CS461 Homework 2

Mannan Shukla

Due: October 20, 2024

Problem 1: Linear Models and MMSE Regression

1.1 Data Matrix

Write out the data matrix Φ based on the given data points.

$$\Phi = \begin{bmatrix} 1 & 4 & 1 & 1 \\ 1 & 7 & 0 & 2 \\ 1 & 10 & 1 & 3 \\ 1 & 13 & 0 & 4 \end{bmatrix}$$

1.2 Exact or MMSE Solution

We can figure out whether or not not the normal equation will yield an exact solution based on Φ. If $\Phi^T\Phi$ is invertible, the equation can be solved without the need for any estimation. To find invertibility, we need to see if Φ has any linearally dependent columns. We can see that there aren't any linearly dependent columns in Φ above, so the normal equation will have an exact solution.

1.3 Solving the Normal Equation

We figured out in 1.2 that Φ is invertible because there are no linearlly dependent columns present. To solve for **w** we calculate values for the equation $\Phi^T\Phi\vec{w} = \Phi^T y$ where y is the output matrix. We can rearrange this to be $\vec{w} = (\Phi^T\Phi)^{-1}\Phi^T y$ We can solve this using numpy. The code `1-3.py` contains the code to solve for **w**. $\mathbf{w} = [1, 0, 1, 0]$

1.4 Comparing the Models

Compare the original model $y = 1 + 2x_1 + 3x_2 + 4x_3$ with your solution and discuss the differences.

1.5 Adding New Data Points

Add the new data points and solve for **w**. Discuss the chance of obtaining the original model.

1.6 Column Removal for Unique Solution

Examine the data matrix and identify which column to remove to ensure a unique solution.

Problem 2: Lagrangian Function and KKT Conditions

2.1 MMSE Objective Function

Define the MMSE objective function $J(\mathbf{w})$ and solve for the optimal solution (w_0, w_1).

2.2 Lagrangian Function

Define the Lagrangian function for the constrained optimization problem.

2.3 Solving for λ and $\mathbf{w}^*$

Based on KKT conditions, compute the optimal Lagrangian parameter λ and $\mathbf{w}^*$ for $C = \{0.5, 1, 2, 3\}$.

Problem 3: Learning Sinusoidal Functions

3.1 Implementing Ordinary MMSE Regression

Describe your code implementation in 'ols_regression.py' for the MMSE regression model. Report the average validation error across five cross-validations.

3.2 Ridge Regression

Describe your code implementation in 'ridge_regression.py' for the ridge regression model. Plot the averaged validation error for different values of λ.

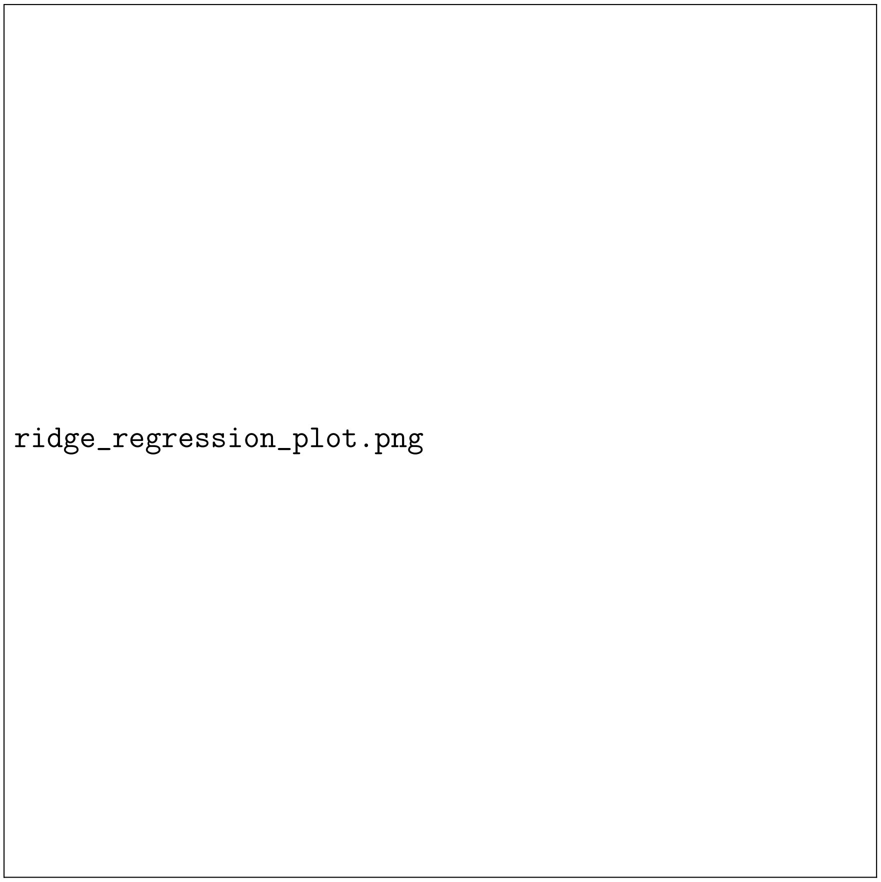

Figure 1: Validation Error vs. λ

3.3 Plot the Models

Plot the models $w(\lambda = 0)$ and $w(\lambda^*)$ over the range $0 \leq x \leq 1$.

3.4 Evaluate with Test Data

Evaluate both models on the test dataset and report the test MSE for each.

3.5 Larger Dataset

Explain the regression model you implemented in ‘ols_regression_largeset.py‘ and plot the model over the range $0 \leq x \leq 1$.

3.6 Controlling Effective Complexity

Propose two solutions to control the effective complexity in machine learning.

Problem 4: Eigenface and Spectral Decomposition

4.1 Covariance Matrix and Spectral Decomposition

Compute the covariance matrix $\mathrm{COV}(X, X)$ and its eigenvalue decomposition.

4.2 Approximating the Test Image

Approximate the test image using different M values (2, 10, 100, 1,000, 4,000). Present the corresponding images in your report.

`eigenface_M2.png` `eigenface_M10.png` `eigenface_M100.png`

Figure 2: Approximations with different M values

4.3 Eigenvectors for the Largest Eigenvalues

Show the grayscale images of the eigenvectors corresponding to the ten largest eigenvalues and explain how they capture facial features.

Problem 5: Extra Points - Year of Made Prediction

5.1 Dimensional Reduction and Whitening

Reproduce the scatter plots for $M = 1$ and $M = 2$. Discuss why the 2-D projection is more promising for year prediction.

5.2 Training the Model

Explain your implementation in ‘year_train.py‘ and describe how you selected the polynomial basis and validated the model.

5.3 Testing the Model

Report the test MSE and identify the most and least accurate predictions.