Homework 2

1. Polynomial fit. You are given a dataset polyfit.json with raw data $u_i \in \mathbb{R}$ represented in an $n$-vector $u$ and raw output $v_i \in \mathbb{R}$ represented in an $n$-vector $v$.

(a) So far, we have been training and testing on the entire dataset. In class, we have introduced the notion of out-of-sample validation where we divide the dataset into training set and test set. Subsequently, we use training set to choose the predictor and test set to evaluate it.

Split the given dataset using 80-20 train-test split. Apply embeddings $\phi(u) = u$ and $\psi(v) = v$ to the raw input and raw output, respectively. Fit a least squares model to the train part of the dataset using the function find_theta used in Homework 1. Perform out of sample validation for the introduced data split and report root-mean-square error (RMSE) for the training set and test set.

(b) Now, we will introduce polynomial embedding which maps the raw input to the polynomial features of the degree $d-1$ as shown below.

$$\phi(u) = (1, u, u^2, ..., u^{d-1})$$

Write a Julia function $X=$polynomial_features$(u,d)$ which performs a polynomial embedding of degree $d-1$ described above for $n$-vector $u$ with raw input data and returns a $n \times d$ matrix $X$, where $x_i^T$ row represents the polynomial mapping for a scalar raw data input $u_i$. Embedding for raw output remains the same as in part (a).

(c) Find the optimal parameters $\theta^*$ that minimizes the square loss for the polynomial fits up to degree 20 for the entire dataset. At each iteration, report the RMSE for the dataset and compare it with the model provided in polyfit.jl file.

Plot the RMSE for the dataset versus the degree. In the same figure, plot the RMSE between your prediction and the model versus the degree. Select the degree which enables the best fit for a given dataset. Compare and plot the fittings for one example of underfitting, one example of overfitting, the best fit and the model. Comment the results.

(d) In class, we have introduced the cross validation. Implement a Julia function $\text{avg_RMSE, std = cross_validation}(X, y, k)$ for the $n \times d$ matrix $X$ of features, the $n$-vector $y$ of corresponding labels, and the $k$ number of folds. The output should contain the average of root-mean-square errors for the test set and its standard deviation for the $k$ fits.

Plot the average RMSE versus the degree for the test set with standard deviation error bars for $k = 5$. Select the best degree choice for a given dataset.

Disclaimer: Explore the Plots.jl documentation to plot the error bars.

2. Introduction to feature engineering. In this example, we will consider various feature mappings which will be used in our already created least-squares regression fitting.
(a) You are given 3 different plots representing some unknown to us relationships between raw output $v \in \mathbb{R}$ and raw input $u \in \mathbb{R}$. Our task is to identify the decision boundary, line dividing the area between blue and red points. Introduce mapping for our raw input $x = \phi(u)$ and/or raw output $y = \psi(v)$ which you would use in least-squares fitting to represent the decision boundary as accurately as possible.

![Figure 2a](image1.png)

![Figure 2b](image2.png)

![Figure 2c](image3.png)

(b) You are given categorical data and asked to choose the embedding which will faithfully represent the reality. Depending on the scenario, we might prefer one-hot embedding over ordinal embedding or the opposite. Below you are given three examples of categorical data. Explain which embedding would you pick for each example.

i. Type of transportation: bus, train, plane, car, metro
ii. Service rating: very poor, poor, average, good, excellent
iii. Gender: male, female

(c) In `features.json`, you will find an $n \times d$ matrix $U$ of data, with rows $(u^i)^T$, with $u^i \in \mathbb{R}^2$, and $n$-vector $v$ with labels. Consider the embeddings for raw input data $x = \phi(u)$:
i. $\phi_1(u) = (1, u_1, u_2)$

ii. $\phi_2(u) = (1, u_1, u_2, u_1 u_2)$

iii. $\phi_3(u) = (1, u_1, u_2, u_1 u_2, \log(u_1), \log(u_2))$

For raw output data $y = \phi(v)$, apply $\psi_2(v) = \log(v)$.

Split the data 80-20 train-test split. Fit a least squares model to the train part of the dataset. Iterate over given embeddings for raw data input. Report the RMSE for the test set for all configurations and pick the best one.

(d) In part (c), you have selected the best embedding for your raw data. Even though, this configuration resulted in the best predictor, you might be interested in the predicted raw output data, not necessarily the predicted label after the applied embedding. Hence, we need a function which will un-embed our prediction to the predicted raw output data $\hat{v}$.

Create a Julia function $\text{v\_hat} = \text{un\_embed}(\text{y\_hat})$ which un-embeds the data. Apply it to the prediction from the best model picked from part (c). Report the RMSE of the test set for the raw output values and our un-embedded prediction.