Homework 1
CS534 Machine Learning, Spring 2019

This homework explores the concepts of math with random variables, covariance, and linear regression. Points are noted in each problem. There are no time limits.

Problem 1 - Expectations (10 points)

Given a model of a random process

$$Y = f(X) + \epsilon \tag{1}$$

where Y is what is measured, X are variables, and $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ is gaussian noise.

1.a. Suppose you have a prediction model $\hat{f}$, show that the expected error $Err(x) = E[(Y - \hat{f}(x))^2 | X = x]$ can be written as

$$Err(x) = \sigma_\epsilon^2 + (E[\hat{f}(x)] - f(x))^2 + E[(\hat{f}(x) - E[\hat{f}(x)])^2] \tag{2}$$

Let $f = f(x)$, $\hat{f} = \hat{f}(x)$

$$\begin{aligned}
E[(Y - \hat{f})^2] &= E[(Y - f + f - \hat{f})^2] \\
&= E[(Y - f)^2] + E[(f - \hat{f})^2] + 2E[(Y - f)(f - \hat{f})] \\
&= E[(f + \epsilon - f)^2] + E[(f - \hat{f})^2] + 2E[(Y - f)(f - \hat{f})] \\
&= \underbrace{\sigma_\epsilon^2}_{\textit{noise variance}} + E[(f - \hat{f})^2] + 2E[(Y - f)(f - \hat{f})] \qquad (\epsilon \text{ zero-mean})
\end{aligned}$$

Look more closely the term $2E[(Y - f)(f - \hat{f})]$

$$\begin{aligned}
E[(Y - f)(f - \hat{f})] &= E[Yf - Y\hat{f} - f^2 + f\hat{f}] \\
&= E[(f + \epsilon)f - (f + \epsilon)\hat{f} - f^2 + f\hat{f}] \\
&= f^2 + fE[\epsilon] - E[f\hat{f}] - E[\epsilon\hat{f}] - f^2 + E[f\hat{f}] \qquad (f \text{ not R.V., } \epsilon \text{ zero-mean}) \\
&= f^2 - f^2 + f \cdot 0 - 0 \cdot E[\hat{f}] + E[f\hat{f}] - E[f\hat{f}] \qquad (\hat{f}, \epsilon \text{ independent}) \\
&= 0
\end{aligned}$$

So

$$E[(Y - \hat{f})^2] = \sigma_\epsilon^2 + E[(f - \hat{f})^2].$$

Now examine the term $E[(f - \hat{f})^2]$

$$\begin{aligned} E[(f - \hat{f})^2] &= E[(f - E[\hat{f}]) + (E[\hat{f}] - \hat{f}))^2] \\ &= E[(f - E[\hat{f}])^2] + \underbrace{E[(\hat{f} - E[\hat{f}])^2]}_{model\ variance} + 2E[(f - E[\hat{f}])(E[\hat{f}] - \hat{f})] \end{aligned}$$

The first term here $E[(f - E[\hat{f}])^2]$ looks close to $(E[\hat{f}] - f)^2$, see if we can make it work

$$\begin{aligned} E[(f - E[\hat{f}])^2] &= E[(f - E[\hat{f}])(f - E[\hat{f}])] \\ &= E[f^2 - 2fE[\hat{f}] + (E[\hat{f}])^2] \\ &= f^2 - 2fE[\hat{f}] + (E[\hat{f}])^2 \qquad (f, E[\hat{f}] \text{ not random}) \\ &= \underbrace{(f - E[\hat{f}])^2}_{model\ bias} \end{aligned}$$

Now analyze the second term

$$\begin{aligned} E[(f - E[\hat{f}])(E[\hat{f}] - \hat{f})] &= E[fE[\hat{f}] - f\hat{f} - (E[\hat{f}])^2 + \hat{f}E[\hat{f}]] \\ &= fE[\hat{f}] - fE[\hat{f}] - (E[\hat{f}])^2 + (E[\hat{f}])^2 \qquad (f, E[\hat{f}] \text{ not random}) \\ &= 0 \end{aligned}$$

Putting it all together

$$E[(Y - \hat{f})^2] = \sigma_\epsilon^2 + (f - E[\hat{f}])^2 + E[(\hat{f} - E[\hat{f}])^2]$$

1.b. Describe what each of these terms represents in plain English.

σ_ϵ^2 - represents the variance or power of the measurement noise. The choice of model cannot reduce this error term.

$(f - E[\hat{f}])^2$ - is the bias between the true and estimated models. The choice of model can reduce this error term.

$E[(\hat{f} - E[\hat{f}])^2$ - is the variance of the estimated model. This represents variability in model fitting and sensitivity to training data.

Problem 2 - Covariance (10 points)

2.a. Suppose you want to generate samples from a normally-distributed random variable $X \sim \mathcal{N}(\mu_X, \Sigma_X), X \in \mathbb{R}^p$. Show with math that you can transform samples from the standard normal distribution $\mathcal{N}(0, I)$ (where I is the identity) to match this distribution using the diagonalization $\Sigma_X = V\Lambda V^T$.

The desired covariance, Σ_X, can be written as $\Sigma_X = V\Lambda V^T$. Let $Y \sim \mathcal{N}(0, I)$, and define the transformed variable $Z = V\Lambda^{0.5}Y + \mu_X$

First, determine the mean of Z

$$\begin{aligned} E[Z] &= E[V\Lambda^{0.5}Y + \mu_X] \\ &= E[V\Lambda^{0.5}Y] + \mu_X \\ &= V\Lambda^{0.5}E[Y] + \mu_X \\ &= \mu_X. \end{aligned}$$

Now compute the covariance of Z

$$\begin{aligned} E[(Z - E[Z])(Z - E[Z])^T] &= E[(V\Lambda^{0.5}Y)(Y^T(\Lambda^{0.5})^T V^T)] \\ &= V\Lambda^{0.5}E[YY^T]\Lambda^{0.5}V^T \qquad (V, \Lambda \text{ not random}) \\ &= V\Lambda^{0.5}I\Lambda^{0.5}V^T \qquad (\text{substituting covariance of } Y,\ \Lambda \text{ is diagonal}) \\ &= V\Lambda V^T \\ &= \Sigma_X. \end{aligned}$$

So Z, X share the same covariance and mean.

One way to interpret the diagonalization is that V is a projection/rotation onto the principal components of Σ_X, Λ contains the variance of the principal components on the diagonal, and V is a change of basis ($V^T = V^{-}1$). So the transformation $Z = V\Lambda^{0.5}Y$ stretches the (independent) variables in Y in the space of principal components, and the change of basis V introduces the necessary dependence by combining these variables.

2.b. Use this procedure to generate $N = 1000$ samples from the distribution

$$X = [x_1, x_2]^T \sim \mathcal{N}(\mu, \Sigma), \quad \mu = [1, 1]^T, \quad \Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 0.5 \end{bmatrix}. \tag{3}$$

Display the samples x using a scatter plot. Superimpose the eigenvectors and the level curves of the PDF on the scatter plot.

See plot below.

2.c. In a new figure superimpose the level curves of the Euclidean distance on the scatter plot

$$D(x) = \|x - \mu\|_2 \tag{4}$$

See plot below.

2.d. In a new figure superimpose the level curves of the *Mahalanobis distance* on the scatter plot

$$M(x) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{5}$$

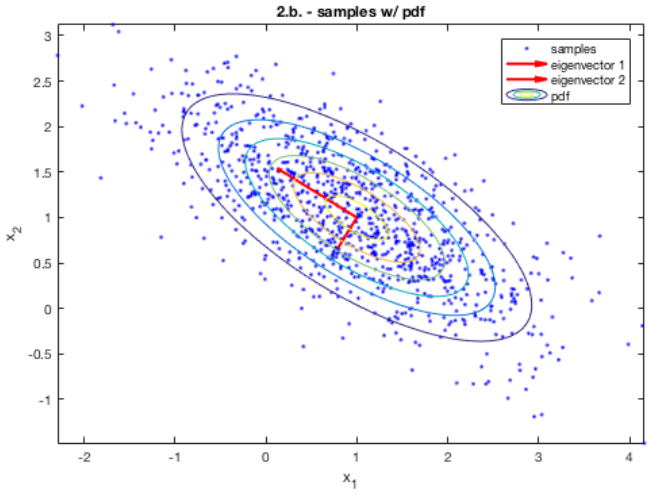

Figure 1: PDF and eigenvectors (problem 2b.)

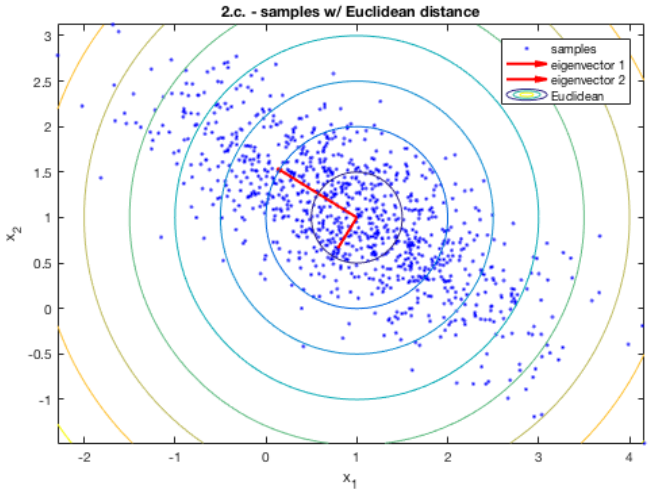

Figure 2: Euclidean distance (problem 2c.)

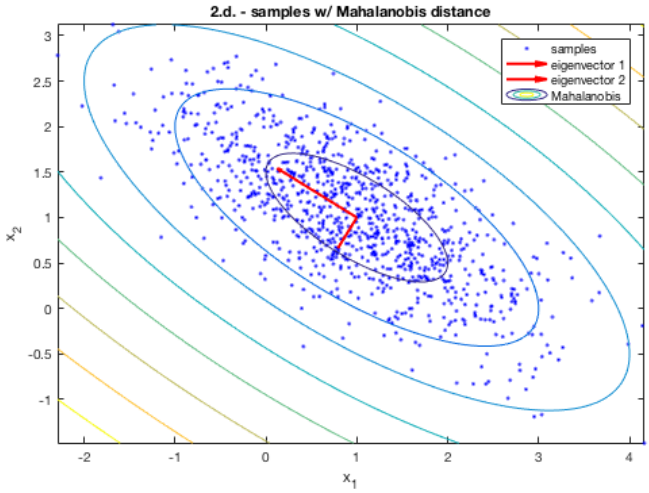

Figure 3: Mahalanobis distance (problem 2d.)

Problem 3 - Regularizing linear regression (15 points)

In this problem we have a dataset that contains $p = 100$ features, but the underlying model that relates X, Y involves only 5 of these

$$Y = \beta_{j_1} X_{j_1} + \beta_{j_2} X_{j_2} + \ldots + \beta_{j_5} X_{j_5} + \epsilon \tag{6}$$

for some $j_1, j_2, \ldots, j_3 \in \{1, \ldots, 100\}$.

LASSO regression incorporates a model penalty in the loss function that effectively encourages a *sparse* solution, forcing many model weights β_j towards zero

$$Loss(\beta) = \sum_{i=1}^{N} (y_i - \beta x_i)^2 + \lambda |\beta|_1. \tag{7}$$

Since the L_1 norm is not differentiable, LASSO models can be derived using the sub-gradient method for gradient-descent. This sub-gradient method represents the gradient of the penalty term using a *soft thresholding* operation

$$S(\lambda, \beta_j) = \begin{cases} \beta_j - \lambda & \text{if } \beta_j > \lambda \\ 0 & \text{if } -\lambda \leq \beta_j \leq \lambda \\ \beta_j + \lambda & \text{if } \beta_j < -\lambda. \end{cases} \tag{8}$$

This sub-gradient term is combined with the gradient of the cost term $\sum_{i=1}^{N} (y_i - \beta x_i)^2$ for gradient descent.

Your professor made an error here and so we won't grade this problem harshly. The soft threshold defined above should be applied to the entire gradient update term

$$\beta^{t+1} = S(\beta^t - \frac{2\gamma}{N} X^t (X\beta^t - y), \gamma\lambda)$$

You can read more about the soft-thresholding approach for LASSO implementation on slide 20 here.

3.a. Find a LASSO model $\hat{\beta}$ to the training data using gradient descent. Use the penalty weight $\lambda = 1$ and learning rate $\gamma = 5e-3$ to train the model

for 10000 gradient updates. Plot the final model coefficients. Can you guess the indices of the nonzero model weights $\{j_1, \ldots, j_5\}$?

See figure next page.

3.b. Fit a Least Squares model without regularization to the training set. Plot the final model coefficients.

See figure next page.

3.c. Compare the mean-square error of the LASSO and ordinary least squares models on the testing set.

See figure titles next page.

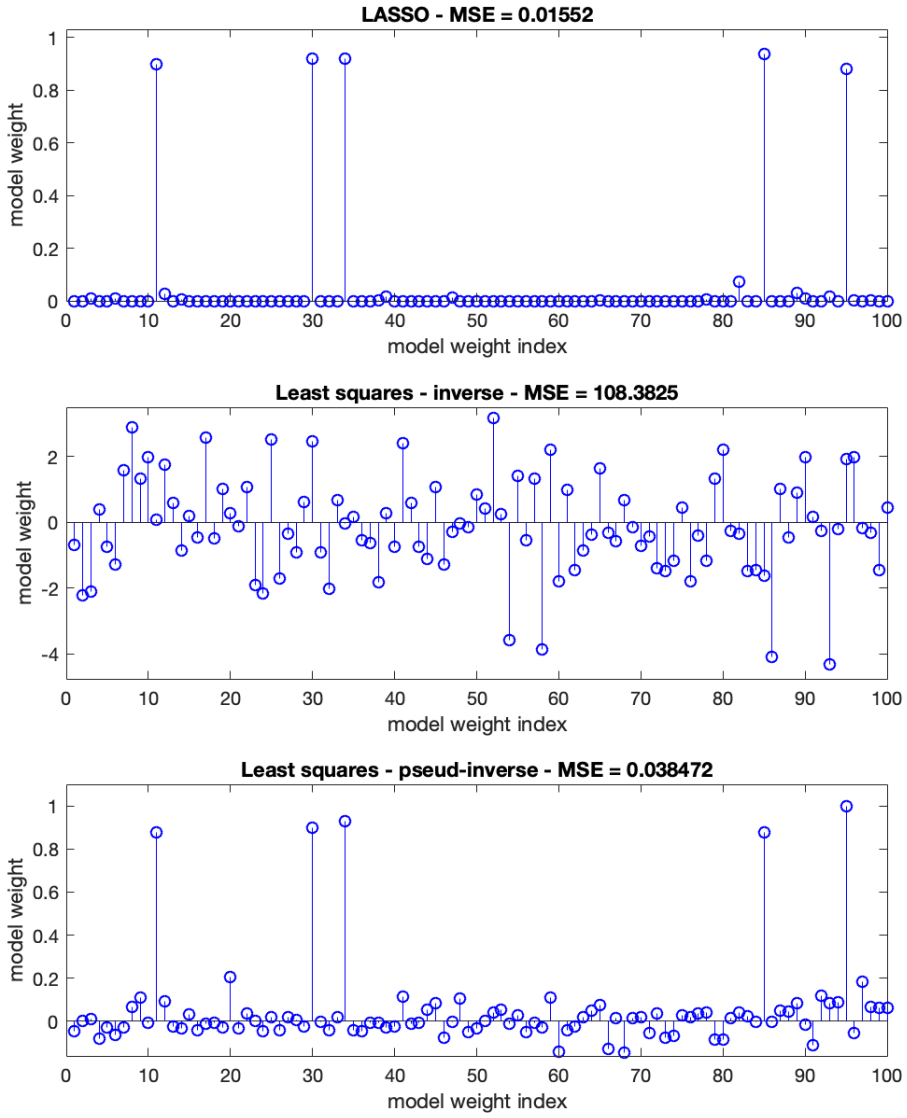

Figure 4: LASSO, least squares (inverse), and least squares (pseudo-inverse) solutions. The selected coefficients are $\{j_1, \ldots, j_5\} = \{11, 30, 34, 85, 95\}$.

Problem 4 - Robust fitting with outliers (15 points)

The *RANdom SAmple Consensus* (RANSAC) algorithm has been used for over 38 years to fit models in the presence of large number of outliers. In this problem you will be using data generated from the process

$$Y = f(X) = -3.2591X^3 + 4.8439X^2 + 1.7046X + 1.0685 + \epsilon \qquad (9)$$

where $\epsilon \sim \mathcal{N}(0, 1e-2)$. These samples contain outliers generated from a uniform distribution.

4.a. Use polynomial least squares to estimate $\hat{f}$. Display the samples in a scatter plot and superimpose the true and estimated model on this plot.

See plot below.

4.b. Implement RANSAC to estimate $\hat{f}$. Choose 5 points at random to fit each model, and use the threshold $|y - \hat{f}(x)| \leq 0.3$ to define the consensus set. Stop when the number of inliers exceeds %40 of the total samples, and recalculate the final model using this consensus set. Display the samples in a scatter plot and indicate the final consensus set. Superimpose f, $\hat{f}$ on this plot.

See plot below.

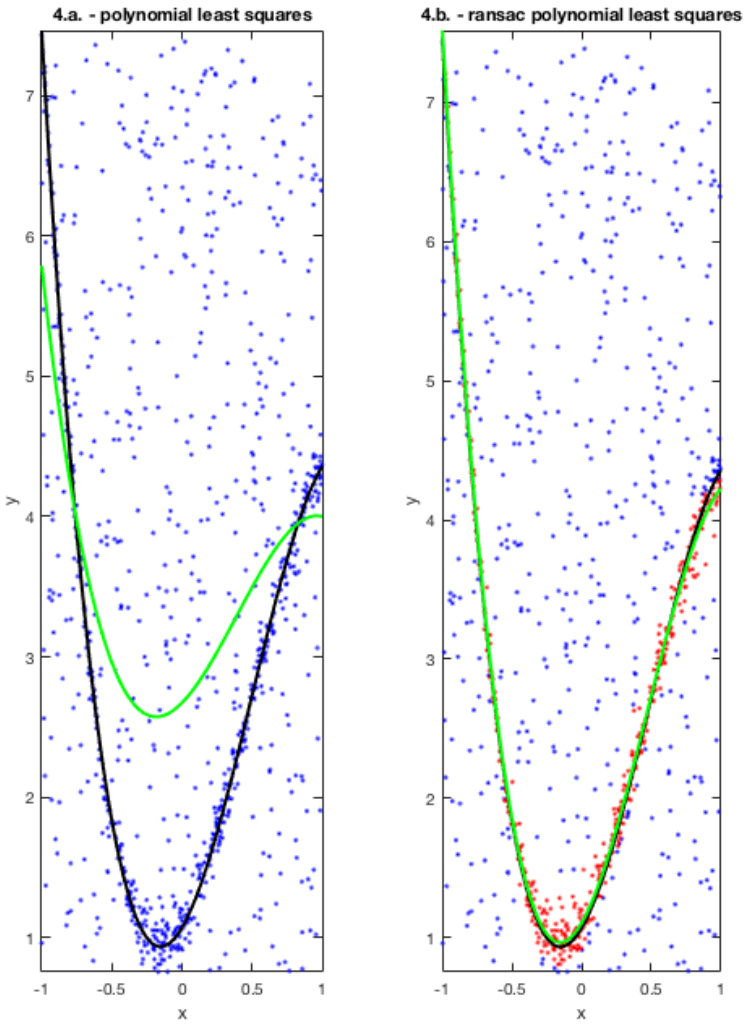

Figure 5: Polynomial least-squares (left) and RANSAC (right) models. Fitted models shown in green. Underlying model f shown in black. RANSAC consensus set points shown in red.