Assignment 1

Ehsan Merrikhi 400101967
github repository

Deep Learning

Dr. Fatemizadeh

October 25, 2024

Deep Learning

Assignment 1

Ehsan Merrikhi 400101967
github repository

Contents

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.

- **For a linear classifier:** If one of the training samples is removed, the decision boundary may shift depending on how influential the removed point is to the boundary. If the removed sample is near the boundary or influences the classification of other points, the decision boundary might move slightly towards or away from the point, depending on its class and proximity. However, if the point is far from the boundary or has minimal influence on it, the decision boundary may remain largely unchanged. For example in SVM classification the only points that influence the boundary are support vectors.

- **For logistic regression:** When a training sample is removed, the decision boundary in logistic regression can change. Logistic regression creates a probabilistic model based on all training samples, so removing one point can affect the overall probabilities, even if the shift might be small. However, the exact change depends on the location and influence of the removed sample. The boundary adjusts to the new data distribution, but there's no need to mention the specific direction of the shift, as the question asks only whether it changes or stays the same.

(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:

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

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

$$\xi_i \geq 0, \quad \forall i \in \{1,\dots,n\}$$

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

Solve the convex optimization problem

$$\begin{aligned}
\mathcal{L}(\omega,\lambda,\alpha) =& \min_{\omega,\xi_i} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{j=1}^{n}\lambda_j(1-\xi_j - y_j(\omega^T x_j)) - \sum_{k=1}^{n}\alpha_k\xi_k \\
\nabla_\omega \mathcal{L}(\omega,\lambda,\alpha) =& \omega - \sum_{j=1}^{n}\lambda_j y_j x_j = 0 \rightarrow \omega = \sum_{j=1}^{n}\lambda_j y_j x_j \\
\nabla_{\xi_i} \mathcal{L}(\omega,\lambda,\alpha) =& C - \lambda_i - \alpha_i = 0 \\
& \dots
\end{aligned}$$

After solving this convex optimization problem, from complementary slackness we have the following:

$$\lambda_i[y_i(\langle\omega_i, x_i\rangle + \omega_0) - 1 + \xi_i] = 0, \quad i = [1,\dots,N]$$

$$\alpha_i\xi_i = 0, \quad i = [1,\dots,N]$$

if $\lambda_i > 0$ then x_i is a support vector thus we have $y_i(\langle\omega_i, x_i\rangle + \omega_0) = 1 - \xi_i$
if $\alpha_i < 0 \implies \xi_i = 0 \quad \& \quad \lambda_i \in (0, C)$
if $\xi_i < 0 \implies x_i$ is missclassified $\quad \& \quad \alpha_i = 0, \lambda_i = C$
if $y_i(\langle\omega_i, x_i\rangle + \omega_0) > 1 - \xi_i$, x_i is not a support vector and classified correctly.
thus the only case where data point crosses margin is when ξ_i is positive.
num of missclassified points $\leq \|\xi\|_0$

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

C multiplier shows our sensitivity on how much data is missclassified or is on the margin.
Consider two extreme states:

- **C = 0** Setting C as 0 means we dont care about how much data is missclassified or is on the margin. We only want to find $\min_{\omega,\xi_i} \frac{1}{2}\|\omega\|^2$.
- **C = ∞** Setting C as ∞ means we can't tolerate the slightest mistake in classification and also no data point can be on the margin. This is the same as hard SVM.

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

Assumption: Two classes are linearly separable.

- Hard SVM focuses on maximizing the margin between two classes and support vectors. To put it another way it means there are only specific data points that affect the SVM boundary.
- Logistic Regression focuses on finding the best boundary based on all data points.

Conclusion:
Hard SVM decides based on sensitive data points and tries to maximize the margin.
Logistic Regression decides based on all data points and provides a suitable probabilistic for different classes.

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

Assumption: Two classes are not linearly separable.

- Soft SVM focuses on maximizing the margin between two classes and support vectors at some cost of missclassification. The difference between Soft SVM and Hard SVM is that Soft SVM allows for some data points to be missclassified for some penalty but Hard SVM does not allow that at all.
- Logistic Regression focuses on finding the best boundary based on all data points.

Conclusion:
Soft SVM decides based on sensitive data points and tries to maximize the margin even at the cost of some missclassification (the cost is determined by the coefficient C)
Logistic Regression decides based on all data points and provides a suitable probabilistic for different classes.

Question 2 (60 Points)

Suppose in PCA, we project each point x_i onto $z_i = V_{1:k}^T x_i$, where $V_{1:k} = [v_1, \dots, v_k]$ 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:

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

First we show $\|\hat{x}_i - \hat{x}_j\|^2 = \|z_i - z_j\|^2$

$$\begin{aligned} \|\hat{x}_i - \hat{x}_j\|^2 =& (\hat{x}_i - \hat{x}_j)^T (\hat{x}_i - \hat{x}_j) = (V_{1:k} z_i - V_{1:k} z_j)^T (V_{1:k} z_i - V_{1:k} z_j) \\ =& (z_i^T - z_j^T) \underbrace{V_{1:k}^T V_{1:k}}_{I_k} (z_i - z_j) = (z_i - z_j)^T (z_i - z_j) = \|z_i - z_j\|^2 \\ \rightarrow& \|\hat{x}_i - \hat{x}_j\|^2 = \|z_i - z_j\|^2 \quad \& \quad \|\hat{x}_i - \hat{x}_j\| = \|z_i - z_j\| \end{aligned}$$

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

$$\sum_{i=1}^{n} \|x_i - \hat{x}_i\|_2^2 = \sum_{i=1}^{n} (x_i - \hat{x}_i)^T (x_i - \hat{x}_i) = \sum_{i=1}^{n} (x_i - V_{1:k} V_{1:k}^T x_i)^T (x_i - V_{1:k} V_{1:k}^T x_i)$$

$$= \sum_{i=1}^{n} x_i^T x_i - x_i^T V_{1:k} V_{1:k}^T x_i - x_i V_{1:k} V_{1:k}^T x_i + x_i^T V_{1:k} \underbrace{V_{1:k}^T V_{1:k}}_{I_k} V_{1:k}^T x_i = \sum_{i=1}^{n} x_i^T x_i - x_i^T V_{1:k} V_{1:k}^T x_i$$

$$= \sum_{i=1}^{n} x_i^T I x_i - x_i^T V_{1:k} V_{1:k}^T x_i = \sum_{i=1}^{n} x_i^T (I - V_{1:k} V_{1:k}^T) x_i = \sum_{i=1}^{n} Tr(x_i^T (I - V_{1:k} V_{1:k}^T) x_i)$$

$$= \sum_{i=1}^{n} Tr((I - V_{1:k} V_{1:k}^T)(x_i^T x_i)) = Tr(I - V_{1:k} V_{1:k}^T) \sum_{i=1}^{n} (x_i^T x_i)$$

$$= (n-1) Tr(I - V_{1:k} V_{1:k}^T) \frac{1}{n-1} \sum_{i=1}^{n} x_i^T x_i$$

we know $\sum_{i=1}^{n} x_i^T x_i$ is the trace of covariance matrix $C = V \Lambda V^T$

$$= (n-1) Tr((I - V_{1:k} V_{1:k}^T) V \Lambda V^T) = (n-1) Tr(V \Lambda V^T - V_{1:k} \Lambda V_{1:k}^T)$$

$$= (n-1) \sum_{i=1}^{n} \lambda_i - \sum_{j=1}^{k} \lambda_j = (n-1) \sum_{i=k+1}^{n} \lambda_i$$

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 Σ is an $m \times n$ matrix with non-zero singular values (σ_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 $\min \|Xw - y\|^2$.

convex optimization problem: $min\|X\omega - y\|^2$

$$\begin{aligned}\|X\omega - y\|^2 =&((X\omega - y)^T(X\omega - y) = \omega^T X^T X\omega - y^T X\omega - \omega^T X^T y + y^T y)\\ \rightarrow&\frac{d}{dw}\|X\omega - y\|^2 = 2(X^TX)\omega - 2X^Ty = 0 \rightarrow 2(X^TX)\omega = 2X^Ty\\ \rightarrow&\omega = (X^TX)^{-1}X^Ty\end{aligned}$$

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

Replace X with SVD form.

$$\begin{aligned}\omega =&(V\Sigma^TU^TU\Sigma V^T)^{-1}V\Sigma^TU^Ty = (V\Sigma^T\Sigma V^T)^{-1}V\Sigma^TU^Ty\\ =&V(\Sigma^T\Sigma)^{-1}V^TV\Sigma^TU^Ty = V(\Sigma^T\Sigma)^{-1}\Sigma^TU^Ty = V\Sigma^\dagger U^Ty\end{aligned}$$

(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 last part we know $A = V\Sigma^\dagger U^T$
If we multiply A by X from left we have:

$$XA = U\Sigma V^TV\Sigma^\dagger U^T = U\Sigma\Sigma^\dagger U^T$$

We know $m > n$ so $\Sigma\Sigma^\dagger = I_{n_{m\times m}}$
(from $I_{n_{m\times m}}$ we mean a $m \times m$ matrix with first n elements 1 and the rest 0)

$$= UI_{n_{m\times m}}U^T$$

This is the projection onto 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?

convex optimization problem:

$$min\|\omega\|^2$$

$$s.t : X\omega = y$$

To solve this we use laugrangian

$$\begin{aligned}\mathcal{L}(\omega,\lambda) =&\omega^T\omega + \lambda^T(X\omega - y)\\ &\rightarrow \nabla_\omega \mathcal{L}(\omega,\lambda) = 2\omega + X^T\lambda = 0 \rightarrow \omega = \frac{-X^T\lambda}{2}\\ y = X\omega = X\frac{-X^T\lambda}{2} &\rightarrow \lambda = -2(XX^T)^{-1}y\\ &\implies \omega = X^T(XX^T)^{-1}y\end{aligned}$$

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

Replace X with SVD form.

$$\begin{aligned}\omega =&V\Sigma^T U^T(U\Sigma \underbrace{V^T V}_{I}\Sigma^T U^T)^{-1}y = V\Sigma^T\underbrace{U^T U}_{I}(\Sigma\Sigma^T)^{-1}U^T y\\ =&V\Sigma^T(\Sigma\Sigma^T)^{-1}U^T y\end{aligned}$$

(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 last part we know $B = V\Sigma^T(\Sigma\Sigma^T)^{-1}U^T$

$$BX = V\Sigma^T(\Sigma\Sigma^T)^{-1}U^T U\Sigma V^T = V\Sigma^T(\Sigma^T)^{\dagger}V^T = VI_{m_{n\times n}}V^T$$

(from $I_{m_{n\times n}}$ we mean a $n \times n$ matrix with first m elements 1 and the rest 0)

$$= VI_{m_{n\times n}}V^T$$

This is the projection onto 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 α 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 ϵ 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 η 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 $t > 0$:

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

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

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

Since the largest eigenvalue of F is α the maximum value of $\|F^T y\|_2$ occurs when y aligns with the eigenvector corresponding to α, giving us:

$$\|F^T y\|_2 \leq \|\alpha y\|_2$$

For the term $\|(I - \eta F^T F)w_{t-1}\|_2$, we know that:

$$\|(I - \eta F^T F)w_{t-1}\|_2 \leq 1 (\text{since all eigenvalues of } I - \eta F^T F \text{ are within (-1,1)})$$

Thus, we have:

$$\|(I - \eta F^T F)w_{t-1}\|_2 \leq \|w_{t-1}\|_2$$

Combining these results:

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

Conclusion: Gradient descent does not diverge if all eigenvalues of $(I - \eta F^T F)$ remain within (-1, 1), ensuring no growth in the weight vector w.

Question 5 (60 Points)

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

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

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

We already know MSE $\triangleq \mathbb{E}[(y - \hat{f}(x))^2]$ and also we have $y = f(x) + \epsilon$
hence we can show

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

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

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

since $\mathbb{E}[\epsilon] = 0$ we have $\mathbb{E}[\epsilon^2] = \sigma^2$

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

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

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

first lets find out about the term $2\mathbb{E}[(f(x) - \mathbb{E}[\hat{f}(x)])(\mathbb{E}[\hat{f}(x)] - \hat{f}(x))]$

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

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

the remaining terms are $\mathbb{E}[(f(x) - \mathbb{E}[\hat{f}(x)])^2] + \mathbb{E}[(\mathbb{E}[\hat{f}(x)] - \hat{f}(x))^2]$

$$\mathbb{E}[(f(x) - \mathbb{E}[\hat{f}(x)])^2] = \underbrace{(f(x) - \mathbb{E}[\hat{f}(x)])^2}_{Bias(\hat{f}(x)^2)} \quad \& \quad \mathbb{E}[(\mathbb{E}[\hat{f}(x)] - \hat{f}(x))^2] = Var(\hat{f}(x))$$

$$\rightarrow \mathbb{E}[(y - \hat{f}(x))^2] = Bias(\hat{f}(x)^2) + Var(\hat{f}(x)) + \sigma^2$$

(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 ϵ_i is independently and identically distributed with zero mean and variance 1. We use the ordinary least squares (OLS) estimator $\hat{\theta}$ to estimate θ 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.

For MSE estimation, we know from part a) that $Error = Bias^2 + Variance + \sigma^2$
First, we calculate $\mathbb{E}[\hat{\theta}]$ & **Bias**

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

Second we calculate $Var(\hat{\theta})$

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

Conclusion: $Error = X(X^T X)^{-1} X^T + \Sigma$