Machine Learning Principles
HW2

October 19, 2025

Tejas Kandri

---

1. Suppose someone asked you to identify a linear model used to generate the noise-free data below ($y = w_0 1 + w_1 x_1 + w_2 x_2 + w_3 x_3$). You plan to use MMSE regression to estimate the original model.
   (a) Please write a data-matrix $\phi(4 \times 4)$.
   (b) When the normal equation is shown as below, will it give you an exact solution or MMSE approximated solution?
   (c) In the normal equation $\phi^t\phi$ is invertible? Please solve the equation and find $\vec{w}$ by using pseudoinverse (numpy.linalg.pinv).
   (d) Try to solve the same normal equation in problem 1.3 by using the spectral decomposition (numpy.linalg.svd) instead of pseudo-inverse. In your computation, assume that the inverse of the smallest eigenvalue ($\approx$ zero) is an arbitrary large number. Is the solution the same as in problem 1.3? The pseudo-inverse operation contrasts with spectral decomposition, as it sets the inverse of the smallest eigenvalue to zero. Explain why pseudo-inverse and spectral decomposition yield the same solution.
   (e) The person showed you the original model as $y = 3 + 0x_1 + 3x_2 + 7x_3$. Is it the same as yours? If not, why this happened?
   (f) The person suggested that by adding four additional data points, you could obtain the same result as the original model. Please add four data points below into your normal equation and compute the new $\vec{w}$. Does your answer differ from the previous one? If so, Why? What is the probability of obtaining the original model using the regression method?

Ans:

(a) The data matrix is a matrix where the first column is all 1's and every row is a data point. The data matrix (ϕ) is:

$$\begin{bmatrix} 1 & 3 & 1 & 1 \\ 1 & 5 & 0 & 2 \\ 1 & 7 & 1 & 3 \\ 1 & 9 & 0 & 4 \end{bmatrix}$$

(b) The provided normal equation will give you an exact solution because we have 4 data points and 4 parameters to estimate. If $\phi^t\phi$ is invertible, the system will be evaluated, so we can find the exact values for all parameters. However, since $\phi^t\phi$ turns out to be singular, we get an exact solution through pseudo-inverse anyway because the data is consistent

(c) The determinant of $\phi^t\phi$ ended up being 0, so we can conclude that $\phi^t\phi$ is not invertible. The code is attached that contains the calculation for the determinant to determine invertibility and therefore also calculates the pseudo-inverse to find $\vec{w}$. The $\vec{w}$ is $[0.16666667, 2.83333333, 3, 1.33333333]$.

(d) Code submitted. In my SVD computation, I set a threshold of 1e-10 to identify near-zero singular values. The smallest singular value (1.77e-16) is treated as zero, and its inverse is set to 0 rather than a large number. This gives w = [0.167, 2.833, 3.0, 1.333], matching question 1.3. Both methods yield the same solution because pseudo-inverse uses SVD and both set inverses of near-zero eigenvalues to 0, returning the minimum-norm solution for singular matrices.

(e) No, they are not the same. The original is [3, 0, 3, 7] but I computed [0.167, 2.833, 3.0, 1.333]. This happens because $det(\phi^t\phi) = 0$, making the system rank-deficient with infinitely many solutions.

(f) Code submitted. The new $\vec{w}$ I got was $\vec{w} = [0.09845833, 2.85779167, 2.68275, 1.37966667]$ and as you can see, the new w vector is not the same as the original one I solved for. They are different because due to the determinant of $\phi^t\phi$ being 0 even with 8 data points, so there can be infinitely many solutions as these are just 2 of them, so the probability of obtaining the original model using regression method is approx. 0%. Additionally, it's 0 because with noisy real-world data, obtaining the same original model is not possible because we can't predict original noise.

2. Suppose you have two data points as below to estimate $y = w_0x_1 + w_1x_2$.

   (a) Please set an MMSE objective function $J(\vec{w})$ with the data. What is the optimal solution (w_0, w_1) that minimizes the function?

   (b) We have only two data points, so the constrained optimization problem for regularization is formulated below. Please define the Lagrangian function for the problem.

   (c) Based on KKT conditions, compute the optimal Lagrangian parameter and $\vec{w}^*$ for the different $C = \{0.5, 1, 2, 3\}$. Hint: geometrical contours will help to find the relation between w_0^* and w_1^*.

Ans:

2.1 Setup: The model is $y = w_0 x_1 + w_1 x_2$ and the data is $d_1 : (1, 0) \to 1$, $d_2 : (0, 1) \to 1$. MMSE objective function:

$$J(\vec{w}) = \frac{1}{2} \sum_{i=1}^{2} \big(y_i - (w_0 x_{1i} + w_1 x_{2i})\big)^2$$

Substitute:

$$J(w_0, w_1) = \frac{1}{2}\Big[(1 - w_0)^2 + (1 - w_1)^2\Big]$$

Optimal solution:
Take derivatives and set to zero:

$$\frac{\partial J}{\partial w_0} = -(1 - w_0) = 0 \quad \Rightarrow \quad w_0 = 1$$

$$\frac{\partial J}{\partial w_1} = -(1 - w_1) = 0 \quad \Rightarrow \quad w_1 = 1$$

Answer: $(w_0, w_1) = (1, 1)$

Problem 2.2 Constrained optimization:

$$\vec{w}^* = \arg\min_{\vec{w}} J(\vec{w}) \quad \text{subject to } ||\vec{w}||^2 \le C$$

Lagrangian function:

$$\mathcal{L}(\vec{w}, \lambda) = J(\vec{w}) + \lambda\big(||\vec{w}||^2 - C\big)$$

Explicitly:

$$\mathcal{L}(w_0, w_1, \lambda) = \frac{1}{2}\big[(1 - w_0)^2 + (1 - w_1)^2\big] + \lambda(w_0^2 + w_1^2 - C), \quad \lambda \ge 0$$

Problem 2.3 KKT conditions:

Stationary: $\nabla J(\vec{w}^*) + \lambda^* \nabla g(\vec{w}^*) = 0, \quad g(\vec{w}) = ||\vec{w}||^2 - C$
Primal feasibility: $||\vec{w}^*||^2 \le C$
Dual feasibility: $\lambda^* \ge 0$
Complementary slackness: $\lambda^*(||\vec{w}^*||^2 - C) = 0$

Stationary equations:

$$\frac{\partial \mathcal{L}}{\partial w_0} = -(1-w_0) + 2\lambda w_0 = 0 \quad \Rightarrow \quad w_0(1+2\lambda) = 1$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = -(1-w_1) + 2\lambda w_1 = 0 \quad \Rightarrow \quad w_1(1+2\lambda) = 1$$

Due to symmetry, $w_0 = w_1 = \frac{1}{1+2\lambda}$

Cases for C:

- $C \geq 2$: constraint is inactive, so $\lambda = 0$, $w_0 = w_1 = 1$
- $C < 2$: constraint is active, so $w_0^2 + w_1^2 = C$, $w_0 = w_1 = \sqrt{C/2}$,

$$\lambda = \frac{1-\sqrt{C/2}}{2\sqrt{C/2}} = \frac{\sqrt{2}-\sqrt{C}}{2\sqrt{C}}$$

Table:

C	$w_0^* = w_1^*$	λ^*
0.5	0.5	0.5
1	0.707	0.207
2	1	0
3	1	0

Overall relationship is: $w_1^* = w_0^*$ for all C due to symmetry.

3. You will recover a sinusoidal function $f(x)$ from noisy data by using MMSE regression.
   (a) Write the code "ols_regression.py" to implement an MMSE regression model (five cross-validation and ordinary MMSE) and report one validation error averaging the five cross-validations.
   (b) Write the code "ridge_regression.py" to implement a ridge regression model (five cross-validation and ridge MMSE) and plot the averaged validation error for different ridge parameters: $0 \leq \lambda \leq 1$. Based on the plot, select the λ^* and save the corresponding models $w(\lambda^*)$. Additionally, save the five models for the ordinary MMSE $w(\lambda = 0)$.
   (c) Plot the two models: $w(\lambda = 0)$ and $w(\lambda^*)$ over the range $0 \leq x \leq 1$. The five models can be averaged: $\bar{w} = \text{np.mean}(w, \text{axis} = 0)$.
   (d) Evaluate the two models with "test.npz" and report the two test MSEs.
   (e) Write the code "ols_regression_largeset.py" to implement a regression model with "train_1000.npz" without any regularization and cross validation. Plot the model over the range $0 \leq x \leq 1$.
   (f) Based on your answers in 3.3, 3.4, and 3.5, please provide two possible solutions to control the effective complexity in machine learning.

Ans:

Code submitted separately for Problems 3.1–3.5.

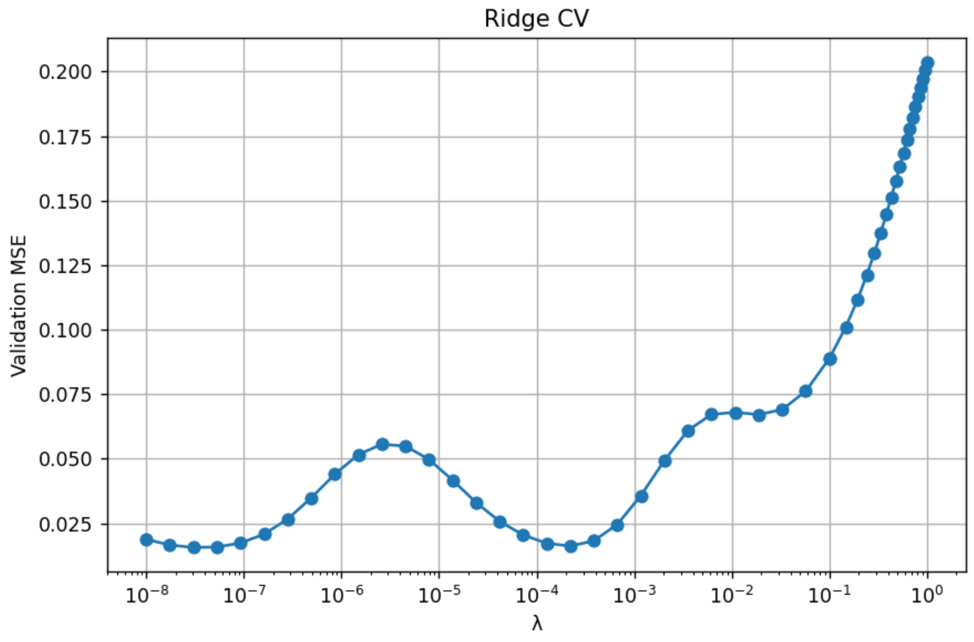
3.1

The implementation was completed in `ols_regression.py`. The results are:

$$\text{Average Validation MSE} = 1.820175$$

using 10 polynomial features (degree 9 + bias term).

3.2

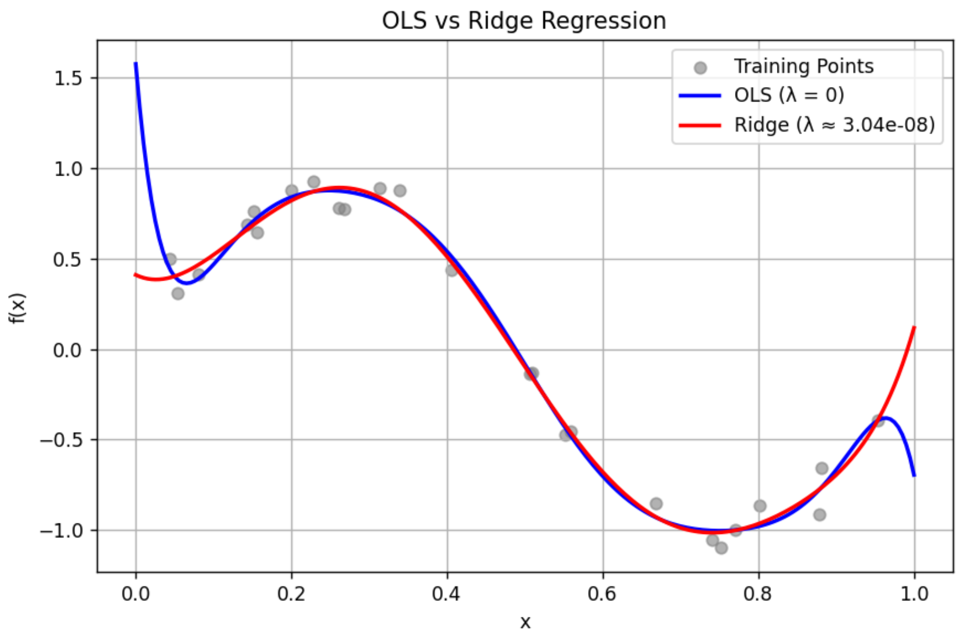


The graph shows how the MSE decreases as lambda increases, which shows the expected bias-variance tradeoff.

$$\lambda^* = 3.04 \times 10^{-8}, \quad \text{MSE}_{\text{val}} = 0.0156.$$

Both the ordinary MMSE weights $w(0)$ and the ridge MMSE weights $w(\lambda^*)$ were saved in `ridge_models.npz`.

3.3



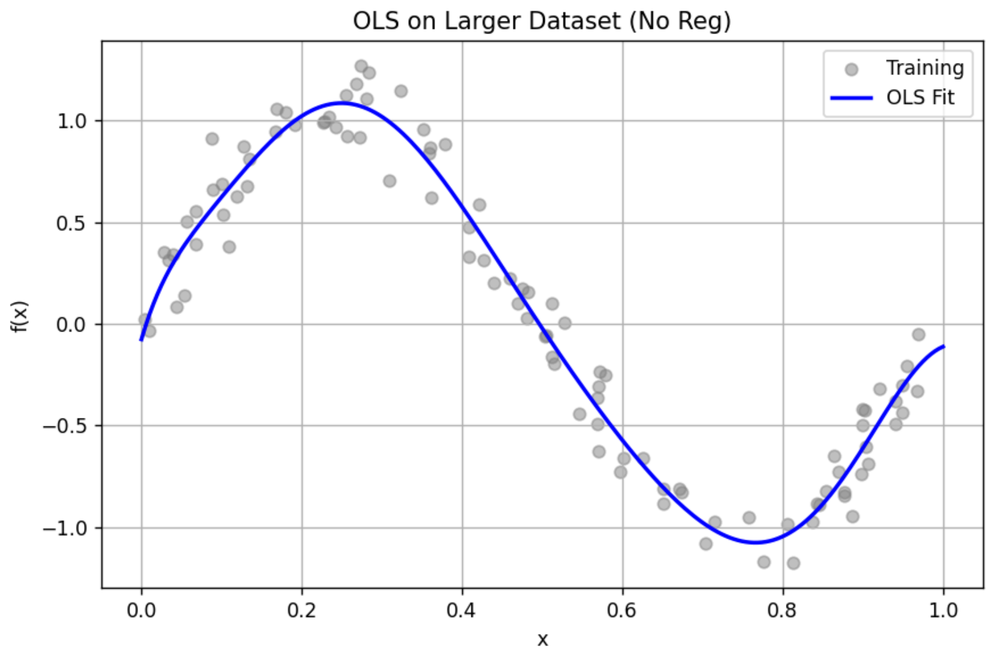
3.4

The test errors obtained were:

$$\text{MSE}_{\text{OLS}} = 1.4467, \quad \text{MSE}_{\text{Ridge}} = 0.0323, \quad \lambda^* = 3.04 \times 10^{-8}.$$

From the results, we see that the ridge regression model is lower than the OLS MSE, which indicates that introducing a small amount of L2 regularization reduces overfitting and improves generalization towards unseen data.

3.5



The fitted curve follows the underlying sinusoidal trend, showing reduced variance compared to models trained on smaller datasets.

3.6

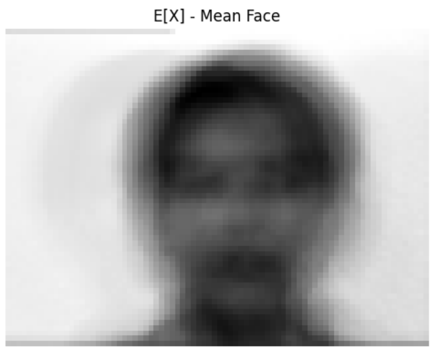
2 possible solutions are regularization and adding more training data. By adding a penalty term such as $\lambda\|w\|^2$, the model is able to discourage large weights and reduce the variance as a result. In 3.3 and 3.4, the ridge model achieved a lowest test MSE than the OLS model, which shows that regularization did control overfitting. By using a larger dataset, such as 3.5, we were able to improve generalization by allowing the model to capture the trend more accurately.

4. Spectral decomposition is applied to a large set of images to extract the most dominant correlations between images. You will approximate one test facial image by using the provided function. EM is the eigenvectors of COV (X, X) that correspond to the M largest eigenvalues where random vector X represents the whole facial data.
   (a) Please compute COV (X, X) and E[X] and spectral decomposition COV (X, X) = E∧Et by using the gif images in "train" folder. You can use numpy and PIL and no need to submit your computations.
   (b) Approximate the test image in "test" folder for different M values: 2, 10, 100, 1,000, 4,000. Present the five images in your report. Select one representative image for each M.

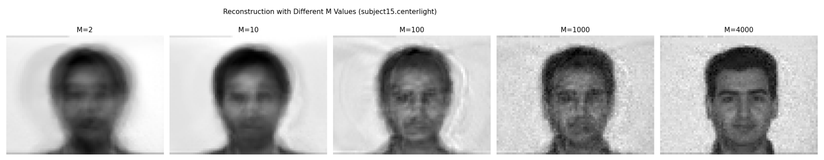
(c) Represent the eigenvectors corresponding to the 10 largest eigenvalues in grayscale images. Please include the ten images in your report and explain how the eigen-images capture the various features and aspects of human faces.

Ans:

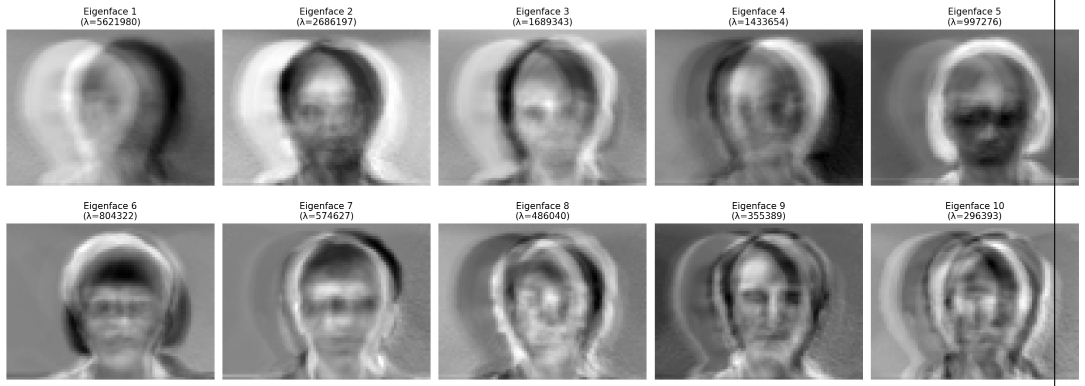
(a) code submitted.



(b) code submitted.


Reconstruction with Different M Values (subject15.centerlight)
M=2
M=10
M=100
M=1000
M=4000

(c) code submitted.



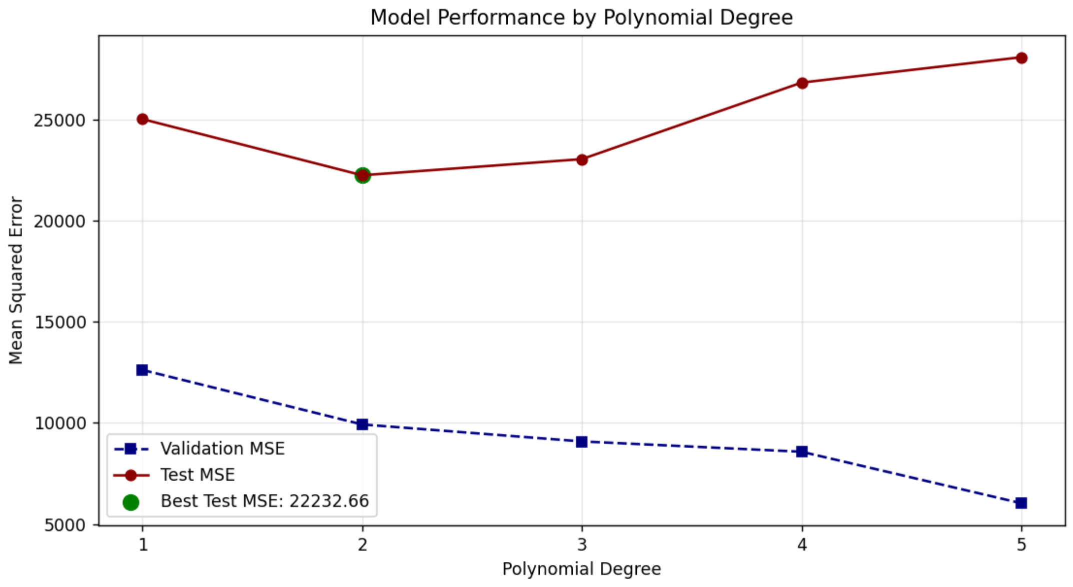
The eigen faces represent orthogonal directions of the max variance in the facial image dimensions. The first picture captures the dominant variation which is overall an illumination and face-background difference, which shows the largest proportion of variance. The next 2 images encode left-right asymmetry and central facial structure, so it contains information such as head shape and facial width. The next 3 capture regional variations which include hair patterns, upper face structure, and lower face characteristics. The final 2 images encode subtle individual differences which are essential for distinguishing faces. The rapid decline is eigenvalues shows why the reconstruction quality improves dramatically.

5. You will build a predictor of the year of made for art by using the embedding collected from a deep-CNN style classifier (VGG-16). It is a high-dimensional embedding as 512-D. After conducting dimensional reduction to 2-D and whitening, a polynomial feature map will be constructed for regression. Data samples and required specifications are given below.

   - Train and test data files: `vgg16.train.npz`, `vgg16.test.npz`, and `readme.txt`.
   - Use any order polynomial basis functions you want; for example: $1, x_1, x_2, x_1^2, x_1x_2, x_2^2$.
   - Use MSE (Mean Square Error) for the performance metric.

Ans: 5.1 Code submitted. In the 1D PCA projection, a single component does not capture enough variation to clearly separate images by year, as many images from different time periods are mapped to similar values along PC1. In contrast, the 2D projection includes an additional component that captures more structure in the data, resulting in better separation between older and newer images.

5.2 Code submitted.



5.3 Code submitted.

Test MSE at Best Degree:	22,232.66
Validation MSE at Best Degree:	9,918.22
Most Accurate Prediction Image:	*12121_through-the-hills-in-sw-texas-1911.jpg*
Least Accurate Prediction Image:	*1151_christ-before-herod-1509.jpg*