IFT 3395/6390 (6390: GRAD) Theoretical Homework 2

Himanshu Arora, Akshay Singh Rana

TOTAL POINTS

61.5 / 70

QUESTION 1

1 Bias-Variance decomposition 2 / 2

✓ **- 0 pts** **Correct**

QUESTION 2

Feature Maps 8 pts

2.1 (a) 2 / 2

✓ **- 0 pts** **correct**

2.2 (b) 2 / 2

✓ **- 0 pts** **Correct**

2.3 (c) 3.5 / 4

✓ **- 0 pts** **Correct**

- 0.5 **Point adjustment**

The kernel function is not complete

QUESTION 3

Optimization 10 pts

3.1 (a) 1 / 1

✓ **- 0 pts** **Correct**

- 1 pts No answer or page not annotated

- 0.5 pts Incomplete

- 0.5 pts Small error

- 1 pts Incorrect

3.2 (b) 1 / 1

✓ **- 0 pts** **Correct**

- 1 pts No answer or page not annotated

- 0.5 pts Incomplete

- 0.5 pts Small error

- 1 pts Incorrect

3.3 (c) 1 / 1

✓ **- 0 pts** **Correct**

- 1 pts No answer or page not annotated

- 0.5 pts Incomplete

- 0.5 pts Small error

- 1 pts Incorrect

- 1 pts Value, f(x*) not calculated

- 0.5 pts Derivation missing

3.4 (d) 1 / 1

✓ **- 0 pts** **Correct**

- 1 pts No answer or page not annotated

- 0.5 pts Incomplete

- 0.5 pts Small error

- 1 pts Incorrect

- 0.5 pts Dimension not calculated.

3.5 (e) 1 / 1

✓ **- 0 pts** **Correct**

- 1 pts No answer or page not annotated

- 0.5 pts Incomplete

- 0.5 pts Small error

- 1 pts Incorrect

- 0.5 pts No step size.

- 0.5 pts Incorrectly used commutativity for matrix-vector multiplication

- 0.5 pts Applied different method

3.6 (f) 1 / 1

✓ **- 0 pts** **Correct**

- 1 pts No answer or page not annotated

- 0.5 pts Incomplete

- 0.5 pts Small error

- 1 pts Incorrect

- 0.5 pts No step size.

- 0.5 pts Incorrectly used commutativity for matrix-

vector multiplication

- **0.5 pts** Applied different method

3.7 (g) 0.5 / 1

- **0 pts** Correct
- **1 pts** No answer or page not annotated
- **0.5 pts** Incomplete
- ✓ **0.5 pts** **Small error**
- **1 pts** Incorrect
- **1 pts** Value, f(x*) not calculated
- **0.5 pts** Derivation missing

3.8 (h) 0.5 / 1

- **0 pts** Correct
- **1 pts** No answer or page not annotated
- **0.5 pts** Incomplete
- ✓ **0.5 pts** **Small error**
- **1 pts** Incorrect
- **0.5 pts** Derivation missing

3.9 (i) 1 / 1

- ✓ **0 pts** **Correct**
- **1 pts** No answer or page not annotated
- **0.5 pts** Incomplete
- **0.5 pts** Small error
- **1 pts** Incorrect
- **0.5 pts** Derivation missing

3.10 (j) 0 / 1

- **0 pts** Correct
- **1 pts** No answer or page not annotated
- **0.5 pts** Incomplete
- **0.5 pts** Small error
- ✓ **1 pts** **Incorrect**
- **0.5 pts** Derivation missing

QUESTION 4

Least Squares Estimator and Ridge Regression 10 pts

4.1 (a) - (i) 2 / 2

- ✓ **0 pts** **correct**
- **0 pts** correct and makes the argument that the extremum is a minimum
- **2 pts** wrong answer
- **1 pts** algebra error but the argument is right
- **2 pts** no answer
- **0.5 pts** wrong usage of argmin

4.2 (a) - (ii) 2 / 2

- ✓ **0 pts** **correct**
- **2 pts** wrong answer
- **1 pts** does not mention n >= d
- **1 pts** does not mention linear independence
- **2 pts** does not answer in terms of properties of the dataset
- **2 pts** no answer

4.3 (b) - (i) 2 / 2

- ✓ **0 pts** **Correct**
- **1 pts** no/flawed argument for the invertibility of X^TX + lambda
- **2 pts** wrong answer
- **2 pts** no answer
- **1 pts** error in calculation

4.4 (b) - (ii) 2 / 2

- ✓ **0 pts** **Correct**
- **2 pts** wrong answer
- **2 pts** no answer

4.5 (b) - (iii) 2 / 2

- ✓ **0 pts** **Correct**
- **2 pts** wrong answer
- **2 pts** no answer

> we do not keep the value of lambda between 0 and 1

QUESTION 5

Leave one out cross-validation 10 pts

5.1 (a) 2 / 2

- ✓ **0 pts** **Correct**

Correct

- **0 pts** Correct

- **2 pts** Not answer or Cannot find the answer.

- **1 pts** Incomplete or not assessed for new data point.

- **0.5 pts** Sign error or some steps omitted.

5.2 (b) **0.5 / 2**

- **0 pts** Correct
- **2 pts** Incomplete or no answer found
- **1 pts** Error in algebra
- **0.5 pts** Minor Error in algebra or error in formula
- **1.5 pts** No proof
- **0 pts** Click here to replace this description.

✓ **- 1 pts No proof written or proof unclear**

✓ **- 0.5 pts unclear proof**

5.3 (c) **2 / 2**

✓ **- 0 pts Correct**

- **2 pts** Incorrect or answer not found.
- **1.5 pts** Marks for formula
- **1 pts** Marks for formula
- **0.25 pts** Minor calculation error.

5.4 (d) **2 / 2**

✓ **- 0 pts Correct**

- **1 pts** Marks for formula
- **0.25 pts** Minor calculation errors
- **2 pts** Incorrect or answer not found

5.5 (e) **1 / 2**

- **0 pts** Correct
- **2 pts** Incorrect or no answer found.
- **0.25 pts** Minor error in calculation
- **1 pts** Marks for formula
- **1 pts** incomplete
- **0.5 pts** Error in calculation
- **0.75 pts** No computational complexity

✓ **- 1 pts No proof**

QUESTION 6

Multivariate Regression 10 pts

6.1 (a) **2.5 / 3**

- **0 pts** Correct
- **3 pts** no answer
- **2.5 pts** major mistakes in derivation

✓ **- 0.5 pts minor mistake**

- **2 pts** wrong derivation
- **1.5 pts** the trace is missing in the expression of J(W)
- **1.5 pts** gradient derivation is wrong / not clear
- **1 pts** incomplete
- **3 pts** wrong
- **1 pts** mistake in the derivation of the gradient
- **1 pts** two answers are given, only one of them is correct
- **1 pts** dimensions do not match

6.2 (b) **3 / 3**

✓ **- 0 pts Correct**

- **3 pts** no answer
- **1 pts** incomplete answer
- **1 pts** this is not a classification problem
- **2 pts** incomplete answer
- **0.5 pts** correct but not enough justification
- **1 pts** small mistake
- **3 pts** incorrect
- **3 pts** No proof

6.3 (c) **1 / 4**

- **0 pts** Correct
- **4 pts** Wrong / incomplete
- **4 pts** no answer
- **4 pts** major errors in derivations
- **0.5 pts** very good! just a few steps missing
- **1 pts** right idea but the derivations are wrong / incomplete

✓ **- 3 pts Wrong derivations / incomplete**

- **0 pts** Not the optimal solution but a correct answer
- **2 pts** good direction but not enough details

QUESTION 7

Practical Report : One-versus-all, L2 loss SVM 20 pts

7.1 Derivative Regularization 5 / 5

✓ **- 0 pts** **Correct**

7.2 Derivative of Hinge Loss 10 / 10

✓ **- 0 pts** **Correct**

Missing a minus sign (EDIT: the minus sign is there after all)

7.3 SVM plots loss/accuracy vs epoch 5 / 5

✓ **- 0 pts** **Correct**

1 Show that the expected prediction error on(x,y)can be decomposed into a sum of 3 terms: $(bias)^2$,variance, and a noise term involving.You need to justify all the steps in your derivation.

Answer 1
As stated in the question,

$$bias = \mathbb{E}[h_D(x')] - f(x') \tag{1}$$

$$variance = \mathbb{E}[(h_D(x') - \mathbb{E}[h_D(x')])^2] \tag{2}$$

$$\begin{aligned}
\mathbb{E}[(h_D(x') - y')^2] &= \mathbb{E}[(h_D(x') - (f(x') + \epsilon))^2] \\
&= \mathbb{E}[(h_D(x') - (f(x') + \epsilon) + \mathbb{E}[h_D(x')] - + \mathbb{E}[h_D(x')])^2] \\
&= \mathbb{E}[((h_D(x') - \mathbb{E}[h_D(x')]) + (\mathbb{E}[h_D(x')] - (f(x')) + \epsilon))^2] \\
&= \mathbb{E}[(h_D(x') - \mathbb{E}[h_D(x')])^2] + \mathbb{E}[(\mathbb{E}[h_D(x')] - f(x'))^2] + \mathbb{E}[\epsilon^2] \\
&+ 2\,\mathbb{E}[h_D(x') - \mathbb{E}[h_D(x')])\,\mathbb{E}[\mathbb{E}[h_D(x')] - f(x')] \\
&+ 2\,\mathbb{E}[h_D(x') - \mathbb{E}[h_D(x']]\,\mathbb{E}[\epsilon] \\
&+ 2\,\mathbb{E}[\mathbb{E}[h_D(x')] - f(x')]\,\mathbb{E}[\epsilon]
\end{aligned}$$

Using $\mathbb{E}[\epsilon] = 0$

Using $\mathbb{E}[x - \mathbb{E}[x]] = 0$

$\mathbb{E}[(h_D(x')] - f(x')$ is a constant

$$\begin{aligned}
\mathbb{E}[(h_D(x') - y')^2] &= \mathbb{E}[(h_D(x') - \mathbb{E}[h_D(x')])^2] + \mathbb{E}[(\mathbb{E}[h_D(x')] - f(x'))^2] + \mathbb{E}[\epsilon^2] \\
&= Var(h_D(x')) + Bias(h_D(x'))^2 + \mathbb{E}[\epsilon^2] \\
&= Variance + Bias^2 + \mathbb{E}[\epsilon^2]
\end{aligned}$$

1 Bias-Variance decomposition 2 / 2

✓ - 0 pts Correct

1. **Feature Maps** [8 points]

   In this exercise, you will design feature maps to transform an original dataset into a linearly separable set of points. For the following questions, if your answer is '*yes*', write the expression for the proposed transformation; and if your answer is '*no*', write a brief explanation. You are expected to provide explicit formulas for the feature maps, and these formulas should only use common mathematical operations.

   (a) [2 points] Consider the following 1-D dataset (Figure 1). Can you propose a 1-D transformation that will make the points linearly separable?

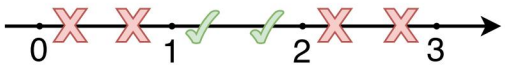


Figure 1:

Answer 2.a. The 1D transformation

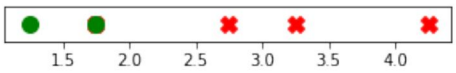


Figure 2: Linearly separable 1D transformation of the dataset in Question 2.a.

$$f : X \rightarrow |X - 1| + 2|X - 2|$$

transforms the given 1D dataset into a linearly separable dataset.

   (b) [2 points] Consider the following 2-D dataset (Figure 6). Can you propose a 1-D transformation that will make the data linearly separable?

Answer 2.b.
The 1D transformation

$$f : (X_1, X_2) \rightarrow X_1^2 + X_2^2$$

transforms the given 2D dataset into a linearly separable dataset.

2.1 (a) **2 / 2**

✓ **- 0 pts** correct

1. **Feature Maps** [8 points]

   In this exercise, you will design feature maps to transform an original dataset into a linearly separable set of points. For the following questions, if your answer is '*yes*', write the expression for the proposed transformation; and if your answer is '*no*', write a brief explanation. You are expected to provide explicit formulas for the feature maps, and these formulas should only use common mathematical operations.

   (a) [2 points] Consider the following 1-D dataset (Figure 1). Can you propose a 1-D transformation that will make the points linearly separable?

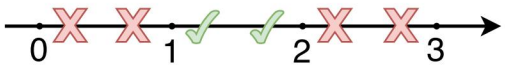


Figure 1:

Answer 2.a. The 1D transformation

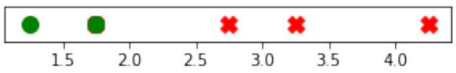


Figure 2: Linearly separable 1D transformation of the dataset in Question 2.a.

$$f : X \rightarrow |X - 1| + 2|X - 2|$$

transforms the given 1D dataset into a linearly separable dataset.

   (b) [2 points] Consider the following 2-D dataset (Figure 6). Can you propose a 1-D transformation that will make the data linearly separable?

Answer 2.b.
The 1D transformation

$$f : (X_1, X_2) \rightarrow X_1^2 + X_2^2$$

transforms the given 2D dataset into a linearly separable dataset.

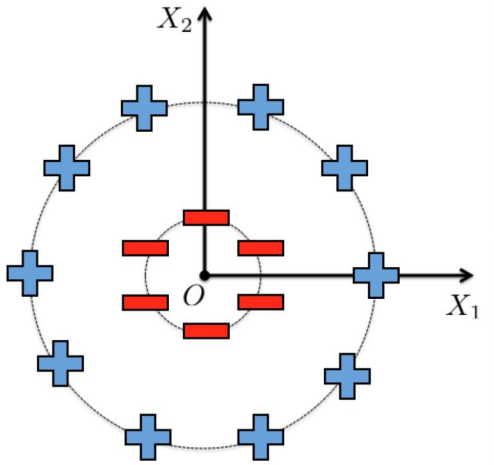


Figure 3:

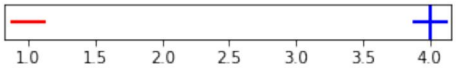


Figure 4: Linearly separable 1D transformation of the dataset in Question 2.b.

(c) [4 points] Using ideas from the above two datasets, can you suggest a 2-D transformation of the following dataset (as shown in Figure 5) that makes it linearly separable? If '*yes*', also provide the kernel corresponding to the feature map you proposed.

Answer 2.c.

Let r_2 be the radius of the middle circle, the 1D transformation f that makes the given 2D dataset linearly separable is then given by:

$$f : (X_1, X_2) \rightarrow |X_1^2 + X_2^2 - r_2^2|$$

The Kernel function K for the transformation f is defined as the dot product of the original data points in the transformed space of f, given by:

$$K(X, Y) = \langle |X_1^2 + X_2^2 - r_2^2|, |Y_1^2 + Y_2^2 - r_2^2| \rangle$$

It is a valid kernel because:

2.2 (b) **2 / 2**

✓ **- 0 pts** Correct

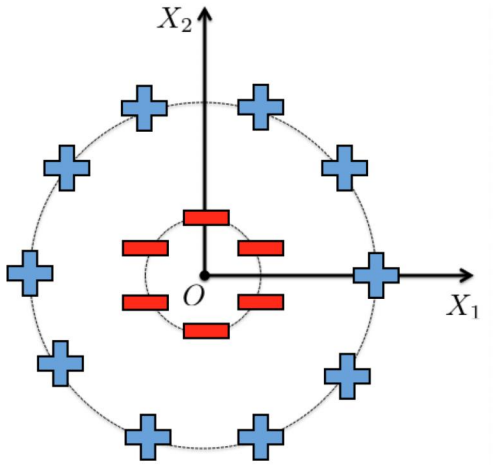


Figure 3:

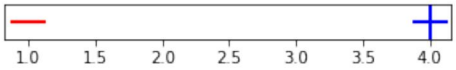


Figure 4: Linearly separable 1D transformation of the dataset in Question 2.b.

(c) [4 points] Using ideas from the above two datasets, can you suggest a 2-D transformation of the following dataset (as shown in Figure 5) that makes it linearly separable? If '*yes*', also provide the kernel corresponding to the feature map you proposed.

Answer 2.c.

Let r_2 be the radius of the middle circle, the 1D transformation f that makes the given 2D dataset linearly separable is then given by:

$$f : (X_1, X_2) \rightarrow |X_1^2 + X_2^2 - r_2^2|$$

The Kernel function K for the transformation f is defined as the dot product of the original data points in the transformed space of f, given by:

$$K(X, Y) = \langle |X_1^2 + X_2^2 - r_2^2|, |Y_1^2 + Y_2^2 - r_2^2| \rangle$$

It is a valid kernel because:

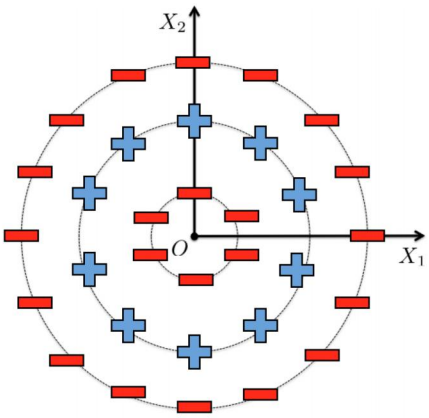


Figure 5:

Figure 6: Linearly separable 1D transformation of the dataset in Question 2.c.

- It is always positive since it is a dot product of two positive numbers.
- It is symmetric since $\langle |X_1^2 + X_2^2 - r_2^2|, |Y_1^2 + Y_2^2 - r_2^2| \rangle$ is equal to $\langle |Y_1^2 + Y_2^2 - r_2^2|, |X_1^2 + X_2^2 - r_2^2| \rangle$.
- It is continuous.
- It is a dot product in a valid transformation space of the given data.

2.3 (c) **3.5 / 4**

✓ **- 0 pts** **Correct**

- 0.5 **Point adjustment**

The kernel function is not complete

1. **Optimization** [10 points]

Assume a quadratic objective function of the form:

$$f(x) = \frac{1}{2}x^T Ax + x^T b + a,$$

where $x \in \mathbb{R}^d$, $b \in \mathbb{R}^d$, $a \in \mathbb{R}$ and A is a $d \times d$ symmetric, positive definite matrix. This means that the matrix A admits the eigendecomposition $A = U\Lambda U^T$, where $U \in \mathbb{R}^{d\times d}$ is an orthonormal matrix and $\Lambda \in \mathbb{R}^{d\times d}$ is a diagonal matrix. The i-th column vector of U, denoted by $u_i \in \mathbb{R}^d$, represents the i-th eigenvector of A. The i-th diagonal element of Λ, denoted by $\lambda_i \in \mathbb{R}$, represents the i-th eigenvalue of A. We will assume here that all eigenvalues are unique. Furthermore, without loss of generality, the eigenvectors and eigenvalues in the decomposition can be ordered in such a way that

$$\lambda_1 > \lambda_2 > \ldots > \lambda_d > 0.$$

(a) Find all of the stationary points of $f(x)$ analytically, i.e. through a closed-form expression (Justify).
Answer 3.a.
The derivative of the function f is given by:

$$\begin{aligned} f'(x) &= \frac{d}{dx}(\frac{1}{2}x^T Ax + x^T b + a) \\ &= \frac{1}{2}[Ax + A^T x] + b \\ &= \frac{A + A^T}{2}x + b \\ &= Ax + b \end{aligned}$$

Stationary points are the points on which the derivative is 0.

$$\begin{aligned} f'(x) &= 0 \\ Ax + b &= 0 \\ x &= -A^{-1}b \end{aligned}$$

Since f is a quadratic objective function, x is its only stationary point.

3.1 (a) **1 / 1**

✓ **- 0 pts Correct**

- 1 pts No answer or page not annotated

- 0.5 pts Incomplete

- 0.5 pts Small error

- 1 pts Incorrect

(b) Which of those stationary points are minima, maxima and saddle-points? Give a mathematically rigorous explanation why.
Answer 3.b.
Taking second derivative of f,

$$\begin{aligned} f''(x) &= \frac{d^2}{dx^2}(\frac{1}{2}x^T Ax + x^T b + a) \\ &= \frac{d}{dx}(Ax + b) \\ &= A \end{aligned}$$

Since A is a positive semi-definite matrix, the above second derivative is always non-negative in its entire domain. This means that f is an upwards-open convex function and therefore, the stationary point x is a minima.

(c) Find the location, x^*, and value, $f(x^*)$, of the global minimum.
Answer 3.c.
As proven in 3.a. and 3.b., the function f has only one minima at $x^* = -A^{-1}b$. The value of f at x_* is:

$$\begin{aligned} f(x^*) &= \frac{1}{2}(x^*)^T Ax^* + (x^*)^T b + a \\ &= \frac{1}{2}(-A^{-1}b)^T A(-A^{-1}b) + (-A^{-1}b)^T b + a \\ &= \frac{1}{2}b^T(A^{-1})^T AA^{-1}b - b^T(A^{-1})^T b + a \\ &= \frac{1}{2}b^T A^{-1}AA^{-1}b - b^T A^{-1}b + a \\ &= \frac{1}{2}b^T A^{-1}b - b^T A^{-1}b + a \\ &= -\frac{1}{2}b^T A^{-1}b + a \end{aligned}$$

(d) Find the gradient of $f(x)$ at some point x. What are the dimensions of the gradient?
Answer 3.d.
As proven in 3.a., $\triangledown f(x) = Ax + b$. The gradient vector is of $d \times 1$ dimensions.

3.2 (b) 1 / 1

✓ **- 0 pts Correct**

- 1 pts No answer or page not annotated

- 0.5 pts Incomplete

- 0.5 pts Small error

- 1 pts Incorrect

(b) Which of those stationary points are minima, maxima and saddle-points? Give a mathematically rigorous explanation why.
Answer 3.b.
Taking second derivative of f,

$$\begin{aligned} f''(x) &= \frac{d^2}{dx^2}(\frac{1}{2}x^T Ax + x^T b + a) \\ &= \frac{d}{dx}(Ax + b) \\ &= A \end{aligned}$$

Since A is a positive semi-definite matrix, the above second derivative is always non-negative in its entire domain. This means that f is an upwards-open convex function and therefore, the stationary point x is a minima.

(c) Find the location, x^*, and value, $f(x^*)$, of the global minimum.
Answer 3.c.
As proven in 3.a. and 3.b., the function f has only one minima at $x^* = -A^{-1}b$. The value of f at x_* is:

$$\begin{aligned} f(x^*) &= \frac{1}{2}(x^*)^T Ax^* + (x^*)^T b + a \\ &= \frac{1}{2}(-A^{-1}b)^T A(-A^{-1}b) + (-A^{-1}b)^T b + a \\ &= \frac{1}{2}b^T(A^{-1})^T AA^{-1}b - b^T(A^{-1})^T b + a \\ &= \frac{1}{2}b^T A^{-1}AA^{-1}b - b^T A^{-1}b + a \\ &= \frac{1}{2}b^T A^{-1}b - b^T A^{-1}b + a \\ &= -\frac{1}{2}b^T A^{-1}b + a \end{aligned}$$

(d) Find the gradient of $f(x)$ at some point x. What are the dimensions of the gradient?
Answer 3.d.
As proven in 3.a., $\triangledown f(x) = Ax + b$. The gradient vector is of $d \times 1$ dimensions.

3.3 (c) **1 / 1**

✓ **- 0 pts Correct**

- 1 pts No answer or page not annotated

- 0.5 pts Incomplete

- 0.5 pts Small error

- 1 pts Incorrect

- 1 pts Value, f(x*) not calculated

- 0.5 pts Derivation missing

(b) Which of those stationary points are minima, maxima and saddle-points? Give a mathematically rigorous explanation why.
Answer 3.b.
Taking second derivative of f,

$$\begin{aligned} f''(x) &= \frac{d^2}{dx^2}(\frac{1}{2}x^T Ax + x^T b + a) \\ &= \frac{d}{dx}(Ax + b) \\ &= A \end{aligned}$$

Since A is a positive semi-definite matrix, the above second derivative is always non-negative in its entire domain. This means that f is an upwards-open convex function and therefore, the stationary point x is a minima.

(c) Find the location, x^*, and value, $f(x^*)$, of the global minimum.
Answer 3.c.
As proven in 3.a. and 3.b., the function f has only one minima at $x^* = -A^{-1}b$. The value of f at x_* is:

$$\begin{aligned} f(x^*) &= \frac{1}{2}(x^*)^T Ax^* + (x^*)^T b + a \\ &= \frac{1}{2}(-A^{-1}b)^T A(-A^{-1}b) + (-A^{-1}b)^T b + a \\ &= \frac{1}{2}b^T(A^{-1})^T AA^{-1}b - b^T(A^{-1})^T b + a \\ &= \frac{1}{2}b^T A^{-1}AA^{-1}b - b^T A^{-1}b + a \\ &= \frac{1}{2}b^T A^{-1}b - b^T A^{-1}b + a \\ &= -\frac{1}{2}b^T A^{-1}b + a \end{aligned}$$

(d) Find the gradient of $f(x)$ at some point x. What are the dimensions of the gradient?
Answer 3.d.
As proven in 3.a., $\triangledown f(x) = Ax + b$. The gradient vector is of $d \times 1$ dimensions.

3.4 (d) 1 / 1

✓ **- 0 pts Correct**

- 1 pts No answer or page not annotated

- 0.5 pts Incomplete

- 0.5 pts Small error

- 1 pts Incorrect

- 0.5 pts Dimension not calculated.

(e) Show how the gradient descent update rule looks like in this case by substituting $f(x)$ with its quadratic form above. Use the following notation: x_0 represents our point at initialization, x_1 represents our point after one step, etc.

Answer 3.e.
A gradient descent update step is performed by going in the opposite direction of the gradient at the current point i.e. if x_0 is the initialization point,

$$\begin{aligned} x_1 &= x_0 - \eta \nabla f(x) \\ &= x_0 - \eta(Ax_0 + b) \\ &= (I - \eta A)x_0 - \eta b \\ x_2 &= (I - \eta A)x_1 - \eta b \\ &\vdots \\ x_k &= (I - \eta A)x_{k-1} - \eta b \end{aligned}$$

(f) Consider the squared distance from optimum, $d(x_k) = \|x_k - x^*\|_2^2$. Find an exact expression (equality) of $d(x_k)$ that only depends on x_0 (not on other iterates x_i for $i > 0$), the number of iterations, k, as well as the eigenvectors, u_i, and eigenvalues, λ_i of A.

Answer 3.f.

3.5 (e) 1 / 1

✓ **- 0 pts Correct**

- 1 pts No answer or page not annotated

- 0.5 pts Incomplete

- 0.5 pts Small error

- 1 pts Incorrect

- 0.5 pts No step size.

- 0.5 pts Incorrectly used commutativity for matrix-vector multiplication

- 0.5 pts Applied different method

As proven in 3.e.,

$$\begin{aligned}
x_k &= (I - \eta A)x_{k-1} - \eta b \\
&= (I - \eta A)[(I - \eta A)x_{k-2} - \eta b] - \eta b \\
&= (I - \eta A)^2 x_{k-2} - \eta(I + I - \eta A)b \\
&= (I - \eta A)^2[(I - \eta A)x_{k-3} - \eta b] - \eta[I + I - \eta A]b \\
&= (I - \eta A)^3 x_{k-3} - \eta[(I - \eta A)^0 + (I - \eta A)^1 + (I - \eta A)^2]b \\
&\vdots \\
&= (I - \eta A)^k x_0 - \eta[(I - \eta A)^0 + (I - \eta A)^1 + \ldots + (I - \eta A)^{k-1}]b \\
&= (I - \eta A)^k x_0 - \eta \frac{(I - \eta A)^k - I}{I - \eta A - I} b \\
&= (I - \eta A)^k x_0 + [(I - \eta A)^k - I]A^{-1}b \\
&= (I - \eta A)^k (x_0 + A^{-1}b) - A^{-1}b \\
&= (I - \eta A)^k (x_0 - x_*) + x_* \\
x_k - x_* &= (I - \eta A)^k (x_0 - x_*) \\
&= (I - \eta U \Lambda U^T)^k (x_0 - x_*)
\end{aligned}$$

Taking Euclidean norm and squaring both the sides,

$$d(x_k) = \|x_k - x_*\|_2^2 = \|(I - \eta U \Lambda U^T)^k (x_0 - x_*)\|_2^2$$

(g) Prove that there exist some assumptions on the hyperparameters of the algorithm, under which the sequence $d(x_k)$ converges to 0 as k goes to infinity. What are the exact necessary and sufficient conditions on the hyperparameters in order for $d(x_k)$ to converge to 0?

Answer 3.g.
As proven in 3.f.,

$$d(x_k) = \|x_k - x_*\|_2^2 = \|(I - \eta U \Lambda U^T)^k (x_0 - x_*)\|_2^2$$

3.6 (f) 1 / 1

✓ **- 0 pts Correct**

- 1 pts No answer or page not annotated

- 0.5 pts Incomplete

- 0.5 pts Small error

- 1 pts Incorrect

- 0.5 pts No step size.

- 0.5 pts Incorrectly used commutativity for matrix-vector multiplication

- 0.5 pts Applied different method

As proven in 3.e.,

$$\begin{aligned}
x_k &= (I - \eta A)x_{k-1} - \eta b \\
&= (I - \eta A)[(I - \eta A)x_{k-2} - \eta b] - \eta b \\
&= (I - \eta A)^2 x_{k-2} - \eta(I + I - \eta A)b \\
&= (I - \eta A)^2[(I - \eta A)x_{k-3} - \eta b] - \eta[I + I - \eta A]b \\
&= (I - \eta A)^3 x_{k-3} - \eta[(I - \eta A)^0 + (I - \eta A)^1 + (I - \eta A)^2]b \\
&\vdots \\
&= (I - \eta A)^k x_0 - \eta[(I - \eta A)^0 + (I - \eta A)^1 + \ldots + (I - \eta A)^{k-1}]b \\
&= (I - \eta A)^k x_0 - \eta \frac{(I - \eta A)^k - I}{I - \eta A - I} b \\
&= (I - \eta A)^k x_0 + [(I - \eta A)^k - I]A^{-1}b \\
&= (I - \eta A)^k (x_0 + A^{-1}b) - A^{-1}b \\
&= (I - \eta A)^k (x_0 - x_*) + x_* \\
x_k - x_* &= (I - \eta A)^k (x_0 - x_*) \\
&= (I - \eta U \Lambda U^T)^k (x_0 - x_*)
\end{aligned}$$

Taking Euclidean norm and squaring both the sides,

$$d(x_k) = \|x_k - x_*\|_2^2 = \|(I - \eta U \Lambda U^T)^k (x_0 - x_*)\|_2^2$$

(g) Prove that there exist some assumptions on the hyperparameters of the algorithm, under which the sequence $d(x_k)$ converges to 0 as k goes to infinity. What are the exact necessary and sufficient conditions on the hyperparameters in order for $d(x_k)$ to converge to 0?

Answer 3.g.
As proven in 3.f.,

$$d(x_k) = \|x_k - x_*\|_2^2 = \|(I - \eta U \Lambda U^T)^k (x_0 - x_*)\|_2^2$$

Using Cauchy–Schwarz inequality,

$$\begin{aligned}
\|(I-\eta U\Lambda U^T)^k(x_0-x_*)\|_2^2 &\le \|(I-\eta U\Lambda U^T)^k\|_2^2\|x_0-x_*\|_2^2\\
d(x_k) &\le \|(I-\eta U\Lambda U^T)^k\|_2^2\|x_0-x_*\|_2^2\\
&\le \|(UU^T-\eta U\Lambda U^T)^k\|_2^2\|x_0-x_*\|_2^2\\
&\le \|(U(I-\eta\Lambda)U^T)^k\|_2^2\|x_0-x_*\|_2^2\\
&\le \|U^k\|_2^2\|(I-\eta\Lambda)^k\|_2^2\|U^{Tk}\|_2^2\|x_0-x_*\|_2^2\\
&\le C\|(I-\eta\Lambda)^k\|_2^2\\
&\le C\sum_{i=1}^{d}(1-\eta\lambda_i)^{2k}\\
&< Cd\max_{i\in[1,d]}((1-\eta\lambda_i)^{2k})\\
&< Cd(1-\eta\lambda_d)^{2k}
\end{aligned}$$

$d(x_k)\to 0$ when $k\to\infty$ if and only if,

$$\begin{aligned}
|1-\eta\lambda_d| &< 1\\
-1 < 1-\eta\lambda_d &< 1\\
-2 < -\eta\lambda_d &< 0\\
2 > \eta\lambda_d &> 0\\
0 < \eta &< \frac{2}{\lambda_d}
\end{aligned}$$

(h) The distance that you computed above is said to converge to 0 at an exponential rate (some other research communities use the term linear rate for the same type of convergence). We often care about the asymptotic rate of convergence, defined for this squared distance as

$$\rho = \exp\left(\lim_{k\to\infty}\frac{1}{2k}\ln d(x_k)\right),$$

where $\ln(\cdot)$ denotes the natural logarithm. Keep in mind that this rate depends on both the objective function, but also on the choice of hyperparameters.

Find an expression of ρ that only depends on the eigenvalues of A and the hyperparameter values.

3.7 (g) **0.5 / 1**

- **0 pts** Correct
- **1 pts** No answer or page not annotated
- **0.5 pts** Incomplete

✓ **- 0.5 pts** **Small error**

- **1 pts** Incorrect
- **1 pts** Value, f(x*) not calculated
- **0.5 pts** Derivation missing

Answer 3.h.
As proven in 3.g.,

$$d(x_k) \leq Cd(1-\eta\lambda_d)^{2k}$$

Since ln is a monotonically increasing function, we can take ln on both the sides, without affecting the inequality,

$$\begin{aligned}\ln d(x_k) &\leq \ln Cd(1-\eta\lambda_d)^{2k} \\ &\leq \ln C + \ln d + 2k\ln(1-\eta\lambda_d)\end{aligned}$$

Dividing both the sides by 2k,

$$\ln d(x_k) \leq \frac{\ln C}{2k} + \frac{\ln d}{2k} + \ln(1-\eta\lambda_d)$$

Applying limit $k \to \infty$,

$$\begin{aligned}\lim_{k\to\infty} \ln d(x_k) &\leq \lim_{k\to\infty} (\frac{\ln C}{2k} + \frac{\ln d}{2k} + \ln(1-\eta\lambda_d)) \\ \lim_{k\to\infty} \ln d(x_k) &\leq \ln(1-\eta\lambda_d)\end{aligned}$$

Exponentiating both the sides,

$$e^{\lim_{k\to\infty} \ln d(x_k)} \leq 1-\eta\lambda_d \qquad \ldots(1)$$

Also, according to matrix norm equivalency,

$$\begin{aligned}d(x_k) = \|(I-\eta U\Lambda U^T)^k(x_0-x_*)\|_2^2 &\geq \frac{\|(I-\eta U\Lambda U^T)^k\|^2\|x_0-x_*\|^2}{d} \\ &\geq \frac{\|U^k\|^2\|(I-\eta\Lambda)^k\|^2\|U^{Tk}\|^2\|x_0-x_*\|^2}{d} \\ &\geq \frac{C\|(I-\eta\Lambda)^k\|^2}{d} \\ &\geq \frac{C\max_{i\in[1,d]}((1-\eta\lambda_i)^{2k})}{d} \\ &\geq \frac{C(1-\eta\lambda_d)^{2k}}{d}\end{aligned}$$

Taking ln on both the sides,

$$\begin{aligned}\ln d(x_k) &\geq \ln(1-\eta\lambda_d)^{2k} + \ln C - \ln d \\ &\geq 2k\ln(1-\eta\lambda_d) + \ln C - \ln d\end{aligned}$$

Dividing by $2k$ and applying $\lim k \to \infty$ on both the sides,

$$\lim_{k\to\infty} \ln d(x_k) \geq \lim_{k\to\infty} (\ln(1-\eta\lambda_d) + \frac{\ln C}{2k} - \frac{\ln d}{2k})$$
$$\lim_{k\to\infty} \ln d(x_k) \geq \ln(1-\eta\lambda_d)$$

Exponentiating both the sides,

$$e^{\lim_{k\to\infty} \ln d(x_k)} \geq 1-\eta\lambda_d \qquad \ldots(2)$$

Using (1) and (2), we can say that,

$$\rho = e^{\lim_{k\to\infty} \ln d(x_k)} = 1-\eta\lambda_d$$

(i) Prove that, for any choice of hyperparameter values, there exist constants $k_0 \geq 0$ and $C > 0$ such that

$$d(x_k) \leq C\rho^{2k}, \quad \forall k > k_0.$$

Answer 3.i.
As proven in 3.g.,

$$d(x_k) \leq Cd(1-\eta\lambda_d)^{2k}$$

Also, as proven in 3.h.,

$$\rho = 1-\eta\lambda_d$$

Hence,

$$d(x_k) \leq C\rho^{2k}$$

(j) Based on the above, we gather that in order to get fast convergence, we need a small ρ value. Find a value for the hyperparameter(s) of gradient descent that achieves the fastest asymptotic convergence rate possible.

3.8 (h) **0.5 / 1**

- **0 pts** Correct
- **1 pts** No answer or page not annotated
- **0.5 pts** Incomplete

✓ - **0.5 pts** **Small error**

- **1 pts** Incorrect
- **0.5 pts** Derivation missing

Dividing by $2k$ and applying $\lim k \to \infty$ on both the sides,

$$\lim_{k\to\infty} \ln d(x_k) \geq \lim_{k\to\infty} (\ln(1-\eta\lambda_d) + \frac{\ln C}{2k} - \frac{\ln d}{2k})$$
$$\lim_{k\to\infty} \ln d(x_k) \geq \ln(1-\eta\lambda_d)$$

Exponentiating both the sides,

$$e^{\lim_{k\to\infty} \ln d(x_k)} \geq 1-\eta\lambda_d \qquad \ldots(2)$$

Using (1) and (2), we can say that,

$$\rho = e^{\lim_{k\to\infty} \ln d(x_k)} = 1-\eta\lambda_d$$

(i) Prove that, for any choice of hyperparameter values, there exist constants $k_0 \geq 0$ and $C > 0$ such that

$$d(x_k) \leq C\rho^{2k}, \quad \forall k > k_0.$$

Answer 3.i.
As proven in 3.g.,

$$d(x_k) \leq Cd(1-\eta\lambda_d)^{2k}$$

Also, as proven in 3.h.,

$$\rho = 1-\eta\lambda_d$$

Hence,

$$d(x_k) \leq C\rho^{2k}$$

(j) Based on the above, we gather that in order to get fast convergence, we need a small ρ value. Find a value for the hyperparameter(s) of gradient descent that achieves the fastest asymptotic convergence rate possible.

3.9 (i) **1 / 1**

✓ **- 0 pts Correct**

- 1 pts No answer or page not annotated

- 0.5 pts Incomplete

- 0.5 pts Small error

- 1 pts Incorrect

- 0.5 pts Derivation missing

Answer 3.j.
As proven in 3.h.,

$$\rho = 1 - \eta\lambda_d$$

The asymptotic convergence rate will be the fastest when ρ will be as close to 0 as possible:

$$\begin{aligned} \rho &= 1 - \eta\lambda_d = 0 \\ \eta\lambda_d &= 1 \\ \eta &= \frac{1}{\lambda_d} \end{aligned}$$

3.10 (j) **0 / 1**

- **0 pts** Correct
- **1 pts** No answer or page not annotated
- **0.5 pts** Incomplete
- **0.5 pts** Small error

✓ **- 1 pts Incorrect**

- **0.5 pts** Derivation missing

1. **Least Squares Estimator and Ridge Regression** [10 points]

(a) In the problem of linear regression, we are given n observations $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where each input $\mathbf{x}_i$ is a d-dimensional vector. Our goal is to estimate a linear predictor $f(.)$ which predicts y given $\mathbf{x}$ according to the formula

$$f(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\theta}, \tag{1}$$

Let $\mathbf{y} = [y_1, y_2 \ldots y_n]^\top$ be the $n \times 1$ vector of outputs and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n]^\top$ be the $n \times d$ matrix of inputs. One possible way to estimate the parameter $\boldsymbol{\theta}$ is through minimization of the sum of squares. This is the least squares estimator:

$$\arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2. \tag{2}$$

i. Show that the solution of this minimization problem is given by

$$\boldsymbol{\theta}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}.$$

Answer 4.i.

$$\begin{aligned} J(\theta) &= \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2. \\ &= (y - X\theta)^\top (y - X\theta) \\ &= (y^\top - \theta^\top X^\top)(y - X\theta) \\ &= y^\top y - \theta^\top X^\top y - y^\top X\theta + \theta^\top X^\top X\theta \end{aligned}$$

Since the gradient of J will be 0 at its minimum,

$$\begin{aligned} \frac{\partial J}{\partial \theta} &= \frac{\partial (y^\top y - \theta^\top X^\top y - y^\top X\theta + \theta^\top X^\top X\theta)}{\partial \theta} = 0 \\ \frac{\partial J}{\partial \theta} &= 2X^\top X\theta - 2X^\top y = 0 \\ X^\top X\theta &= X^\top y \end{aligned}$$

Multiplying both the sides by $(X^\top X)^{-1}$ (assuming that the inverse exits)

$$\theta = (X^\top X)^{-1} X^\top y$$

4.1 (a) - (i) **2 / 2**

✓ **- 0 pts correct**

- 0 pts correct and makes the argument that the extremum is a minimum

- 2 pts wrong answer

- 1 pts algebra error but the argument is right

- 2 pts no answer

- 0.5 pts wrong usage of argmin

ii. When will the matrix $\mathbf{X}^\top\mathbf{X}$ be invertible and when will it be non-invertible? Give your answer in terms of properties of the dataset.

Answer 4.ii.
A non-invertible square matrix is a singular matrix whose determinant is zero. In terms of the dataset, matrix X can be non-invertible if we have some linearly dependent (redundant) features or if the number of features are more than the number of samples.

(b) A variation of the least squares estimation problem known as ridge regression considers the following optimization problem:

$$\arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2 \tag{3}$$

where $\lambda > 0$ is a regularization parameter. The regularizing term penalizes large components in $\boldsymbol{\theta}$ which causes the optimal $\boldsymbol{\theta}$ to have a smaller norm.

i. Derive the solution of the ridge regression problem. Do we still have to worry about the invertibility of $\mathbf{X}^\top\mathbf{X}$?
Answer 4.b.i.

$$J(\theta) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda\|\theta\|_2^2$$
$$J(\theta) = (y - X\theta)^\top(y - X\theta) + \lambda\theta^\top\theta$$

The gradient of J

will be 0 at the solution

$$\frac{\partial J}{\partial \theta} = \frac{\partial(y^\top y - \theta^\top X^\top y - y^\top X\theta + \theta^\top X^\top X\theta + \lambda\theta^\top\theta)}{\partial\theta} = 0$$
$$\frac{\partial J}{\partial \theta} = 2X^\top X\theta - 2X^\top y + 2\lambda\theta = 0$$
$$X^\top X\theta + \lambda\theta = X^\top y$$
$$(X^\top X + \lambda I)\theta = X^\top y$$

Since $\lambda > 0$, the inverse $(X^\top X + \lambda I)^{-1}$ exists. Multiplying it to both sides,

$$\theta = (X^\top X + \lambda I)^{-1}X^\top y$$

As long as the regularization term λ is strictly above zero, we don't have to worry about the non-invertibility of $X^\top X$

4.2 (a) - (ii) 2 / 2

✓ **- 0 pts correct**

- 2 pts wrong answer

- 1 pts does not mention n >= d

- 1 pts does not mention linear independence

- 2 pts does not answer in terms of properties of the dataset

- 2 pts no answer

ii. When will the matrix $\mathbf{X}^\top\mathbf{X}$ be invertible and when will it be non-invertible? Give your answer in terms of properties of the dataset.

Answer 4.ii.
A non-invertible square matrix is a singular matrix whose determinant is zero. In terms of the dataset, matrix X can be non-invertible if we have some linearly dependent (redundant) features or if the number of features are more than the number of samples.

(b) A variation of the least squares estimation problem known as ridge regression considers the following optimization problem:

$$\arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2 \tag{3}$$

where $\lambda > 0$ is a regularization parameter. The regularizing term penalizes large components in $\boldsymbol{\theta}$ which causes the optimal $\boldsymbol{\theta}$ to have a smaller norm.

i. Derive the solution of the ridge regression problem. Do we still have to worry about the invertibility of $\mathbf{X}^\top\mathbf{X}$?
Answer 4.b.i.

$$J(\theta) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda\|\theta\|_2^2$$
$$J(\theta) = (y - X\theta)^\top(y - X\theta) + \lambda\theta^\top\theta$$

The gradient of J

will be 0 at the solution

$$\frac{\partial J}{\partial \theta} = \frac{\partial(y^\top y - \theta^\top X^\top y - y^\top X\theta + \theta^\top X^\top X\theta + \lambda\theta^\top\theta)}{\partial\theta} = 0$$
$$\frac{\partial J}{\partial \theta} = 2X^\top X\theta - 2X^\top y + 2\lambda\theta = 0$$
$$X^\top X\theta + \lambda\theta = X^\top y$$
$$(X^\top X + \lambda I)\theta = X^\top y$$

Since $\lambda > 0$, the inverse $(X^\top X + \lambda I)^{-1}$ exists. Multiplying it to both sides,

$$\theta = (X^\top X + \lambda I)^{-1}X^\top y$$

As long as the regularization term λ is strictly above zero, we don't have to worry about the non-invertibility of $X^\top X$

as the matrix formed after adding the new term will not be singular and thus, will be invertible.

ii. Explain why the ridge regression estimator is likely to be more robust to issues of high variance compared with the least squares estimator.

Answer 4.b.ii.
In linear regression, we try to find the weights by reducing the error, but, it creates the problem of high variance as our model captures noise and outliers, thus overfitting. In order to control this, we add constraints to the weight coefficients. Ridge regression adds a regularization parameter which penalizes high coefficient values. It helps in shrinking the coefficients and thus decreases the complexity of the model and variance.

iii. How does the value of λ affect the bias and the variance of the estimator?

Answer 4.b.iii.
λ is a hyperparameter and its value is set using a cross validation set. It's value can vary a lot but practically we keep it between 0 and 1. Higher the value of λ, the more it will penalise the coefficients and reduce the impact of variables. It reduces the variance by adding more bias. Whereas if you reduce the value of λ, the bias will be reduced and variance will increase as the value of coefficients increases.

4.3 (b) - (i) **2 / 2**

✓ **- 0 pts Correct**

- 1 pts no/flawed argument for the invertibility of X^TX + lambda

- 2 pts wrong answer

- 2 pts no answer

- 1 pts error in calculation

as the matrix formed after adding the new term will not be singular and thus, will be invertible.

ii. Explain why the ridge regression estimator is likely to be more robust to issues of high variance compared with the least squares estimator.

Answer 4.b.ii.
In linear regression, we try to find the weights by reducing the error, but, it creates the problem of high variance as our model captures noise and outliers, thus overfitting. In order to control this, we add constraints to the weight coefficients. Ridge regression adds a regularization parameter which penalizes high coefficient values. It helps in shrinking the coefficients and thus decreases the complexity of the model and variance.

iii. How does the value of λ affect the bias and the variance of the estimator?

Answer 4.b.iii.
λ is a hyperparameter and its value is set using a cross validation set. It's value can vary a lot but practically we keep it between 0 and 1. Higher the value of λ, the more it will penalise the coefficients and reduce the impact of variables. It reduces the variance by adding more bias. Whereas if you reduce the value of λ, the bias will be reduced and variance will increase as the value of coefficients increases.

4.4 (b) - (ii) 2 / 2

✓ **- 0 pts Correct**

- 2 pts wrong answer

- 2 pts no answer

as the matrix formed after adding the new term will not be singular and thus, will be invertible.

ii. Explain why the ridge regression estimator is likely to be more robust to issues of high variance compared with the least squares estimator.

Answer 4.b.ii.
In linear regression, we try to find the weights by reducing the error, but, it creates the problem of high variance as our model captures noise and outliers, thus overfitting. In order to control this, we add constraints to the weight coefficients. Ridge regression adds a regularization parameter which penalizes high coefficient values. It helps in shrinking the coefficients and thus decreases the complexity of the model and variance.

iii. How does the value of λ affect the bias and the variance of the estimator?

Answer 4.b.iii.
λ is a hyperparameter and its value is set using a cross validation set. It's value can vary a lot but practically we keep it between 0 and 1. Higher the value of λ, the more it will penalise the coefficients and reduce the impact of variables. It reduces the variance by adding more bias. Whereas if you reduce the value of λ, the bias will be reduced and variance will increase as the value of coefficients increases.

4.5 (b) - (iii) **2 / 2**

✓ **- 0 pts Correct**

- 2 pts wrong answer

- 2 pts no answer

we do not keep the value of lambda between 0 and 1

1. Recall the definition of the risk of a hypothesis h for a regression problem with the mean squared error loss function.

   Answer 5a:

$$R(h) = \mathbb{E}[(y - h_D(x))^2]$$

$$R(h) = \frac{1}{n}\sum_{x=1}^{n}[(y_i - h_D(x_i))^2]$$

2. Let D' denote a dataset of size $n-1$. Show that

$$\mathop{\mathbb{E}}_{D\sim p}[\text{error}_{LOO}] = \mathop{\mathbb{E}}_{\substack{D'\sim p,\\(x,y)\sim p}}[(y - h_{D'}(x))^2]$$

   where the notation $D \sim p$ means that D is drawn i.i.d. from the distribution p and where h_D denotes the hypothesis returned by the learning algorithm trained on D. Explain how this shows that error_{LOO} is an (almost) unbiased estimator of the risk of h_D.

   Answer 5b:

$$\mathop{\mathbb{E}}_{D\sim p}[\text{error}_{LOO}] = \frac{1}{n}\sum_{i=1}^{n}\ell(i(x_i), y_i)$$

   Question 5c: Assuming that the time complexity of inverting a matrix of size $m \times m$ is in $\mathcal{O}\left(m^3\right)$, what is the complexity of computing the solution of linear regression on the dataset D?

   Answer 5c:

   For n observations and d weights, complexity of Matrix multiplication of $X^\top X$ is $O(d^2n)$ and the complexity for matrix inverse is O(d^3).

   Although for d<n, we can ignore $O(d^3)$ but total complexity is $O(d^2n)+$ $O(d^3)$

   Question 5d: Using $\mathbf{X}_{-i} \in \mathbb{R}^{(n-1)\times d}$ and $\mathbf{y}_{-i} \in \mathbb{R}^{(n-1)}$ to denote the data matrix and output vector obtained by removing the ith row of $\mathbf{X}$ and the ith entry of $\mathbf{y}$, write down a formula of the LOO-CV error for linear regression. What is the complexity of evaluating this formula?

   Answer 5d:

$$Error = \frac{1}{n}\sum_{i=1}^{n}(Y_i - (X_i)(X_i^\top X_i)^{-1}X_i^\top Y_i)$$

5.1 (a) **2 / 2**

✓ **- 0 pts** **Correct**

Correct

- 0 pts Correct

- 2 pts Not answer or Cannot find the answer.

- 1 pts Incomplete or not assessed for new data point.

- 0.5 pts Sign error or some steps omitted.

1. Recall the definition of the risk of a hypothesis h for a regression problem with the mean squared error loss function.

   Answer 5a:

$$R(h) = \mathbb{E}[(y - h_D(x))^2]$$

$$R(h) = \frac{1}{n}\sum_{x=1}^{n}[(y_i - h_D(x_i))^2]$$

2. Let D' denote a dataset of size $n-1$. Show that

$$\mathop{\mathbb{E}}_{D\sim p}[\text{error}_{LOO}] = \mathop{\mathbb{E}}_{\substack{D'\sim p,\\(x,y)\sim p}}[(y - h_{D'}(x))^2]$$

   where the notation $D \sim p$ means that D is drawn i.i.d. from the distribution p and where h_D denotes the hypothesis returned by the learning algorithm trained on D. Explain how this shows that error_{LOO} is an (almost) unbiased estimator of the risk of h_D.

   Answer 5b:

$$\mathop{\mathbb{E}}_{D\sim p}[\text{error}_{LOO}] = \frac{1}{n}\sum_{i=1}^{n}\ell(i(x_i), y_i)$$

   Question 5c: Assuming that the time complexity of inverting a matrix of size $m \times m$ is in $\mathcal{O}\left(m^3\right)$, what is the complexity of computing the solution of linear regression on the dataset D?

   Answer 5c:

   For n observations and d weights, complexity of Matrix multiplication of $X^\top X$ is $O(d^2n)$ and the complexity for matrix inverse is O(d^3).

   Although for d<n, we can ignore $O(d^3)$ but total complexity is $O(d^2n)+$ $O(d^3)$

   Question 5d: Using $\mathbf{X}_{-i} \in \mathbb{R}^{(n-1)\times d}$ and $\mathbf{y}_{-i} \in \mathbb{R}^{(n-1)}$ to denote the data matrix and output vector obtained by removing the ith row of $\mathbf{X}$ and the ith entry of $\mathbf{y}$, write down a formula of the LOO-CV error for linear regression. What is the complexity of evaluating this formula?

   Answer 5d:

$$Error = \frac{1}{n}\sum_{i=1}^{n}(Y_i - (X_i)(X_i^\top X_i)^{-1}X_i^\top Y_i)$$

5.2 (b) **0.5 / 2**

- **0 pts** Correct
- **2 pts** Incomplete or no answer found
- **1 pts** Error in algebra
- **0.5 pts** Minor Error in algebra or error in formula
- **1.5 pts** No proof
- **0 pts** Click here to replace this description.

✓ **- 1 pts No proof written or proof unclear**

✓ **- 0.5 pts unclear proof**

1. Recall the definition of the risk of a hypothesis h for a regression problem with the mean squared error loss function.

   Answer 5a:

$$R(h) = \mathbb{E}[(y - h_D(x))^2]$$

$$R(h) = \frac{1}{n}\sum_{x=1}^{n}[(y_i - h_D(x_i))^2]$$

2. Let D' denote a dataset of size $n-1$. Show that

$$\mathop{\mathbb{E}}_{D\sim p}[\text{error}_{LOO}] = \mathop{\mathbb{E}}_{\substack{D'\sim p,\\(x,y)\sim p}}[(y - h_{D'}(x))^2]$$

   where the notation $D \sim p$ means that D is drawn i.i.d. from the distribution p and where h_D denotes the hypothesis returned by the learning algorithm trained on D. Explain how this shows that error_{LOO} is an (almost) unbiased estimator of the risk of h_D.

   Answer 5b:

$$\mathop{\mathbb{E}}_{D\sim p}[\text{error}_{LOO}] = \frac{1}{n}\sum_{i=1}^{n}\ell(i(x_i), y_i)$$

   Question 5c: Assuming that the time complexity of inverting a matrix of size $m \times m$ is in $\mathcal{O}\left(m^3\right)$, what is the complexity of computing the solution of linear regression on the dataset D?

   Answer 5c:

   For n observations and d weights, complexity of Matrix multiplication of $X^\top X$ is $O(d^2n)$ and the complexity for matrix inverse is O(d^3).

   Although for d<n, we can ignore $O(d^3)$ but total complexity is $O(d^2n)+$ $O(d^3)$

   Question 5d: Using $\mathbf{X}_{-i} \in \mathbb{R}^{(n-1)\times d}$ and $\mathbf{y}_{-i} \in \mathbb{R}^{(n-1)}$ to denote the data matrix and output vector obtained by removing the ith row of $\mathbf{X}$ and the ith entry of $\mathbf{y}$, write down a formula of the LOO-CV error for linear regression. What is the complexity of evaluating this formula?

   Answer 5d:

$$Error = \frac{1}{n}\sum_{i=1}^{n}(Y_i - (X_i)(X_i^\top X_i)^{-1}X_i^\top Y_i)$$

5.3 (c) **2 / 2**

✓ **- 0 pts** **Correct**

- 2 pts Incorrect or answer not found.

- 1.5 pts Marks for formula

- 1 pts Marks for formula

- 0.25 pts Minor calculation error.

1. Recall the definition of the risk of a hypothesis h for a regression problem with the mean squared error loss function.

   Answer 5a:

$$R(h) = \mathbb{E}[(y - h_D(x))^2]$$

$$R(h) = \frac{1}{n}\sum_{x=1}^{n}[(y_i - h_D(x_i))^2]$$

2. Let D' denote a dataset of size $n-1$. Show that

$$\mathop{\mathbb{E}}_{D\sim p}[\text{error}_{LOO}] = \mathop{\mathbb{E}}_{\substack{D'\sim p,\\(x,y)\sim p}}[(y - h_{D'}(x))^2]$$

   where the notation $D \sim p$ means that D is drawn i.i.d. from the distribution p and where h_D denotes the hypothesis returned by the learning algorithm trained on D. Explain how this shows that error_{LOO} is an (almost) unbiased estimator of the risk of h_D.

   Answer 5b:

$$\mathop{\mathbb{E}}_{D\sim p}[\text{error}_{LOO}] = \frac{1}{n}\sum_{i=1}^{n}\ell(i(x_i), y_i)$$

   Question 5c: Assuming that the time complexity of inverting a matrix of size $m \times m$ is in $\mathcal{O}\left(m^3\right)$, what is the complexity of computing the solution of linear regression on the dataset D?

   Answer 5c:

   For n observations and d weights, complexity of Matrix multiplication of $X^\top X$ is $O(d^2n)$ and the complexity for matrix inverse is O(d^3).

   Although for d<n, we can ignore $O(d^3)$ but total complexity is $O(d^2n)+$ $O(d^3)$

   Question 5d: Using $\mathbf{X}_{-i} \in \mathbb{R}^{(n-1)\times d}$ and $\mathbf{y}_{-i} \in \mathbb{R}^{(n-1)}$ to denote the data matrix and output vector obtained by removing the ith row of $\mathbf{X}$ and the ith entry of $\mathbf{y}$, write down a formula of the LOO-CV error for linear regression. What is the complexity of evaluating this formula?

   Answer 5d:

$$Error = \frac{1}{n}\sum_{i=1}^{n}(Y_i - (X_i)(X_i^\top X_i)^{-1}X_i^\top Y_i)$$

As we can see, the same complexity of linear regression can be used and be computed n times.

So total complexity is $O(d^2n^2) + O(d^3n)$

Question 5e:

It turns out that for the special case of linear regression, the leave-one-out error can be computed more efficiently. Show that in the case of linear regression we have

$$\text{error}_{LOO} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \mathbf{w}^{*\top}\mathbf{x}_i}{1 - \mathbf{x}_i^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_i}\right)^2$$

where $\mathbf{w}^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$ is the solution of linear regression computed on the whole dataset D. What is the complexity of evaluating this formula?

Answer 5e: The complexity using this formula can be reduced compared to our previous complexity as we can compute $(X^\top X)^{-1}$ only once and then store it in the memory. Then all we have in the numerator is matrix vector multiplication of $O(d^2)$

So total complexity of this formula is the same as for linear regression i.e. $O(d^2n) + O(d^3)$

5.4 (d) **2 / 2**

✓ **- 0 pts** **Correct**

- 1 pts Marks for formula

- 0.25 pts Minor calculation errors

- 2 pts Incorrect or answer not found

As we can see, the same complexity of linear regression can be used and be computed n times.

So total complexity is $O(d^2n^2) + O(d^3n)$

Question 5e:

It turns out that for the special case of linear regression, the leave-one-out error can be computed more efficiently. Show that in the case of linear regression we have

$$\text{error}_{LOO} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \mathbf{w}^{*\top}\mathbf{x}_i}{1 - \mathbf{x}_i^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_i}\right)^2$$

where $\mathbf{w}^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$ is the solution of linear regression computed on the whole dataset D. What is the complexity of evaluating this formula?

Answer 5e: The complexity using this formula can be reduced compared to our previous complexity as we can compute $(X^\top X)^{-1}$ only once and then store it in the memory. Then all we have in the numerator is matrix vector multiplication of $O(d^2)$

So total complexity of this formula is the same as for linear regression i.e. $O(d^2n) + O(d^3)$

5.5 (e) **1 / 2**

- **0 pts** Correct
- **2 pts** Incorrect or no answer found.
- **0.25 pts** Minor error in calculation
- **1 pts** Marks for formula
- **1 pts** incomplete
- **0.5 pts** Error in calculation
- **0.75 pts** No computational complexity

✓ **- 1 pts** **No proof**

1. **Multivariate Regression** [10 points]

We consider the problem of learning a vector-valued function $f : \mathbb{R}^d \to \mathbb{R}^p$ from input-output training data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ where each $\mathbf{x}_i$ is a d-dimensional vector and each $\mathbf{y}_i$ is a p-dimensional vector. We choose our hypothesis class to be the set of linear functions from $\mathbb{R}^d$ to $\mathbb{R}^p$, that is function satisfying $f(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}$ for some $d \times p$ regression matrix $\mathbf{W}$, and we want to minimize the squared error loss function

$$J(\mathbf{W}) = \sum_{i=1}^{n} \|\mathbf{W}^\top \mathbf{x}_i - \mathbf{y}_i\|_2^2 \tag{1}$$

over the training data.

Let $\mathbf{W}^*$ be the minimizer of the empirical risk:

$$\mathbf{W}^* = \arg\min_{\mathbf{W} \in \mathbb{R}^{d \times p}} J(\mathbf{W}).$$

(a) Derive a closed-form solution for $\mathbf{W}^*$ as a function of the data matrices $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $\mathbf{Y} \in \mathbb{R}^{n \times p}$.
(hint: once you have expressed $J(\mathbf{W})$ as a function of $\mathbf{X}$ and $\mathbf{Y}$, you may find the matrix cookbook useful to compute gradients w.r.t. to the matrix $\mathbf{W}$)
Answer 6.a.
$J(W)$ can be expressed as a function of X and Y by:

$$\begin{aligned}
J(W) &= \|XW - Y\|_2^2 \\
&= Tr((XW - Y)^T (XW - Y)) \\
&= Tr((W^T X^T - Y^T)(XW - Y)) \\
&= Tr(W^T X^T XW - W^T X^T Y - Y^T XW + Y^T Y) \\
&= Tr(W^T X^T XW) - Tr(W^T X^T Y) - Tr(Y^T XW) + Tr(Y^T Y)
\end{aligned}$$

The gradient of J with respect to W is given by,

$$\begin{aligned}
\nabla_W J(W) &= \frac{d}{dW}(Tr(W^T X^T XW) - Tr(W^T X^T Y) - Tr(Y^T XW) + Tr(Y^T Y)) \\
&= \frac{d}{dW}Tr(W^T X^T XW) - \frac{d}{dW}Tr(W^T X^T Y) - \frac{d}{dW}Tr(Y^T XW) + \frac{d}{dW}Tr(Y^T Y)) \\
&= W^T X^T \frac{d}{dW}XW + (XW)^T \frac{d}{dW}W^T X^T - (X^T Y)^T - Y^T X + 0 \\
&= W^T X^T X + W^T X^T X - Y^T X - Y^T X \\
&= 2(W^T X^T X - Y^T X)
\end{aligned}$$

$\nabla_W J(W^*)$ will be zero,

$$\begin{aligned}
\nabla_W J(W^*) = 2((W^*)^T X^T X - Y^T X) &= 0 \\
(W^*)^T X^T X - Y^T X &= 0 \\
X^T XW^* - X^T Y &= 0 \\
X^T XW^* &= X^T Y \\
W^* &= (X^T X)^{-1} X^T Y
\end{aligned}$$

(b) Show that solving the problem from the previous question is equivalent to independently solving p independent classical linear regression problems (one for each component of the output vector), and give an example of a multivariate regression task where performing **independent** regressions for each output variables is not the best thing to do.

Answer 6.b.
Let $J^i(w^i)$ be the squared error of the i^{th} independent regressor, where w^i is a $(d, 1)$ dimensional vector. J^i is given by:

$$J^i(w^i) = \|w^i X - Y^i\|_2^2$$

The gradient of $J^i(w^i)$ with respect to w^i is

6.1 (a) **2.5 / 3**

- **0 pts** Correct
- **3 pts** no answer
- **2.5 pts** major mistakes in derivation

✓ **- 0.5 pts minor mistake**

- **2 pts** wrong derivation
- **1.5 pts** the trace is missing in the expression of J(W)
- **1.5 pts** gradient derivation is wrong / not clear
- **1 pts** incomplete
- **3 pts** wrong
- **1 pts** mistake in the derivation of the gradient
- **1 pts** two answers are given, only one of them is correct
- **1 pts** dimensions do not match

The gradient of J with respect to W is given by,

$$\begin{aligned}
\nabla_W J(W) &= \frac{d}{dW}(Tr(W^T X^T XW) - Tr(W^T X^T Y) - Tr(Y^T XW) + Tr(Y^T Y)) \\
&= \frac{d}{dW}Tr(W^T X^T XW) - \frac{d}{dW}Tr(W^T X^T Y) - \frac{d}{dW}Tr(Y^T XW) + \frac{d}{dW}Tr(Y^T Y)) \\
&= W^T X^T \frac{d}{dW}XW + (XW)^T \frac{d}{dW}W^T X^T - (X^T Y)^T - Y^T X + 0 \\
&= W^T X^T X + W^T X^T X - Y^T X - Y^T X \\
&= 2(W^T X^T X - Y^T X)
\end{aligned}$$

$\nabla_W J(W^*)$ will be zero,

$$\begin{aligned}
\nabla_W J(W^*) = 2((W^*)^T X^T X - Y^T X) &= 0 \\
(W^*)^T X^T X - Y^T X &= 0 \\
X^T XW^* - X^T Y &= 0 \\
X^T XW^* &= X^T Y \\
W^* &= (X^T X)^{-1} X^T Y
\end{aligned}$$

(b) Show that solving the problem from the previous question is equivalent to independently solving p independent classical linear regression problems (one for each component of the output vector), and give an example of a multivariate regression task where performing **independent** regressions for each output variables is not the best thing to do.

Answer 6.b.
Let $J^i(w^i)$ be the squared error of the i^{th} independent regressor, where w^i is a $(d, 1)$ dimensional vector. J^i is given by:

$$J^i(w^i) = \|w^i X - Y^i\|_2^2$$

The gradient of $J^i(w^i)$ with respect to w^i is

$$
\begin{aligned}
\nabla_{w^i} J^i(w^i) &= \frac{d}{dw^i}\|Xw^i - Y^i\|_2^2 \\
&= \frac{d}{dw^i}((Xw^i - Y^i)^T(Xw^i - Y^i)) \\
&= \frac{d}{dw^i}((w^{iT}X^T - Y^T)(Xw^i - Y^i)) \\
&= \frac{d}{dw^i}(w^{iT}X^TXw^i - w^{iT}X^TY^i - Y^TXw^i + Y^{iT}Y^i) \\
&= 2(w^{iT}X^TX - Y^{iT}X)
\end{aligned}
$$

$\nabla_{w^i} J^i$ will be 0 at $w^i = w^{*i}$,

$$
\begin{aligned}
\nabla^i_w J^i(w^{i*}) = 2(w^{iT}X^TX - Y^{iT}X) &= 0 \\
X^TXw^{i*} &= X^TY^i \\
w^{i*} &= (X^TX)^{-1}X^TY^i
\end{aligned}
$$

w^{i*} is a d-dimensional row vector. Since the matrix formed by stacking p w^{i*} vectors as column vectors is the same as the matrix W^* derived in question 6.a., we can conclude that performing independent regressions for each component of the output vector is the same as performing one multivariate regression on the output vector.

Doing this is not the best thing for the tasks in which the output variables are linearly dependent, for example, when trying to predict the price and size of houses. We should use low rank regression to capture the relationship between them.

(c) The low rank regression algorithm addresses the issue described in the previous question by imposing a low rank constraint on the regression matrix $\mathbf{W}$. Intuitively, the low rank constraint encourages the model to capture linear dependencies in the components of the output vector.

Propose an algorithm to minimize the squared error loss over the training data subject to a low rank constraint on the regression matrix $\mathbf{W}$:

$$
\min_{\mathbf{W}\in\mathbb{R}^{d\times p}} J(\mathbf{W}) \qquad \text{s.t. } \operatorname{rank}(\mathbf{W}) \leq R.
$$

(hint: There are different ways to do that. You could for example leverage the fact that $\operatorname{rank}(\mathbf{W}) \leq R$ *if and only if there exists*

6.2 (b) 3 / 3

✓ - 0 pts Correct

- 3 pts no answer
- 1 pts incomplete answer
- 1 pts this is not a classification problem
- 2 pts incomplete answer
- 0.5 pts correct but not enough justification
- 1 pts small mistake
- 3 pts incorrect
- 3 pts No proof

$\mathbf{A} \in \mathbb{R}^{d \times R}$ *and* $\mathbf{B} \in \mathbb{R}^{R \times p}$ *such that* $\mathbf{W} = \mathbf{AB}$.*)*

Answer 6.c.
To constraint the rank of W, we can perform singular value decomposition (SVD) on W such that,

$$W = USV^T,$$

where U is $d \times R$, S is $R \times R$ diagonal matrix, and V is $p \times R$. The cost function J is given by,

$$\begin{aligned} J(W) &= \|XW - Y\|_2^2 \\ &= \|XUSV^T - Y\|_2^2 \end{aligned}$$

The above cost function can be minimized by a closed-form solution or iteratively using gradient descent. In either case, the gradient of the decomposition with respect to W can be computed by:

$$dW = \frac{dU}{dW} SV^T + U \frac{dS}{dW} V^T + US \frac{dV^T}{dW}$$

The bottom rows of S can also be clipped to reduce the rank of W without losing much information about the weights.

6.3 (c) **1 / 4**

- **0 pts** Correct
- **4 pts** Wrong / incomplete
- **4 pts** no answer
- **4 pts** major errors in derivations
- **0.5 pts** very good! just a few steps missing
- **1 pts** right idea but the derivations are wrong / incomplete

✓ **- 3 pts Wrong derivations / incomplete**

- **0 pts** Not the optimal solution but a correct answer
- **2 pts** good direction but not enough details

Question 1 What is the derivative of the regularization term

$$\frac{1}{2}\sum_{j'=1}^{m}\|\mathbf{w}^{j'}\|_2^2$$

with respect to w_k^j (the kth weight of the weight vector for the jth class)? Show all your work and write your answer in the report.

Answer:

$$Reg = \frac{1}{2}\sum_{j'=1}^{m}\|\mathbf{w}^{j'}\|_2^2$$

$$Reg = \frac{1}{2}\sum_{j'=1}^{m}(\sqrt{(w_1^{j'})^2 + (w_2^{j'})^2 + ..(w_p^{j'})^2})^2$$

$$Reg = \frac{1}{2}\sum_{j'=1}^{m}\sum_{k'=1}^{p} w_k^{j'2}$$

$$\frac{\partial Reg}{\partial w_k^j} = \frac{\partial \frac{1}{2}\sum_{j'=1}^{m}\sum_{k'=1}^{p}(w_k^{j'})^2}{\partial w_k^j}$$

Since gradient is w.r.t kth weight for jth class, the gradient for all other weights w.r.t w_k^j goes to zero.

$$\frac{\partial Reg}{\partial w_k^j} = w_k^j$$

7.1 Derivative Regularization **5 / 5**

✓ **- 0 pts** **Correct**

Answer 2:

$$Loss(w) = \frac{C}{n} \sum_{(\mathbf{x}_i, y_i) \in S} \sum_{j'=1}^{m} l(\mathbf{w}^{j'}; (\mathbf{x}_i, y_i))^2$$

Using the squared hinge loss in the function

$$Loss(w) = \frac{C}{n} \sum_{(\mathbf{x}_i, y_i) \in S} \sum_{j'=1}^{m} (\max\{0, 1 - (\langle \mathbf{w}^{j'}, \mathbf{x}_i \rangle) 1\{y_i = j'\}\})^2$$

where,

$$1\{y_i = j'\} = \begin{cases} 1 & \text{if } y_i = j' \\ -1 & \text{if } y_i \neq j' \end{cases}$$

Using

$$\frac{\partial}{\partial a} \max\{0, a\} = \begin{cases} 1 & \text{if } a > 0 \\ 0 & \text{if } a \leq 0 \end{cases}$$

$$\frac{\partial Loss(w)}{\partial w_k^j} = \frac{2C}{n} \sum_{(x_i, y_i) \in S} \sum_{j'=1}^{m} \max\{0, 1 - (\langle \mathbf{w}^{j'}, \mathbf{x}_i \rangle) 1\{y_i = j'\}\} \frac{\partial max(1 - \langle \mathbf{w}^{j'}, \mathbf{x}_i \rangle 1\{y_i = j'\})}{\partial w_k^j}$$

if $(\langle \mathbf{w_k}^{j'}, \mathbf{x}_{i,k} \rangle 1\{y_i = j'\} < 1)$

$\frac{\partial Loss(w)}{\partial w_k^j} = 0$ *if* $(\langle \mathbf{w}_k^{j'}, \mathbf{x}_{i,k} \rangle 1\{y_i = j'\} \geq 1)$

Since the gradient is computed w.r.t w_k^j, all other terms will be zero and hence the summation of all classes will be removed and we'll be left with $-x_{i,k} 1\{y_i = j'\}$.

$$\frac{\partial Loss(w)}{\partial w_k^j} = \frac{2C}{n} \sum_{(x_i, y_i) \in S} \max\{0, 1 - (\langle \mathbf{w}_k^j, \mathbf{x}_{i,k} \rangle) 1\{y_i = j\}\} (-\mathbf{x}_{i,k}) 1\{y_i = j'\}$$

if $(\langle \mathbf{w_k}^j, \mathbf{x}_{i,k} \rangle 1\{y_i = j\} < 1)$, where $1\{y_i = j'\} = \begin{cases} 1 & \text{if } y_i = j \\ -1 & \text{if } y_i \neq j \end{cases}$

$\frac{\partial Loss(w)}{\partial w_k^j} = 0$ *if* $(\langle \mathbf{w}^j, \mathbf{x}_{i,k} \rangle 1\{y_i = j\} \geq 1)$

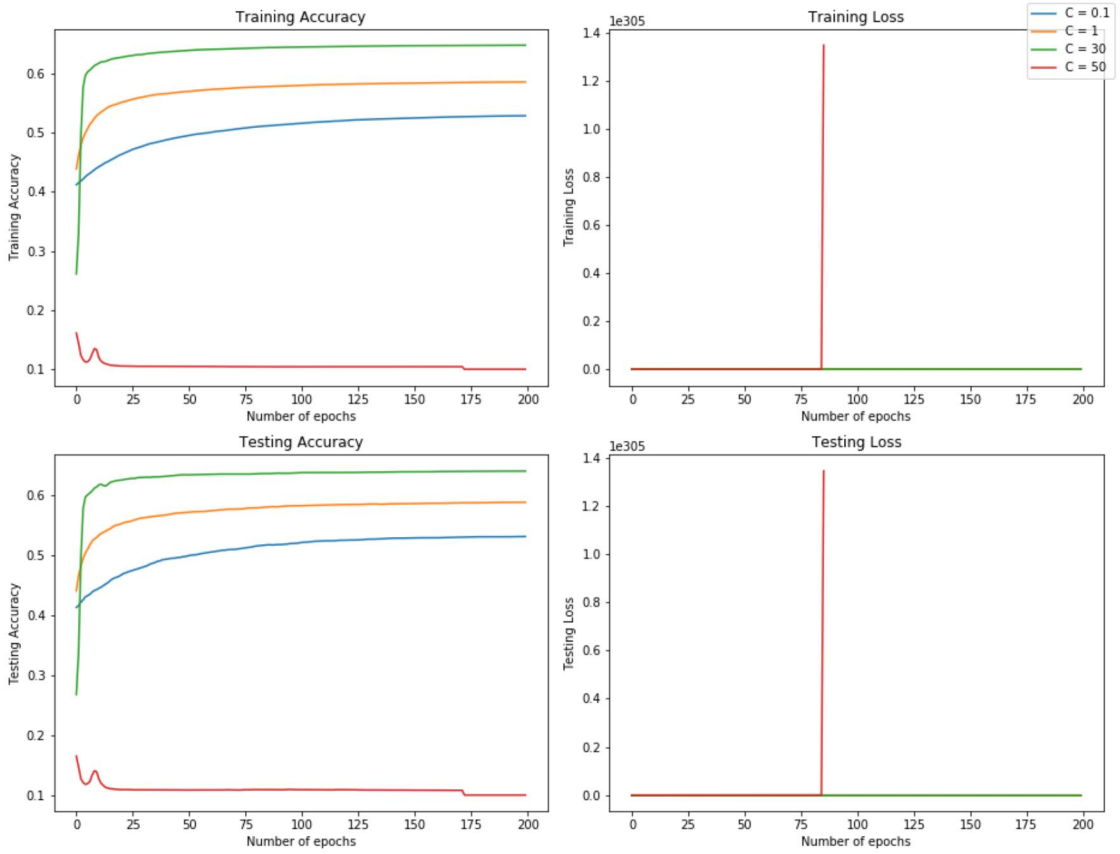
7.2 Derivative of Hinge Loss 10 / 10

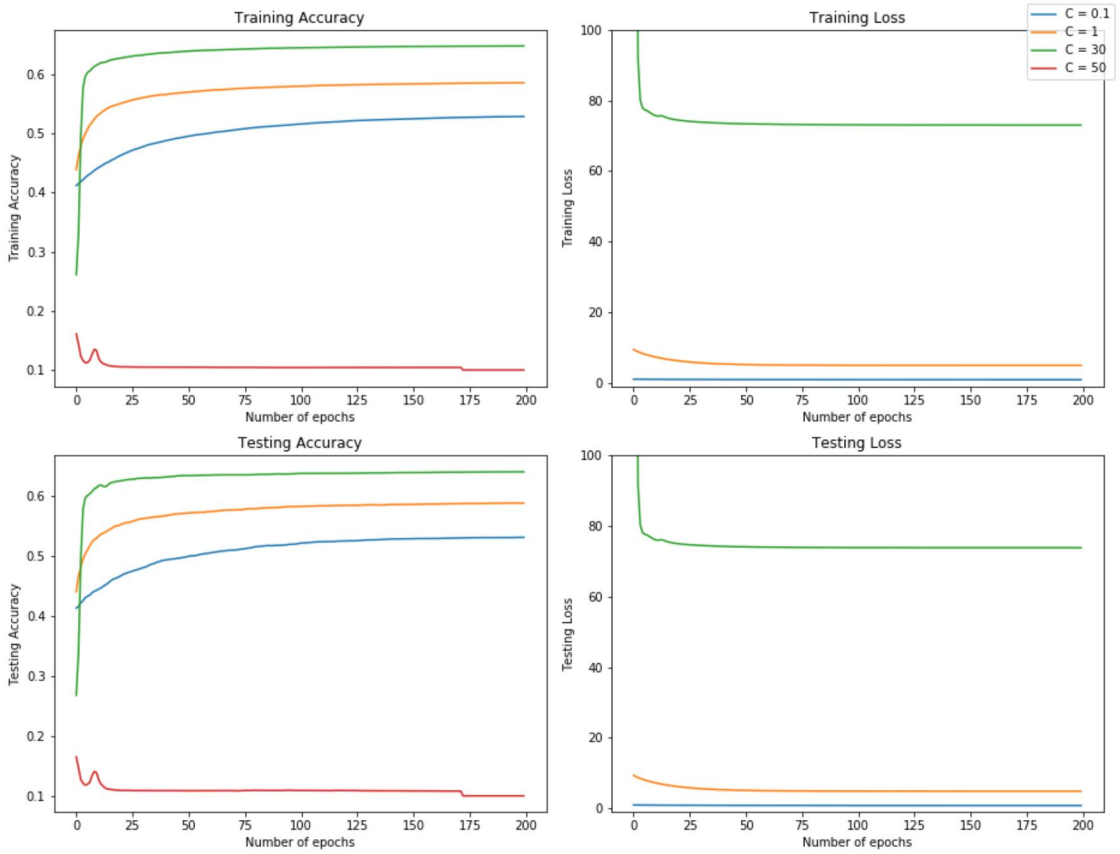
✓ **- 0 pts Correct**

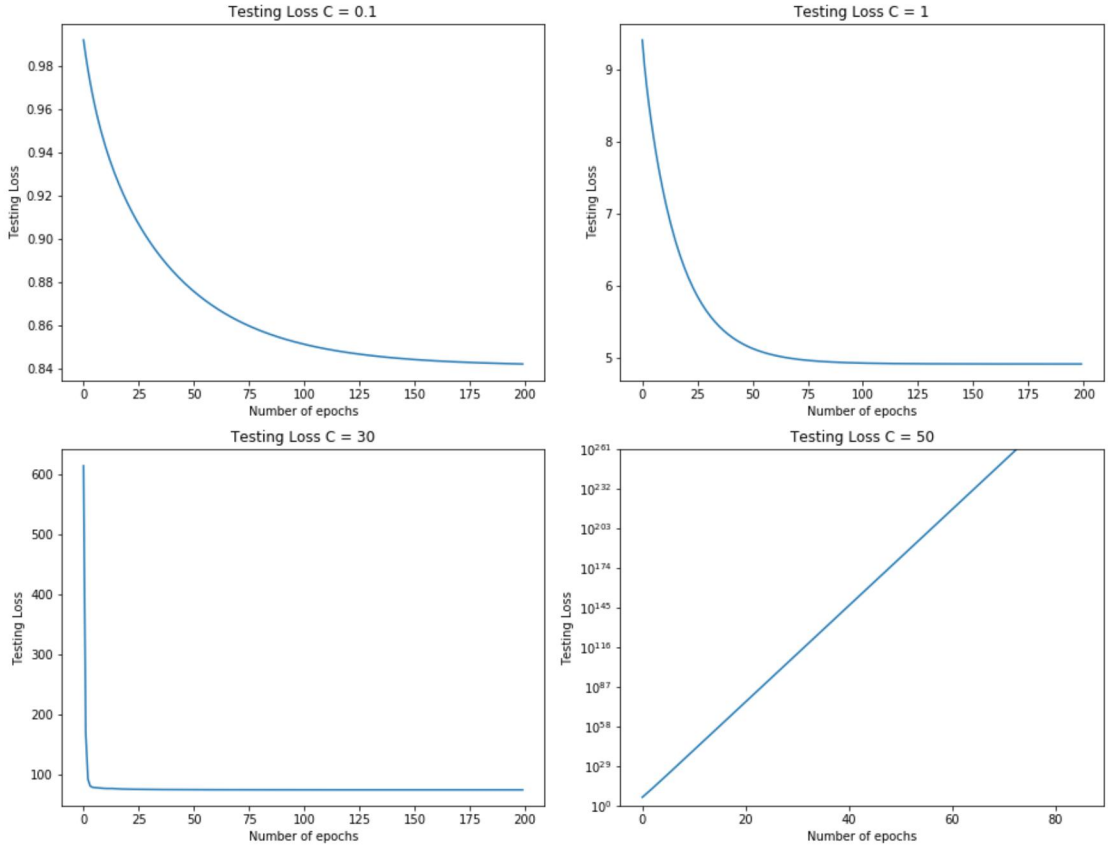
Missing a minus sign (EDIT: the minus sign is there after all)

Answer 3

Since the losses for C=50 are approaching infinity, I have also drawn plots by limiting y in the loss and individual plots for losses on the next 2 pages for more clarity.




Training Accuracy
Training Accuracy
Number of epochs
Training Loss
Training Loss
Number of epochs
C = 0.1
C = 1
C = 30
C = 50
Testing Accuracy
Testing Accuracy
Number of epochs
Testing Loss
Testing Loss
Number of epochs


Testing Loss C = 0.1
Testing Loss
Number of epochs
Testing Loss C = 1
Testing Loss
Number of epochs
Testing Loss C = 30
Testing Loss
Number of epochs
Testing Loss C = 50
Testing Loss
Number of epochs

7.3 SVM plots loss/accuracy vs epoch **5 / 5**

✓ **- 0 pts** **Correct**