma^2}\right)$.

   The function `test_sine_toydata` helps you to debug your code. It generates toy data that follows a sine function, $x_i \in \{0, 0.01, 0.02, \ldots, 10\}, y_i = \sin(x_i) + \epsilon, \epsilon \sim \mathcal{N}(0, 0.5)$. You should get a result similiar to Figure 1 .

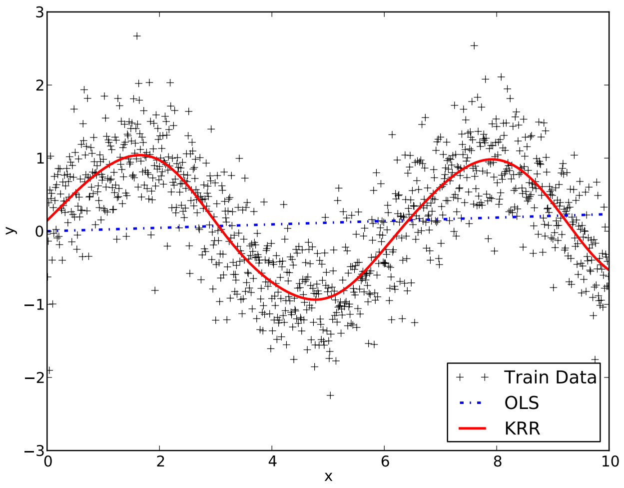

Figure 1: Ordinary Least Squares (OLS) and Kernel Ridge Regression (KRR) ($\lambda = 1, \sigma = 1$) on sine toydata

2. (**6 points**) We want to analyse how the Kernel Ridge solution depends on its hyperparameters, the kernel width σ and the regularization parameter λ.
   a) Call the function `test_sine_toydata` with $\lambda = 1$ for three different Kernel widths, $\sigma = \{0.1, 1, 10\}$. How does the Kernel width affect the solution? Explain the observed behaviour.
   b) Call the function `test_sine_toydata` with $\sigma = 1$ for three different regularization parameters, $\lambda = \{10^{-10}, 1, 500\}$. How does the regularization parameter affect the solution? Explain the observed behaviour.

3. (**2 points**) Explain briefly how λ and σ are chosen within the function `crossvalidate_krr`.

4. (**3 points**) Predict two dimensional hand positions with Kernel Ridge Regression by calling the function `test_handpositions`. It shows a boxplot for the linear regression and the Kernel Ridge Regression. What does a boxplot show? (check the help function in python or the wikipedia article). Do we gain something from Kernel Ridge Regression as compared to simple linear regression?

5. (**1 point**) For the last question, we have applied the function `test_handpositions` only to the first 1000 data points out of the 10255 available data points. Why did we do so in this exercice?

Please hand in your completed `assignment5.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 plot generated in Question 4, and at least two plots to illustrate your answers from Question 2. Please also copy your code of the functions `krr_train` and `krr_apply` in the PDF file.