
# Assignment 1

Mahdi Tabatabaei 400101515
github repository

**Deep Learning**

Dr. Fatemizadeh

October 19, 2024

# Deep Learning

Assignment 1

Mahdi Tabatabaei 400101515
github repository

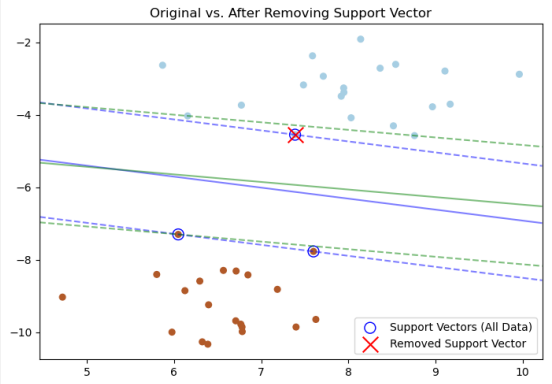
## Question 1 (60 Points)

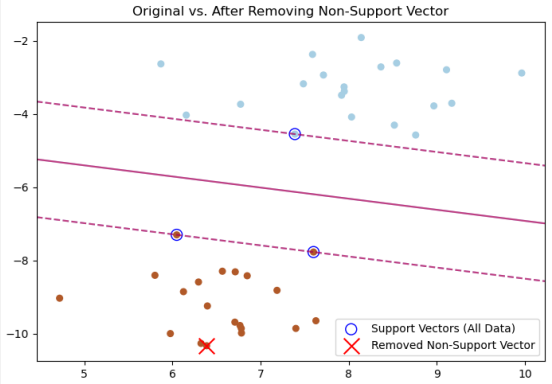
(a) In the case of linear separability, if one of the training samples is removed, will the decision boundary change or remain the same? Does the boundary move away from or toward the removed sample? Justify your answer.

The answer depends on where the removed training sample locates.

- **The removed point is close to the decision boundary:** If the removed point was near the decision boundary and played a role in determining its position (for example, as a support vector in algorithms like Support Vector Machines - SVM), the decision boundary is likely to shift towards or away from the removed point.
- **The removed point is far from the decision boundary:** If the removed point was far from the decision boundary and did not contribute significantly to its determination, the removal of the point will likely not affect the decision boundary.

There is an example of SVM below:



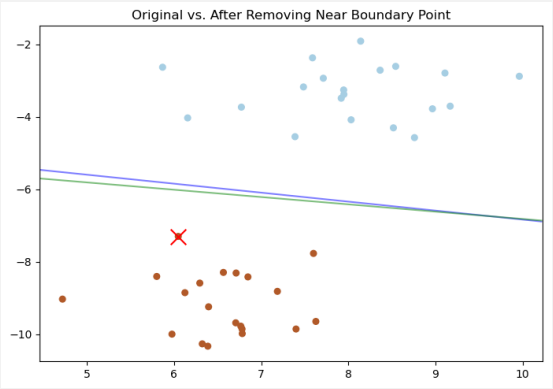

- **If the removed sample is a support vector:** In this case, removing the training sample may shift the decision boundary toward or away. Because supprot vectors have critical role in location of decision boundary.
- **If the removed sample is not a support vector:** In this case, removing that sample may cause a little to no effect in location of boundary.

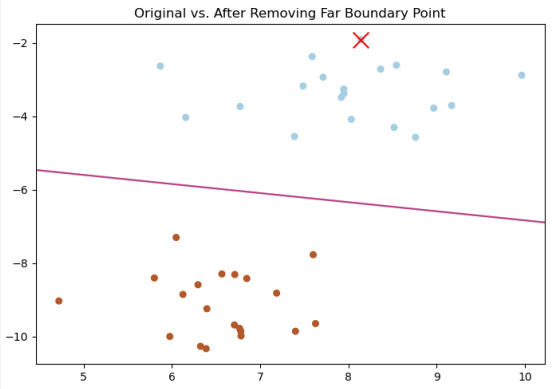
In summary, the result depends on the distribution and importance of the other points in the training set.

Now, if we consider the decision boundary from a logistic regression perspective, will the decision boundary change or remain the same? Justify your answer. (There is no need to specify the direction of change.)

Decision boundary is based on probabalistic models logistic regression. As we know, in training of logistic regression and its loss function's formula we use all the samples. But the amount of effect depends on distance of data from the boundary.

- **The removed sample is close to the decision boundary:** If the removed sample is near the decision boundary, removing it could have a more significant effect on the boundary's position, as these points contribute more to determining the boundary. In this case, the boundary might shift slightly towards the remaining data points.
- **The removed sample is far from the decision boundary:** If the removed sample is far from the decision boundary and has a smaller influence on it, the change in the boundary might be minimal or negligible. However, since logistic regression considers the entire dataset, even distant points may have some influence, albeit small.




(b) i. From the lecture notes, remember that if we allow some training data to be misclassified, the soft-margin SVM is formulated as follows:

$$\min_{\omega,\xi_i} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \quad y_i\left(\omega^T x_i\right) \geq 1-\xi_i, \quad \forall i \in \{1,\ldots,n\}$$
$$\xi_i \geq 0, \quad \forall i \in \{1,\ldots,n\}$$

Where $\xi_1, \xi_2, \ldots, \xi_n$ are slack variables. Suppose $\xi_1, \xi_2, \ldots, \xi_n$ have been computed. Use $\xi_i$ to determine an upper bound for the number of misclassified points.

If we solve the convex optimization problem, we have:

$$\alpha_i\left[y_i(\langle w, x_i\rangle + w_0) - 1 + \xi_i\right] = 0, \quad i = 1,\ldots,N$$
$$\mu_i\xi_i = 0, \quad i = 1,\ldots,N$$

- $\alpha_i > 0 \Rightarrow x_i$ is a Support Vector $\Rightarrow y_i(\langle w, x_i\rangle + w_0) = 1 - \xi_i$
  - $\mu_i > 0 \Rightarrow \begin{cases} \xi_i = 0 \Rightarrow x_i \text{ is on the margin} \\ C - \alpha_i - \mu_i = 0 \Rightarrow 0 < \alpha_i < C \end{cases}$
  - $\xi_i > 0 \Rightarrow \begin{cases} x_i \text{ crosses the margin} \\ \mu_i = 0 \Rightarrow \alpha_i = C \end{cases}$
- $y_i\left(\langle w, x_i\rangle + w_0\right) > 1 - \xi_i \Rightarrow x_i$ is **NOT** Support Vector and classified correctly.

$$\begin{cases} C - \alpha_i - \mu_i = 0 \\ \alpha_i = 0 \end{cases} \Rightarrow \mu_i = C > 0 \Rightarrow \xi_i = 0 \Rightarrow y_i\left(\langle w, x_i\rangle + w_0\right) > 1$$

So, we understand $\xi_i \geq 0$ is the only case that the point crosses the margin and it may missclassify or classify correctly. So:

$$\text{number of missclassified points} \leq \sum_{i=1}^{n} \mathbb{I}(\xi_i > 0)$$

ii. What is the role of the SVM multiplier $C$? Provide your answer by considering two cases: $C \to 0$ and $C \to \infty$.

**Case 1:** $C \to 0$

$$\min_{\boldsymbol{\omega}, b} \quad \frac{1}{2}\|\boldsymbol{\omega}\|^2$$

$$\text{subject to} \quad y_i(\boldsymbol{\omega}^T \boldsymbol{x}_i + b) \geq 1, \quad i = 1, \ldots, n$$

In this case, the SVM classifier focuses on minimizing $||w||^2$ which is equivalent to maximizing $||w||^{-1}$ which means the marging between classes. This can cause underfitting of model, larger margin and higher misclassification errors.

**Case 2:** $C \to \infty$

$$\min_{\boldsymbol{\omega}, b, \boldsymbol{\xi}} \quad C \sum_{i=1}^{n} \xi_i$$

$$\text{subject to} \quad y_i(\boldsymbol{\omega}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$$

In this case, the SVM focuses on minimizing the slack variables $\xi_i$, which leads to fewer classification errors but potentially at the cost of reducing the margin. This can cause overfitting as it forces the decision boundary to fit the training data more precisely.

So, $C$ is a regularization parameter which we use to balance the trade-off between maximizing the margin and minimizing the classification errors.

iii. Compare the hard SVM and logistic regression when the two classes are linearly separable. Explain the difference in their decision boundaries.

When the two classes are linearly separable, **Hard SVM** and **Logistic Regression** both aim to find a decision boundary that separates the two classes, but they approach the task differently.

### 1. Hard SVM

**Objective:** Maximizing the margin between the two classes, Margin is defined as the distance between the decision boundary and the closest data points (support vectors).

**Decision Boundary:** Maximize the distance between the boundary and the nearest points from either class, Highly dependent on the support vectors, Points outside the margin do not affect the decision boundary.

**Behavior:** No misclassification is allowed, Sensitive to outliers, Single misclassified point could significantly change the boundary.

### 2. Logistic Regression

**Objective:** Modeling the probability of a point belonging to one of the two classes using a logistic function, Minimizing the cross-entropy loss across all data points.

**Decision Boundary:** Found where the probability of class membership is 0.5, All data points contribute to determining the boundary.

**Behavior:** More robust to outliers, The boundary is determined based on the overall probability distribution rather than just the support vectors.

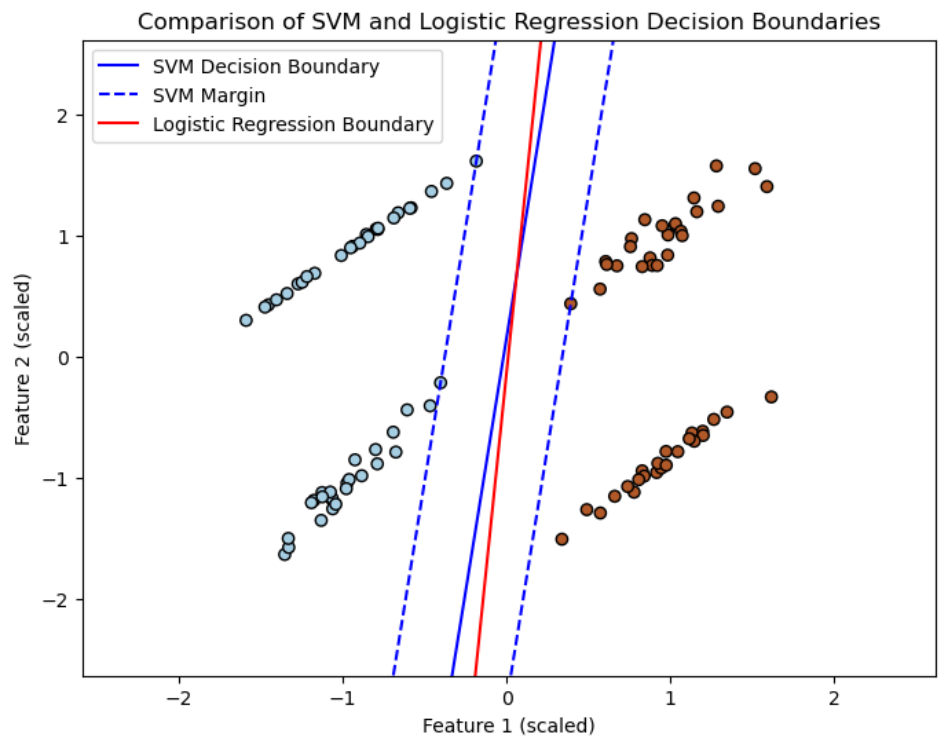



Figure 1: Hard-Margin SVM vs. Logistic Regression

iv. Compare the soft-margin SVM and logistic regression when the two classes are not linearly separable. Explain the difference in their decision boundaries.

**1. Soft-Margin SVM**

**Objective:** Maximizing the margin between the two classes while allowing some misclassification.

**Decision Boundary:** The decision boundary is determined by maximizing the margin while balancing misclassifications based on the parameter $C$. Points close to the margin (support vectors) affect the boundary.

**Behavior:** Outliers close to the margin or on the wrong side can heavily affect the boundary, especially when $C$ is large.

**2. Logistic Regression**

**Objective:** Minimizes the cross-entropy loss across all data points.

**Decision Boundary:** All data points contribute to determining the boundary, as logistic regression aims to minimize the overall classification error based on probability.

**Behavior:** More robust to outliers than SVM because it focuses on minimizing the total log loss, and outliers have less influence on the decision boundary compared to the support vectors in SVM. The decision boundary is smoother, and logistic regression tends to create a boundary that considers the overall distribution of the data points rather than just focusing on the points near the margin.

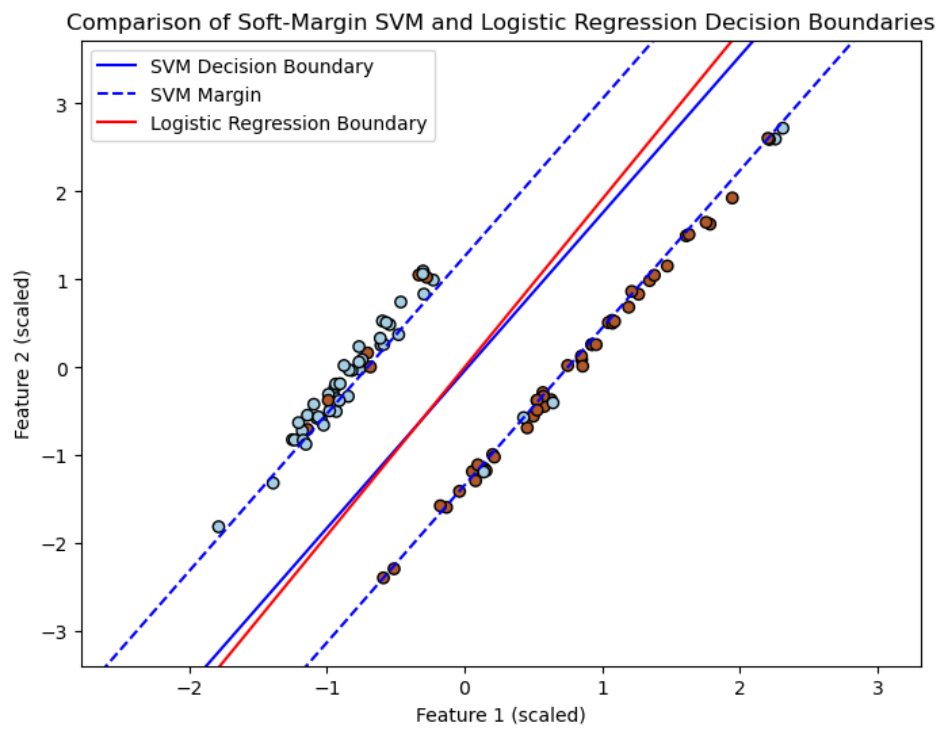



Figure 2: Soft-Margin SVM vs. Logistic Regression

# Question 2 (60 Points)

Suppose in PCA, we project each point $x_i$ onto $z_i = V_{1:k}^T x_i$, where $V_{1:k} = [v_1, \ldots, v_k]$ represents the first $k$ principal components. We can reconstruct $x_i$ from $z_i$ as:

$$\hat{x}_i = V_{1:k} z_i$$

i. Show that:

$$\|\hat{x}_i - \hat{x}_j\| = \|z_i - z_j\|$$

We know that the points $x_i$ are projected onto a lower-dimensional subspace by applying $V_{1:k}^T$, where $V_{1:k}$ contains the first $k$ principal components. The projected points are given by:

$$z_i = V_{1:k}^T x_i$$

The reconstructed points are:

$$\hat{x}_i = V_{1:k} z_i$$

Now, to prove:

$$\|\hat{x}_i - \hat{x}_j\| = \|V_{1:k} z_i - V_{1:k} z_j\|$$

Since $V_{1:k}$ is an orthonormal matrix:

$$\|V_{1:k} z_i - V_{1:k} z_j\| = \|V_{1:k}\| \|z_i - z_j\| = \|z_i - z_j\|$$

So,

$$\|\hat{x}_i - \hat{x}_j\| = \|z_i - z_j\|$$

ii. Show that the reconstruction error is:

$$\sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2 = (n-1) \sum_{j=k+1}^{p} \lambda_j$$

where $\lambda_{k+1}, \ldots, \lambda_p$ are the smallest eigenvalues. Thus, the more principal components we use for reconstruction, the more accurate the reconstruction becomes.

$$\|x_i - \hat{x}_i\|^2 = (x_i - \hat{x}_i)^T (x_i - \hat{x}_i) = x_i x_i^T$$

$$\|x_i - \hat{x}_i\|^2 = x_i^\top x_i - x_i^\top \hat{x}_i - \hat{x}_i^\top x_i + \hat{x}_i^\top \hat{x}_i$$

$$\|x_i - \hat{x}_i\|^2 = x_i^\top x_i - x_i^\top W_k W_k^\top x_i - x_i^\top W_k W_k^\top x_i + x_i^\top W_k W_k^\top W_k W_k^\top x_i$$

Simplify, noting that $W_k^\top W_k = I$ (identity matrix) because eigenvectors are orthonormal:

$$\begin{aligned} \|x_i - \hat{x}_i\|^2 &= x_i^\top x_i - 2 x_i^\top W_k W_k^\top x_i + x_i^\top W_k W_k^\top x_i \\ &= x_i^\top x_i - x_i^\top W_k W_k^\top x_i \\ &= x_i^\top (I - W_k W_k^\top) x_i \end{aligned}$$

Now, let's sum over all points:

$$\sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2 = \sum_{i=1}^{n} x_i^\top (I - W_k W_k^\top) x_i$$

We assume that mean of sample is zero in PCA:

$$\begin{aligned}\sum_{i=1}^{n} x_i^\top (I - W_k W_k^\top) x_i &= \sum_{i=1}^{n} Tr((I - W_k W_k^T) x_i x_i^T) \\ &= Tr(I - W_k W_k^T) \sum_{i=1}^{n} x_i x_i^T \\ &= (n-1) Tr(I - W_k W_k^T)(\frac{1}{n-1})(\sum_{i=1}^{n} x_i x_i^T) \\ &= (n-1) Tr((I - W_k W_k^T) S)\end{aligned}$$

So, we can write the equation as follow:

$$\sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2 = (n-1) \cdot \mathrm{Tr}((I - W_k W_k^\top) S)$$

Where $S$ is the sample covariance matrix and $n$ is the number of data points.
The covariance matrix can be decomposed as $S = W \Lambda W^\top$, where $\Lambda$ is a diagonal matrix of eigenvalues.
Substituting this in:

$$\sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2 = (n-1) \cdot \mathrm{Tr}((I - W_k W_k^\top) W \Lambda W^\top)$$

Using properties of trace and the fact that $W$ is orthogonal:

$$\begin{aligned}\sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2 &= (n-1) \cdot \mathrm{Tr}(\Lambda - W_k^\top W \Lambda W^\top W_k) \\ &= (n-1) \cdot \left( \sum_{j=1}^{p} \lambda_j - \sum_{j=1}^{k} \lambda_j \right) \\ &= (n-1) \cdot \sum_{j=k+1}^{p} \lambda_j\end{aligned}$$

# Question 3 (60 Points)

Consider the equation $Xw = y$, where $X \in \mathbb{R}^{m \times n}$ is a non-square data matrix, $w$ is a weight vector, and $y$ is a vector of labels corresponding to each data point in each row of $X$.

Assume $X = U\Sigma V^T$ (full SVD of $X$). Here, $U$ and $V$ are square and orthogonal matrices, and $\Sigma$ is an $m \times n$ matrix with non-zero singular values ($\sigma_i$) on the diagonal.

For this problem, $\Sigma^\dagger$ is defined as an $n \times m$ matrix with the inverse of singular values ($\frac{1}{\sigma_i}$) along the diagonal.

(a) First, consider the case where $m > n$, meaning the data matrix $X$ has more rows than columns (tall matrix) and the system is overdetermined. How do we find the weights $w$ that minimize the error between $Xw$ and $y$? In other words, we want to solve $\min \|Xw - y\|^2$.

$$\min \|Xw - y\|^2$$

The objective function can be written as:

$$\|Xw - y\|^2 = (Xw - y)^T(Xw - y)$$

$$(Xw - y)^T(Xw - y) = w^T X^T Xw - 2y^T Xw + y^T y$$

Next, we differentiate the quadratic expression with respect to $w$:

$$\frac{d}{dw}\left(w^T X^T Xw - 2y^T Xw\right)$$

$$2X^T Xw - 2X^T y = 0$$

Simplifying this equation:

$$X^T Xw = X^T y$$

$$w = (X^T X)^{-1} X^T y$$

(b) Use the SVD of $X = U\Sigma V^T$ and simplify.

Given that we have the SVD of $X$ as $X = U\Sigma V^T$:

$$(U\Sigma V^T)^T(U\Sigma V^T)w = (U\Sigma V^T)^T y$$

$$V\Sigma^T U^T U\Sigma V^T w = V\Sigma^T U^T y$$

Since $U$ is orthogonal, $U^T U = I$. Also, $V$ is orthogonal, so $V^T V = I$:

$$V\Sigma^T \Sigma V^T w = V\Sigma^T U^T y$$
$$V(\Sigma^T \Sigma)V^T w = V\Sigma^T U^T y$$
$$w = V(\Sigma^T \Sigma)^{-1}\Sigma^T U^T y$$

We can simplify by the definition of pseudoinverse $A^\dagger = (A^T A)^{-1} A^T$:

$$w = V\Sigma^\dagger U^T y$$

(c) You will notice that the least squares solution is of the form $w^* = Ay$. What happens if we multiply $X$ from the left by matrix $A$? For this reason, matrix $A$ is called the left inverse least squares.

From part (b), we know:

$$A = V\Sigma^\dagger U^T$$

So, we can write:

$$\begin{aligned} XA &= (U\Sigma V^T)(V\Sigma^\dagger U^T) \\ &= U\Sigma(V^T V)\Sigma^\dagger U^T \\ &= U\Sigma\Sigma^\dagger U^T \end{aligned}$$

Now, let's consider $\Sigma\Sigma^\dagger$:

- For an $m \times n$ matrix where $m > n$ (our case), $\Sigma\Sigma^\dagger$ is an $m \times m$ matrix
- It has 1's for the first $n$ diagonal elements (corresponding to non-zero singular values)
- The remaining $(m - n)$ diagonal elements are 0

Therefore, $\Sigma\Sigma^\dagger$ is equivalent to an $m \times m$ identity matrix with only the first $n$ diagonal elements as 1, and the rest 0. Let's call this matrix $I_n$.
So,

$$XA = UI_nU^T$$

This result, $UI_nU^T$, is the projection matrix onto the column space of $X$. It projects any vector onto the space spanned by the columns of $X$.
This is why $A$ is called the left inverse of $X$ in the least squares sense: when applied to $X$ from the left, it produces the projection onto the column space of $X$, which is the best possible approximation of the identity transformation given the column space of $X$.

(d) Now, consider the case where $m < n$, meaning the data matrix $X$ has more columns than rows and the system is underdetermined. There are infinite solutions for $w$, but we are looking for the minimum norm solution, i.e., we want to solve $\min \|w\|^2$ subject to $Xw = y$. What is the minimum norm solution?

$$\begin{aligned} &\min \|w\|^2 \\ \text{s.t} \quad &\|Xw - y\|^2 \end{aligned}$$

To solve this optimization problem:

$$\mathcal{L} = \|w\|^2 + \lambda^T(Xw - y)$$

$$\frac{d\mathcal{L}}{dw} = 2w + X^T\lambda = 0$$

$$X^T\lambda = -2w$$

$$\lambda = -\frac{1}{2}(XX^T)^{-1}Xw = -2(XX^T)^{-1}y$$

$$w = X^T(XX^T)^{-1}y$$

(e) Use the SVD of $X = U\Sigma V^T$ and simplify.

$$w = (U\Sigma V^T)^T((U\Sigma V^T)(U\Sigma V^T)^T)^{-1}y$$

$$w = V\Sigma^T(\Sigma\Sigma^T)^{-1}U^{-1}y$$

(f) You will notice that the minimum norm solution is of the form $w^* = By$. What happens if we multiply $X$ from the right by matrix $B$? For this reason, matrix $B$ is called the right inverse minimum norm.

from part (e), we know:

$$B = V\Sigma^T(\Sigma\Sigma^T)^{-1}U^{-1}$$

So, we can write:

$$\begin{aligned} BX &= (V\Sigma^T(\Sigma\Sigma^T)^{-1}U^{-1})(U\Sigma V^T) \\ &= V\Sigma^T(\Sigma\Sigma^T)^{-1}\Sigma V^T \\ &= V\Sigma^T(\Sigma^T)^{\dagger}V^T \end{aligned}$$

Now, let's consider $\Sigma^T(\Sigma^T)^{\dagger}$:

- For an $m \times n$ matrix where $n > m$ (our case), $\Sigma^T(\Sigma^T)^{\dagger}$ is an $n \times n$ matrix
- It has 1's for the first $m$ diagonal elements (corresponding to non-zero singular values)
- The remaining $(n - m)$ diagonal elements are 0

Therefore, $\Sigma^T(\Sigma^T)^{\dagger}$ is equivalent to an $n \times n$ identity matrix with only the first $m$ diagonal elements as 1, and the rest 0. Let's call this matrix $I_m$.
So,

$$BX = VI_mV^T$$

This result, $VI_mV^T$, is the projection matrix onto the row space of $X$. It projects any vector onto the space spanned by the rowa of $X$.
This is why $B$ is called the right inverse of $X$ in the least squares sense: when applied to $X$ from the right, it produces the projection onto the row space of $X$, which is the best possible approximation of the identity transformation given the row space of $X$.

# Question 4 (60 Points)

Consider a linear regression problem that includes $n$ data points and $d$ features. When $n = d$, the matrix $F \in \mathbb{R}^{n \times n}$ has an eigenvalue $\alpha$ with a very small value. Let's ignore this small value and noise. We have $y = Fw + \epsilon$. If we calculate $\hat{w}_{inv} = F^{-1}y$, we can observe a small value $\epsilon$ and noise $F$ such that $\|\hat{w}_{inv} - w^*\| = 10^{-11}$. Let's ignore the reason behind this small value.

Instead of inverting $F$, assume we use gradient descent. We repeat gradient descent $k$ times starting from $w = 0$ with a loss function $\ell(w) = \frac{1}{2}\|y - Fw\|^2$. We assume that the learning rate $\eta$ is small enough to ensure the stability of gradient descent for the given problem (this is an important point).

The gradient descent update formula for $t > 0$ is as follows:

$$w_t = w_{t-1} - \eta \left(F^T \left(Fw_{t-1} - y\right)\right)$$

We are looking for the error $\|w_k - w^*\|_2$. We want to show that, in the worst case, this error can be bounded by the following:

$$\|w_k - w^*\|_2 \leq k\eta\alpha\|y - \hat{w}\|_2$$

In other words, the error cannot go out of bounds, at least not too quickly.

To complete this task, we only need to prove the key idea using the triangle inequality and the norm properties, as the result will follow naturally.

Show that for $t > 0$:

$$\|w_t\|_2 \leq \|w_{t-1}\|_2 + \eta\alpha\|y\|_2$$

We start with the update rule for gradient descent:

$$w_t = w_{t-1} - \eta \left(F^T \left(Fw_{t-1} - y\right)\right)$$

Expanding this expression:

$$w_t = w_{t-1} - \eta F^T F w_{t-1} - \eta F^T y$$

$$= \left(I - \eta F^T F\right) w_{t-1} - \eta F^T y$$

Next, we want to find the second norm of $w_t$:

$$\|w_t\|_2 = \| \left(I - \eta F^T F\right) w_{t-1} - \eta F^T y\|_2$$

Using the triangle inequality, we can bound the norm:

$$\|w_t\|_2 \leq \|I - \eta F^T F\|_2 \|w_{t-1}\|_2 + \|\eta F^T y\|_2$$

Since the spectral norm of $I - \eta F^T F$ is related to its largest eigenvalue, we substitute the bound using $\alpha$, the largest eigenvalue of $F^T F$:

$$= \sqrt{1 - \eta\alpha}\|w_{t-1}\|_2 + \eta\|F^T y\|_2$$

$$\leq \|w_{t-1}\|_2 + \eta\|F^T y\|_2$$

Finally, we arrive at the inequality:

$$\|w_t\|_2 \leq \|w_{t-1}\|_2 + \eta\alpha\|y\|_2$$

This shows that the norm of the weights decreases over iterations, provided $\eta$ is chosen appropriately.

If gradient descent cannot diverge, what can be said about the eigenvalues of $I - \eta F^T F$? What shape do these eigenvalues take?

The eigenvalues of the matrix $I - \eta F^T F$ are given by:

$$\mu_i = 1 - \eta\lambda_i$$

where $\lambda_i$ is an eigenvalue of $F^T F$.

**Condition for Convergence:**

$$-1 \leq \mu_i \leq 1$$

$$-1 \leq 1 - \eta\lambda_i \leq 1$$

$$0 \leq \lambda_i \leq \tfrac{2}{\eta}$$

Solving this inequality for $\eta$, we get:

$$0 < \eta < \frac{2}{\lambda_{\max}}$$

where $\lambda_{\max}$ is the largest eigenvalue of $F^T F$.

# Question 5 (60 Points)

1. (a) Show that the expected squared error can be decomposed into three parts: bias, variance, and irreducible error $\sigma^2$:

$$Error = Bias^2 + Variance + \sigma^2$$

Formally, assume we have a randomly sampled training set $\mathcal{D}$ (which is independent of our test data), and we select an estimator $\theta = \hat{\theta}(\mathcal{D})$ (for example, using empirical risk minimization). The expected squared error for a test input $x$ is decomposed as follows:

$$\mathbb{E}Y \sim p(y|x), \mathcal{D}\left[(Y - \hat{f}\hat{\theta}(\mathcal{D})(x))^2\right] = Bias\left(\hat{f}\hat{\theta}(\mathcal{D})(x)\right)^2 + Var\left(\hat{f}\hat{\theta}(\mathcal{D})(x)\right) + \sigma^2$$

The formula definitions of variance and bias given below may be useful to recall:

$$Bias(\hat{f}\hat{\theta}(\mathcal{D})(x)) = \mathbb{E}Y \sim p(Y|x), \mathcal{D}\left[\hat{f}_{\hat{\theta}(\mathcal{D})}(x) - Y\right]$$

$$Var(\hat{f}\hat{\theta}(\mathcal{D})(x)) = \mathbb{E}\mathcal{D}\left[\left(\hat{f}\hat{\theta}(\mathcal{D})(x) - \mathbb{E}\mathcal{D}[\hat{f}_{\hat{\theta}(\mathcal{D})}(x)]\right)^2\right]$$

The derivation of the bias–variance decomposition for squared error proceeds as follows. For convenience, we drop the $D$ subscript in the following lines, such that $\hat{f}(x; D) = \hat{f}(x)$. Let us write the mean-squared error of our model:

$$\text{MSE} \triangleq \mathbb{E}\left[(y - \hat{f}(x))^2\right]$$

$$= \mathbb{E}\left[(f(x) + \epsilon - \hat{f}(x))^2\right]$$

$$= \mathbb{E}\left[(f(x) - \hat{f}(x))^2\right] + 2\mathbb{E}\left[(f(x) - \hat{f}(x))\epsilon\right] + \mathbb{E}[\epsilon^2]$$

We can show that the second term of this equation is null:

$$\mathbb{E}\left[(f(x) - \hat{f}(x))\epsilon\right] = \mathbb{E}[f(x) - \hat{f}(x)]\mathbb{E}[\epsilon] = 0$$

since $\epsilon$ is independent from $x$, and $\mathbb{E}[\epsilon] = 0$.
Moreover, the third term of this equation is nothing but $\sigma^2$, the variance of $\epsilon$.
Let us now expand the remaining term:

$$\mathbb{E}\left[(f(x) - \hat{f}(x))^2\right] = \mathbb{E}\left[(f(x) - \mathbb{E}[\hat{f}(x)] + \mathbb{E}[\hat{f}(x)] - \hat{f}(x))^2\right]$$

$$= \mathbb{E}\left[(f(x) - \mathbb{E}[\hat{f}(x)])^2\right] + 2\mathbb{E}\left[(f(x) - \mathbb{E}[\hat{f}(x)])(\mathbb{E}[\hat{f}(x)] - \hat{f}(x))\right] + \mathbb{E}\left[(\mathbb{E}[\hat{f}(x)] - \hat{f}(x))^2\right]$$

We show that:

$$\mathbb{E}\left[(f(x) - \mathbb{E}[\hat{f}(x)])(\mathbb{E}[\hat{f}(x)] - \hat{f}(x))\right] = \mathbb{E}\left[f(x)\mathbb{E}[\hat{f}(x)] - f(x)\hat{f}(x) - \mathbb{E}[\hat{f}(x)]\hat{f}(x) + \hat{f}(x)^2\right]$$

$$= f(x)\mathbb{E}[\hat{f}(x)] - f(x)\mathbb{E}[\hat{f}(x)] - \mathbb{E}[\hat{f}(x)]^2 + \mathbb{E}[\hat{f}(x)]^2 = 0$$

Eventually, we plug our derivations back into the original equation and identify each term:

$$\text{MSE} = \Big(f(x) - \mathbb{E}[\hat{f}(x)]\Big)^2 + \mathbb{E}\left[\Big(\mathbb{E}[\hat{f}(x)] - \hat{f}(x)\Big)^2\right] + \sigma^2$$
$$= \text{Bias}(f(x))^2 + \text{Var}(\hat{f}(x)) + \sigma^2$$

Finally, the MSE loss function (or negative log-likelihood) is obtained by taking the expectation value over $x \sim P$:

$$\text{MSE} = \mathbb{E}_x\left\{\text{Bias}_D[\hat{f}(x;D)]^2 + \text{Var}_D[\hat{f}(x;D)]\right\} + \sigma^2.$$

(b) Suppose our training set consists of $D = \{(x_i, y_i)\}_{i=1}^n$, where the only randomness comes from the labels $Y$, which are generated from the linear model $Y = X\theta^* + \epsilon$, and each noise variable $\epsilon_i$ is independently and identically distributed with zero mean and variance 1. We use the ordinary least squares (OLS) estimator $\hat{\theta}$ to estimate $\theta$ based on this data.

You are asked to estimate the error and variance of $\hat{\theta}$ in predicting the outputs for a specific test input $x$. The OLS solution is given as:

$$\hat{\theta} = (X^\top X)^{-1} X^\top Y,$$

where $Y \in \mathbb{R}^n$ consists of independent and identically distributed data points. Suppose that $X^\top X$ is diagonal for simplicity.

First, we calculate $\mathbb{E}[\hat{\theta}]$

$$\begin{aligned}
\mathbb{E}[\hat{\theta}] &= \mathbb{E}[(X^T X)^{-1} X^T Y] \\
&= \mathbb{E}[(X^T X)^{-1} X^T (X\theta^* + \epsilon)] \\
&= \mathbb{E}[\underbrace{(X^T X)^{-1} X^T X}_{I}\, \theta^*] + \mathbb{E}[(X^T X)^{-1} X^T \epsilon)] \\
&= \mathbb{E}[\theta^*] + (X^T X)^{-1} X^T \underbrace{\mathbb{E}[\epsilon)]}_{0} \\
&\rightarrow \mathbb{E}[\hat{\theta}] = \theta^*
\end{aligned}$$

Now, we have to calculate the $Cov(\hat{\theta})$

$$\begin{aligned}
Cov(\hat{\theta}) &= \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])(\hat{\theta} - \mathbb{E}[\hat{\theta}])^T] \\
&= ((X^\top X)^{-1} X^\top (X\theta^* + \epsilon))((X^\top X)^{-1} X^\top (X\theta^* + \epsilon))^T \\
&= (X^T X)^{-1} X^T (X\theta^* + \epsilon)(X\theta^* + \epsilon)^T X (X^T X)^{-T}
\end{aligned}$$

As we know $\mathbb{E}[\epsilon] = 0$:

$$\begin{aligned}
Cov(\hat{\theta}) &= \mathbb{E}[\underbrace{(X^T X)^{-1}(X^T X)}_{I}\, \theta^* (\theta^*)^T \underbrace{X^T X (X^T X)^{-T}}_{I}] + \mathbb{E}[(X^T X)^{-1} X^T \epsilon\epsilon^T X (X^T X)^{-T}] \\
&= \mathbb{E}[\theta^* (\theta^*)^T] + (X^T X)^{-1} X^T \underbrace{\mathbb{E}[\epsilon\epsilon^T]}_{I} X (X^T X)^{-T}
\end{aligned}$$

$$\rightarrow \quad Cov(\hat{\theta}) = (\theta^*)^T\theta^* + (X^TX)^{-1}$$

To find bias:

$$\begin{aligned} Bias &= \mathbb{E}[\hat{f} - Y] \\ &= \mathbb{E}[X\hat{\theta}] - \mathbb{E}[X\theta^* + \epsilon] \\ &= X\theta^* - X\theta^* - 0 = 0 \end{aligned}$$

To find variance:

$$\begin{aligned} Variance &= \mathbb{E}[(\hat{f} - \mathbb{E}[\hat{f}])(\hat{f} - \mathbb{E}[\hat{f}])^T] \\ &= \mathbb{E}[(\hat{f} - X\theta^*)(\hat{f} - X\theta^*)^T] \\ &= \mathbb{E}[\hat{f}\hat{f}^T] - X\theta^*\mathbb{E}[\hat{f}^T] - \mathbb{E}[\hat{f}](\theta^*)^TX^T + \mathbb{E}[X\theta^*(\theta^*)^TX^T] \\ &= \mathbb{E}[X\hat{\theta}\hat{\theta}^TX^T] - X\theta^*(\theta^*)^TX^T \end{aligned}$$

From matrix covarince and expected value of $\hat{\theta}$:

$$\begin{aligned} Variance &= \mathbb{E}[X(X^TX)^{-1}X^TYY^TX(X^TX)^{-T}X^T] - X\theta^*(\theta^*)^TX^T \\ &= \mathbb{E}[X(X^TX)^{-1}X^T(X\theta^* + \epsilon)((\theta^*)^TX^T + \epsilon^T)X(X^TX)^{-T}X^T] - X\theta^*(\theta^*)^TX^T \\ &= \mathbb{E}[X \underbrace{(X^TX)^{-1}X^TX}_{I}\theta^*(\theta^*)^T \underbrace{X^TX(X^TX)^{-T}}_{I}X^T] \\ &+ \mathbb{E}[X(X^TX)^{-1}X^T\epsilon\epsilon^TX(X^TX)^{-T}X^T] - X\theta^*(\theta^*)^TX^T \\ &= X\theta^*(\theta^*)^TX^T + X(X^TX)^{-1}X^T\underbrace{\mathbb{E}[\epsilon\epsilon^T]}_{I}X(X^TX)^{-T}X^T - X\theta^*(\theta^*)^TX^T \\ &= X(X^TX)^{-1}\underbrace{X^TX(X^TX)^{-T}}_{I}X^T = X(X^TX)^{-1}X^T \end{aligned}$$

So,

$$MSE = X(X^TX)^{-1}X^T + \Sigma$$