Homework 1

Caitlin Tang 505683631

Due: Monday 1/20/2025

1. Linear Algebra Refresher (25 points)

(a) (a) (12 points) Let $\mathbf{Q}$ be a real orthogonal matrix.

i. (3 points) Show that $\mathbf{Q^T}$ and $\mathbf{Q^{-1}}$ are also orthogonal.

ii. (3 points) Show that $\mathbf{Q}$ has eigenvalues with norm 1.

iii. (3 points) Show that the determinant of $\mathbf{Q}$ is either +1 or -1.

iv. (3 points) Show that $\mathbf{Q}$ defines a length preserving transformation.

Q is defined as a real orthogonal matrix → a nxn matrix A with real values and the transpose A^T is equal to the inverse (A^{-1})

∴ if $A^T = A^{-1}$ then $A^TA = I$ (identity matrix)

Part (a) (i) Show that Q^T and Q^{-1} are also orthogonal

Matrix Q is orthogonal if $Q^TQ = I$

Showing Q^T is orthogonal

Assume $Q^TQ = I$ → take transpose

$(Q^TQ)^T = I^T$

$(Q^T)(Q^T)^T = I$ → use $(AB)^T = B^TA$ and $I^T = I$

$Q^TQ = I$ → $(Q^T)^T = Q$

Alternatively, given property $A^TA = I$, set $A = Q^T$

$A^TA = I$ → substitute $A = Q^T$

$(Q^T)^TQ^T = I$

$QQ^T = I$ → take the transpose

$(QQ^T)^T = I^T$ → Use the property $(AB)^T = B^TA^T$ and $I^T = I$

$Q^TQ = I$

Therefore, Q^T satisfies the condition $A^TA = I$ and is orthogonal

Showing Q^{-1} is orthogonal

We know that $Q^T = Q^{-1}$ and $QQ^{-1} = I$

$A^TA = I$ → substitute $A = Q^T$

$(Q^T)^TQ^T = I$

$QQ^T = I$ → substitute $Q^T = Q^{-1}$

$(Q^{-1})(Q^{-1})^T = I$

$[(Q^{-1})(Q^{-1})^T] = I^T$ → Apply the $(AB)^T = B^TA^T$ property and $I^T = I$

$(Q^{-1})^T(Q^{-1}) = I$

Since Q^{-1} is as $(Q^{-1})^T(Q^{-1}) = I$ which is of the form $A^TA = I$ then Q^{-1} is also orthogonal

Therefore Q^T is orthogonal and Q^{-1} is orthogonal

Part (a) (ii) Show that Q has eigenvalues with norm 1

Review: The norm of a matrix is a real number which is a measure of the magnitude of the matrix. The following properties hold true for defining the norm of a square matrix A (norm is a non-negative real number denoted $\|A\|$)

1. $\|A\| \geq 0$ for any square matrix A
2. $\|A\| = 0$ iff matrix $= 0$
3. $\|kA\| = |k| \, \|A\|$, for any scalar k
4. $\|A + B\| \leq \|A\| + \|B\|$
5. $\|AB\| \leq \|A\| \, \|B\|$

For the real orthogonal matrix Q ...
if λ is an eigenvalue of Q with its corresponding vector $\vec{x}$ then

$$Q\vec{x} = \lambda\vec{x} \quad \text{where } \vec{x} \neq 0$$

since Q is orthogonal $Q^TQ = I$ and $Q^T = Q^{-1}$

The Euclidean norm of $\vec{x} \in \mathbb{R}^n$ is defined as $\|\vec{x}\|_2 = \sqrt{x^Tx}$
When our orthogonal matrix Q is added it transforms the vector to become $Q\vec{x}$

The Euclidean norm of $Q\vec{x}$ is thus $\|Q\vec{x}\| = \sqrt{(Q\vec{x})^T(Q\vec{x})}$

$(Q\vec{x})^T = \vec{x}^TQ^T$ from the property $(AB)^T = B^TA^T$

$Q^TQ = I$

$$\|Q\vec{x}\| = \sqrt{(\vec{x}^T)(Q^T)(Q)(\vec{x})}$$

$$\|Q\vec{x}\| = \sqrt{(\vec{x})^TI(\vec{x})}$$

$$\|Q\vec{x}\| = \sqrt{\vec{x}^T\vec{x}} = \|\vec{x}\|$$

we know that $\|Q\vec{x}\| = \|\vec{x}\|$ $\therefore$ $\|Q\vec{x}\| = \|\lambda\vec{x}\|$

$\|\lambda\vec{x}\| = \|\vec{x}\|$

$|\lambda| \, \|\vec{x}\| = \|\vec{x}\|$

Separate based on the scalar property $\|kA\| = |k| \, \|A\|$ for some scalar k

since $\vec{x} \neq 0$, $\|\vec{x}\| \neq 0$ and $|\lambda| = 1$. Therefore, the eigenvalues of Q will have a magnitude of 1

NOTE: Eigenvalues are scalars so we can alternatively prove it using the property $\|kA\| = |k| \, \|A\|$

Assume we have a vector $\vec{x}$

$(\lambda\vec{x})^T(\lambda\vec{x}) = (Q\vec{x})^T(Q\vec{x})$ Apply the property $(AB)^T = B^TA^T$

$\vec{x}^T\lambda^T\lambda\vec{x} = \vec{x}^TQ^TQ\vec{x}$

NOTE $\lambda^T = \lambda$ Therefore $\lambda^T\lambda = \lambda \cdot \lambda = \lambda^2$

$\lambda^2\vec{x}^T\lambda^2\vec{x} = \vec{x}^TI\vec{x}$

move the λ^2 because it's the scalar

$\lambda^2\vec{x}^T\vec{x} = \vec{x}^TI\vec{x}$

$\lambda^2 = 1$

$\lambda = \pm 1$

↳ this proves eigen value $= \pm 1$ and norm of $|\pm 1| = 1$ (magnitude (absolute value))

(a)(iii) Show that the determinant of Q is either +1 or -1

Review: Properties of Determinants

1. Determinant of Identity Matrix $\det(I) = 1$
2. Determinant of Transposed Matrix A $\det(A^T) = \det(A)$
3. Determinant of Product of Two Matrices $\det(AB) = \det(A)\cdot\det(B)$
4. Determinant of Inverse: $\det(A^{-1}) = \frac{1}{\det(A)}$
5. If you exchange 2 rows of a matrix, you reverse the sign of the determinant
6. If you multiply a row or column by a scalar k, determinant also scaled

NOTE: determinant of a matrix is product of its eigenvalues $\det(A) = \lambda_1 \cdot \lambda_2 \cdots \lambda_n$

We want to show $\det(Q) \in \{+1, -1\}$

For an orthogonal matrix Q, we know that $Q^TQ = I$

$\det(Q^TQ) = \det(I)$ → use property $\det(I) = 1$

$\det(Q^TQ) = 1$

$\det(Q^T)\det(Q) = 1$ → use property $\det(AB) = \det(A)\det(B)$

$\det(Q)\cdot\det(Q) = 1$ → use property $\det(Q^T) = \det(Q)$

$\sqrt{(\det(Q))^2} = 1$ → take the square root

$\det(Q) = \pm 1$

Therefore, for the real, orthogonal matrix Q has a determinant of 1 or -1

(a)(iv) Show that Q defines a length preserving transformation → show that $\|Qv\| = \|v\|$

For Q, we know that

$$Q^TQ = I$$

For any vector $\vec{v}$, the transformation $Q\vec{v}$ should preserve the Euclidean length of $\vec{v}$

The Euclidean norm of v is $\|\vec{v}\| = \sqrt{\vec{v}^T\vec{v}}$

The Euclidean norm of transformed vector $Q\vec{v}$ is $\|Q\vec{v}\| = \sqrt{(Q\vec{v})^T(Q\vec{v})}$

$\|Q\vec{v}\| = \sqrt{(Q\vec{v})^T(Q\vec{v})}$ → Apply the property $(AB)^T = B^TA^T$

$\|Q\vec{v}\| = \sqrt{\vec{v}^TQ^TQ\vec{v}}$

$\|Q\vec{v}\| = \sqrt{\vec{v}^TI\vec{v}}$ → $Q^TQ = I$

$\|Q\vec{v}\| = \sqrt{\vec{v}^T\vec{v}} = \|\vec{v}\|$

Therefore the transformed vector $Q\vec{v}$ has a Euclidean norm $\|Q\vec{v}\|$ equal to $\|\vec{v}\|$

Problem #1 Part b

(b) (8 points) Let $\mathbf{A}$ be a matrix.

i. (4 points) What is the relationship between the singular vectors of $\mathbf{A}$ and the eigenvectors of $\mathbf{A}\mathbf{A}^T$? What about $\mathbf{A}^T\mathbf{A}$?

ii. (4 points) What is the relationship between the singular values of $\mathbf{A}$ and the eigenvalues of $\mathbf{A}\mathbf{A}^T$? What about $\mathbf{A}^T\mathbf{A}$?

(b)(i) What is the relationship between the singular vectors of A and the eigenvectors of AA^T? What about A^TA?

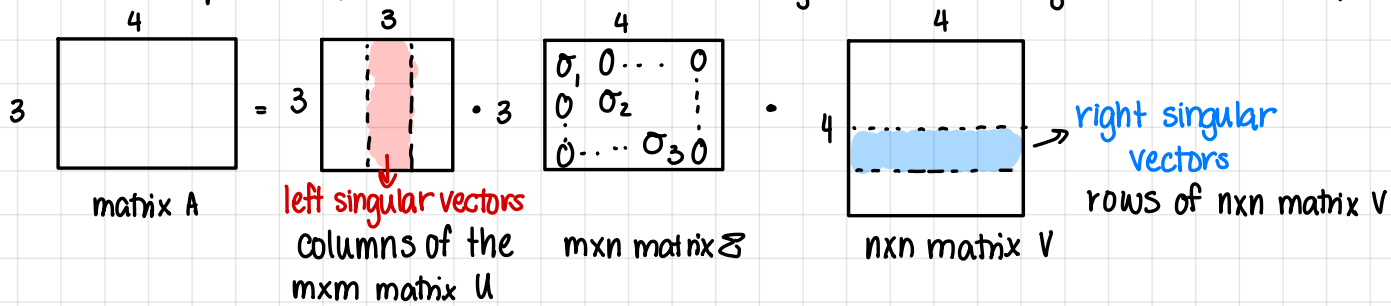
For this problem, we assume that A is a mxn dimensional matrix

1. AA^T will be a square symmetric matrix size mxm
2. A^TA will be a square symmetric matrix size nxn

Singular Value Decomposition (SVD) Review

The SVD of mxn matrix A is $A = U\Sigma V^T$ → U = mxm orthogonal matrix of left singular vectors of A
→ V = nxn orthogonal matrix of right singular vectors of A
→ Σ = mxn diagonal matrix of singular values of A $(\sigma_1, \ldots, \sigma_m)$

Visual example m=3, n=4



Solving for AA^T

$$AA^T = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma \underbrace{V^T V}_{I_n} \Sigma^T U^T = U\Sigma I_n \Sigma^T U^T = U\Sigma\Sigma^T U^T = U\Sigma^2 U^T$$

(mxm) (mxm)

NOTE: $\Sigma\Sigma^T \neq \Sigma^T\Sigma$
(mxn)(nxm) (nxm)(mxn)

$$\Rightarrow \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \sigma_m^2 \end{pmatrix} = \Sigma^2$$

Therefore, AA^T has eigenvectors given by the singular left vectors of A (columns of U) and its corresponding eigenvalues are the squares of the singular values of A $(\sigma_1^2, \ldots, \sigma_m^2)$

Solving for A^TA

$$A^TA = (U\Sigma V^T)^T(U\Sigma V^T) = V\Sigma^T \underbrace{U^TU}_{I_m} \Sigma V^T = V\Sigma^T I_m \Sigma V^T = V\Sigma^T\Sigma V^T = V\Sigma^2 V^T$$

NOTE: $\Sigma\Sigma^T = \Sigma^T\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_m^2 \end{pmatrix} = \Sigma^2$ (nxn matrix)

Therefore A^TA has eigenvectors given by the right singular vectors of A (rows of V)

Summary

1. AA^T and A^TA share the same nonzero eigenvalues $(\sigma_1, \ldots, \sigma_i^2)$
2. The eigenvectors of AA^T are the left singular vectors of A
3. The eigenvectors of A^TA are the right singular vectors of A

(b) (ii) What is the relationship between the singular values of A and the eigenvalues of AA^T? What about A^TA?

Again using the Singular Value Decomposition (SVD)
where $A = U\Sigma V^T$ → U = m×m orthogonal matrix of left singular vectors of A
→ V = n×n orthogonal matrix of right singular vectors of A
→ Σ = m×n diagonal matrix of singular values of A $(\sigma_1, \ldots, \sigma_m)$

Solving AA^T

$$AA^T = (U\Sigma V^T)(U\Sigma V^T)^T$$
$$= U\Sigma V^T V\Sigma^T U$$
$$= U\Sigma I_m \Sigma^T U$$
$$= U\Sigma^2 U$$

Solving A^TA

$$A^TA = (U\Sigma V^T)^T(U\Sigma V^T)$$
$$= V\Sigma^T U^T U\Sigma V^T$$
$$= V\Sigma^T I_n \Sigma V^T$$
$$= V\Sigma^2 V^T$$

in both cases, Σ^2 is a diagonal matrix with σ_i^2 on the diagonal
→ with AA^T, $\Sigma\Sigma^T$ is a m×m matrix with σ_i^2 on the diagonal
→ with A^TA, $\Sigma^T\Sigma$ is a n×n matrix with σ_i^2 on the diagonal

The eigenvalues of both AA^T and A^TA are the squares of the singular values of A. Therefore, the singular values of A $(\sigma_1, \ldots, \sigma_i)$ are the square roots of the eigenvalues of AA^T and A^TA

Problem #1 Part (c) TRUE / FALSE

(c) (5 points) True or False. Partial credit on an incorrect solution may be awarded if you justify your answer.

i. Every linear operator in an n-dimensional vector space has n distinct eigenvalues.
ii. A non-zero sum of two eigenvectors of a matrix $\mathbf{A}$ is an eigenvector.
iii. If a matrix $\mathbf{A}$ has the positive semidefinite property, i.e., $\mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0$ for all $\mathbf{x}$, then its eigenvalues must be non-negative.
iv. The rank of a matrix can exceed the number of distinct non-zero eigenvalues.
v. A non-zero sum of two eigenvectors of a matrix $\mathbf{A}$ corresponding to the same eigenvalue λ is always an eigenvector.

(c) (i) Every linear operator in an n-dimensional vector space has n distinct eigenvalues

Answer: **FALSE**

Counterexample: The identity matrix I_n is a n×n matrix with n eigenvalues of 1 which are not distinct (aka only 1 distinct eigenvalue) $I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & \cdots & & 1 \end{pmatrix}$

Counterexample 2: $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ eigenvalues can have multiplicities so if there is an eigenvalue with m multiplicities (or multiple eigenvalues have multiplicities) then the total number of distinct eigenvalues will be less than n

Therefore, not every linear operator has n distinct eigenvalues

(c)(ii) A non-zero sum of two eigenvectors of a matrix A is an eigenvector

Answer: **FALSE**

Any sum of eigenvectors is an eigenvector if they share the same eigenvalue

Let $\vec{x}_1$ and $\vec{x}_2$ be eigenvectors of A corresponding to two different eigenvalues λ_1 and λ_2

$$A\vec{x}_1 = \lambda_1\vec{x}_1 \; ; \; A\vec{x}_2 = \lambda_2\vec{x}_2$$

Considering the sum $\vec{x}_1+\vec{x}_2$. Applying the matrix A to the sum

$$A(\vec{x}_1+\vec{x}_2) = A\vec{x}_1 + A\vec{x}_2 = \lambda_1\vec{x}_1 + \lambda_2\vec{x}_2$$

unless $\lambda_1=\lambda_2$, it is not a scalar multiple and therefore, $\vec{x}_1+\vec{x}_2$ are generally not an eigenvector

if the eigenvalue is shared $\lambda = \lambda_1 = \lambda_2$ then the sum will be an eigenvector

$$A(\vec{x}_1+\vec{x}_2) = \lambda\vec{x}_1 + \lambda\vec{x}_2 = \lambda(\vec{x}_1+\vec{x}_2)$$

Therefore, the statement is not generally true and only applies when the two eigenvectors share the same eigenvalue

(c)(iii) If a matrix A has the positive semidefinite property (i.e. $\vec{x}^TA\vec{x} \geq 0$ for all x, then its eigenvalues must be non-negative

Answer: **True**

For any eigenvector $\vec{x}$ of matrix A that corresponds to an eigenvalue λ

$$\vec{x}^TA\vec{x} = \lambda\|\vec{x}\|^2$$

$\|\vec{x}\|^2 > 0$ because $\vec{x}$ is non-zero and A is positive semidefinite $v^TAv \geq 0$ which means that $\lambda \geq 0$

Therefore, all eigenvalues of A are non-negative when A is positive semidefinite

(c)(iv) The rank of a matrix can exceed the number of distinct non-zero eigenvalues

Answer: **True**

Since eigenvalues can have multiplicities, the number of distinct non-zero eigenvalues

Example

$$D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

we have one distinct eigenvalue of 2 when the rank of the matrix is 3 $\therefore$ # eigenvalues < rank

$1 < 3$

(c)(v) A non-zero sum of two eigenvectors of a matrix A corresponding to the same eigenvalue λ is always an eigenvector

Answer: **True**

if $\vec{x}_1$ and $\vec{x}_2$ are eigenvectors of A corresponding to the same eigenvalue λ

$$A\vec{x}_1 = \lambda\vec{x}_1 \text{ and } A\vec{x}_2 = \lambda\vec{x}_2$$

$\vec{x}_1+\vec{x}_2$ and apply $A \rightarrow A(\vec{x}_1+\vec{x}_2) = A\vec{x}_1 + A\vec{x}_2 = \lambda\vec{x}_1 + \lambda\vec{x}_2 = \lambda(\vec{x}_1+\vec{x}_2)$

This means it is a scalar multiple and $\vec{x}_1+\vec{x}_2$ is also an eigenvector corresponding to λ

Problem #2: Probability Refresher (25 Points)

(a) (10 points) A and B are involved in a duel. The rules of the duel are that they are to pick up their guns and shoot at each other simultaneously. If one or both are hit, then the duel is over. If both shots miss, then they repeat the process. Suppose that the results of the shots are independent and that each shot of A will hit B with probability p_A and each shot of B will hit A with probability p_B. What is:

i. (2 points) the probability that A is not hit?
ii. (2 points) the probability that both duelists are hit?
iii. (2 points) the probability that the duel ends after the n^{th} round of shots?
iv. (2 points) the conditional probability that the duel ends after the n^{th} round of shots given that A is not hit?
v. (2 points) the conditional probability that the duel ends after the n^{th} round of shots given that both duelists are hit?

NOTE:
p_A: Probability A hits B
p_B: Probability B hits A

(a)(i) Probability A is not hit by B → $P(\text{A not hit by B}) = 1 - p_B$

(a)(ii) Probability that both duelists are hit?

The shots are independent and therefore $P(\text{both are hit}) = p_A \cdot p_B$
↳ mult

(a)(iii) Probability that the duel ends after the n^{th} round of shots?

In order for the duel to end in the n^{th} round, both duelists have to miss for all n-1 rounds prior to the n^{th} round

Probability that both duelists miss in a single round: $(1-p_A)(1-p_B)$

Probability that both duelists miss n-1 rounds: $[(1-p_A)(1-p_B)]^{n-1}$

Probability that at least one duelist hits in the n^{th} round: $1-[(1-p_A)(1-p_B)]$

$P(\text{duel ends after n rounds}) = [(1-p_A)(1-p_B)]^{n-1} \cdot [1-(1-p_A)(1-p_B)]$

(a)(iv) The conditional probability that the duel ends after the n^{th} round of shots given that A is not hit?

Probability that neither duelist is hit for (n-1) rounds: $[(1-p_A)(1-p_B)]^{(n-1)}$

Probability that A hits B aka A is not hit: p_A

Both events have happened independently ∴ multiplied together

$P(\text{The duel ends after } n^{th} \text{ round} \mid \text{A is not hit}) = [(1-p_A)(1-p_B)]^{n-1} p_A$

(a)(v) The conditional probability that the duel ends after the n^{th} round of shots given that both duelists are hit?

Probability that neither duelist is hit after n-1 rounds $[(1-p_A)(1-p_B)]^{(n-1)}$

Probability that both duelists are hit: $p_A \cdot p_B$

$P(\text{the duel ends after } n^{th} \text{ round} \mid \text{Both duelists hit}) = [(1-p_A)(1-p_B)]^{(n-1)} [p_A \cdot p_B]$

Problem 2 Part b

(b) (5 points) Let X be a binary signal, such that $P(X = +1) = P(X = -1) = 0.5$. Suppose X is sent across a noisy channel, with noise N modeled by a zero-mean Gaussian distribution with variance σ^2, where the noise is independent of the signal that was sent. The received signal is $Y = X + N$.

i. (2 points) Find the conditional PDFs of Y given both $\{X = +1\}$ and $\{X = -1\}$.

ii. (2 points) Suppose a detector compares the received signal Y to a fixed threshold γ to decide which signal was sent. Specifically, the detector decides that $X = +1$ was sent if $Y \geq \gamma$ and that $X = -1$ was sent otherwise. For a given threshold γ, express the probability of error in terms of the Φ function, γ, and σ. Recall, if $Z \sim \mathcal{N}(0, 1)$, then

$$P(Z \leq z) = \Phi(z)$$

iii. (1 point) What is the optimal value of γ to minimize the probability of error? For that optimal value of γ, what is the probability of error? Leave your answer for the optimal value of γ as a real number and the probability of error in terms of the Φ function and σ.

(2)(b)(i) Find conditional PDFs of Y

Given $Y = X + N$ and

1. if $X = +1$: Y becomes $Y = 1 + N$, so $Y \sim N(1, \sigma^2)$

conditional pdf: $f_{Y|X=+1}(y) = (2\pi\sigma)^{-\frac{1}{2}} e^{-\frac{(y-1)^2}{2\sigma^2}}$

2. if $X = -1$: Y becomes $Y = -1 + N$ $\therefore$ $Y \sim N(-1, \sigma^2)$

conditional pdf $f_{Y|X=-1}(y) = (2\pi\sigma)^{-\frac{1}{2}} e^{-\frac{(y+1)^2}{2\sigma^2}}$

(2)(b)(ii) Express the probability of error in terms of Φ, γ, and σ

$\begin{cases} X = +1 & \text{if } Y \geq \gamma \\ X = -1 & \text{otherwise} \end{cases}$ → probability of error happens when $\begin{cases} X = +1 & \text{but } Y < \gamma \\ X = -1 & \text{but } Y \geq \gamma \end{cases}$

Case 1: $X = +1$ but $Y < \gamma$ $\quad P(\text{error} \mid X = +1) = P(Y < \gamma \mid X = +1) = \Phi\left(\frac{\gamma - 1}{\sigma}\right)$

Case 2: $X = -1$ but $Y \geq \gamma$ $\quad P(\text{error} \mid X = -1) = P(Y \geq \gamma \mid X = -1) = 1 - \Phi\left(\frac{\gamma + 1}{\sigma}\right)$

Total probability: $P(\text{error}) = 0.5 \cdot \Phi\left(\frac{\gamma - 1}{\sigma}\right) + 0.5 \cdot \left[1 - \Phi\left(\frac{\gamma + 1}{\sigma}\right)\right]$

(2)(b)(iii) What is optimal value of γ to minimize probability of error? What is the probability of error for the optimal value of γ?

$$\frac{\gamma - 1}{\sigma} = \frac{\gamma + 1}{\sigma} \rightarrow \gamma - 1 = \gamma + 1 \rightarrow \gamma = 0$$

Substitute $\gamma = 0$ into the probability of error

$$P(\text{error}) = 0.5\,\Phi\left(-\frac{\gamma - 1}{\sigma}\right) + 0.5\left[1 - \Phi\left(\frac{\gamma + 1}{\sigma}\right)\right] = 0.5\,\Phi\left(-\frac{1}{\sigma}\right) + 0.5\left[1 - \Phi\left(\frac{1}{\sigma}\right)\right]$$

Use $\Phi(-z) = 1 - \Phi(z)$ $\therefore P(\text{error}) = 0.5\left[1 - \Phi\left(\frac{1}{\sigma}\right)\right] + 0.5\left[1 - \Phi\left(\frac{1}{\sigma}\right)\right] = 1 - \Phi\left(\frac{1}{\sigma}\right)$

OR $P(\text{error}) = 0.5\,\Phi\left(-\frac{1}{\sigma}\right) + 0.5\,\Phi\left(-\frac{1}{\sigma}\right) = \Phi\left(-\frac{1}{\sigma}\right)$

$\therefore P(\text{error}) = \Phi\left(-\frac{1}{\sigma}\right) = 1 - \Phi\left(\frac{1}{\sigma}\right)$

Problem 2 Part c

(c) (5 points) There is a screening test for lung cancer that looks at the level of LSA (lung specific antigen) in the blood. There are a number of reasons besides lung cancer that a man can have elevated LSA levels. In addition, many types of lung cancer develop so slowly that they are never a problem. Unfortunately, there is currently no test to distinguish the different types and using the test is controversial because it's hard to quantify the accuracy rates and the harm done by false positives. For this problem, we will call a positive test a true positive if it catches a dangerous type of lung cancer. Also, we will assume the following numbers:

- Rate of dangerous type of lung cancer among men over 30 = 0.0005
- True positive rate for the test = 0.9
- False positive rate for the test = 0.01

Suppose you randomly select a man over 30 and perform a screening test.

i. (3 points) What is the probability that the man has a dangerous type of the disease given that he had a positive test?

ii. (2 points) What is the probability that the man has a dangerous type of the disease given that he had a negative test?

$P(\text{Dangerous}) = 0.0005$
$P(\text{Not Dangerous}) = 1 - P(\text{Dangerous}) = 0.9995$
$P(\text{Positive} \mid \text{Dangerous}) = 0.9$ (The true positive rate)
$P(\text{Positive} \mid \text{Not Dangerous}) = 0.01$ (The false positive rate)
$P(\text{Negative} \mid \text{Not Dangerous}) = 0.99$ (The negative rate)
$P(\text{Negative} \mid \text{Dangerous}) = 0.1$ (The false negative rate)

Bayes Theorem

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Part (i): $P(\text{Dangerous} \mid \text{Positive})$

$$P(\text{Dangerous} \mid \text{Positive}) = \frac{P(\text{Positive} \mid \text{Dangerous})\, P(\text{Dangerous})}{P(\text{Positive})}$$ using Bayes Theorem

$$P(\text{Positive}) = P(\text{Positive} \mid \text{Dangerous}) \cdot P(\text{Dangerous}) + P(\text{Positive} \mid \text{Not Dangerous}) \cdot P(\text{Not Dangerous})$$
$$= (0.9)(0.0005) + (0.01)(0.9995) = 0.00045 + 0.009995 = 0.010445$$

$$P(\text{Dangerous} \mid \text{Positive}) = \frac{(0.9)(0.0005)}{0.010445} = \frac{0.00045}{0.010445} \approx 0.0431$$

$P(\text{Dangerous} \mid \text{Positive}) = 0.0431$

Part (ii): $P(\text{Dangerous} \mid \text{Negative})$

$$P(\text{Dangerous} \mid \text{Native}) = \frac{P(\text{Negative} \mid \text{Dangerous})\, P(\text{Dangerous})}{P(\text{Negative})}$$ using Bayes Theorem

$$P(\text{Negative}) = P(\text{Negative} \mid \text{Dangerous}) \cdot P(\text{Dangerous}) + P(\text{Negative} \mid \text{Not Dangerous})\, P(\text{Not Dangerous})$$
$$= (0.1)(0.0005) + (0.99)(0.9995) = 0.00005 + 0.989505 = 0.989555$$

$$P(\text{Dangerous} \mid \text{Negative}) = \frac{0.00005}{0.989555} \approx 0.0000505$$

$P(\text{Dangerous} \mid \text{Negative}) = 0.0000505$

Problem 2 Part d

(d) (5 points) A family with three daughters and three sons needs to go to the grocery store. Besides the father, who is driving the car, exactly three of the children can come along to the grocery store with him. Suppose that the three children to join the father are chosen randomly, and all such choices are equally likely. Let X denote the number of daughters who accompany the father to the grocery. Let $X_i = 1$ if the i^{th} child who joins the father is a girl, and $X_i = 0$ otherwise.

i. (1 point) Find $Cov(X_i, X_i) = E(X_iX_i) - E(X_i)E(X_i)$, for $1 \leq i \leq 3$. Leave your answer as a fraction.

ii. (2 points) Find $Cov(X_i, X_j) = E(X_iX_j) - E(X_i)E(X_j)$, for $1 \leq i, j \leq 3$ and $i \neq j$. Leave your answer as a fraction.

iii. (2 points) Find $Var(X)$. Leave your answer as a fraction.

(i) Cov(Xi, Xi)

$Cov(X_i) = var(X_i) = E[X_i^2] - E[X_i]E[X_i] = E[X_i^2] - (E[X_i])^2$

$E[X_i] = P(X_i = 1) = \frac{3}{6} = \frac{1}{2}$

$\therefore Cov(X_i) = E[X_i^2] - E[X_i]^2 = \frac{1}{2} - (\frac{1}{2})^2 = \frac{1}{2} - \frac{1}{4} = \frac{1}{4}$

Σ: variance covariance matrix

$$\begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2 & \cdots & \sigma_1\sigma_n \\ \sigma_2\sigma_1 & \sigma_2^2 & & \sigma_2 n \\ \vdots & & \ddots & \vdots \\ \sigma_n\sigma_1 & \cdots & & \sigma_n^2 \end{pmatrix}$$

(ii) Find Cov(Xi, Xj)

$Cov(X_i, X_j) = E[X_iX_j] - E[X_i]E[X_j]$ $\quad E[X_i] = P(X_i = 1) = \frac{1}{2}$, $E[X_j] = P(X_j = 1) = \frac{1}{2}$

$X_iX_j = 1$ if both i^{th} and j^{th} children are both daughters

choosing 2 girls out of 3 children → $\frac{\left[\binom{3}{2}\binom{3}{1}\right]}{\binom{6}{3}} = \frac{3 \cdot 3}{20} = \frac{9}{20}$ ← total ways of choosing 3 kids from 6 children

$\therefore E[X_iX_j] = \frac{9}{20} \rightarrow \frac{9}{20} \times \frac{1}{3} = \frac{3}{20}$ ← we want combinations

$cov(X_i, X_j) = E[X_iX_j] - E[X_i]E[X_j] = \frac{3}{20} - (\frac{1}{2})(\frac{1}{2}) = \frac{3}{20} - \frac{1}{4} = \frac{3}{20} - \frac{5}{20} = -\frac{2}{20} = -\frac{1}{10}$

(iii) Var(X)

NOTE: From stats 100B
variance of sum = sum of variances

$var(X) = var(X_1 + X_2 + X_3)$

$= \sum_{i=1}^{3} var(X_i) + 2 \sum_{1 \leq i < j \leq 3} cov(X_i, X_j)$

$= (\frac{1}{4} + \frac{1}{4} + \frac{1}{4}) + 2 \cdot (-\frac{1}{10} - \frac{1}{10} - \frac{1}{10})$

$= \frac{3}{4} + 2(-\frac{3}{10}) = \frac{3}{4} - \frac{6}{10} = \frac{15}{20} - \frac{12}{20} = \frac{3}{20}$

Final Answers

i $Cov(X_i, X_i) = \frac{1}{4}$

ii $cov(X_i, X_j) = -\frac{2}{20} = -\frac{1}{10}$

iii $var(X) = \frac{3}{20}$

Problem 3 Multivariate Derivatives

(a) (1 points) Let $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$, and $\mathbf{A} \in \mathbb{R}^{n \times m}$. What is $\nabla_{\mathbf{x}} \mathbf{x}^T \mathbf{A} \mathbf{y}$?

(b) (1 points) Let $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$, and $\mathbf{A} \in \mathbb{R}^{n \times m}$. What is $\nabla_{\mathbf{y}} \mathbf{x}^T \mathbf{A} \mathbf{y}$?

(c) (1 points) Let $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$, and $\mathbf{A} \in \mathbb{R}^{n \times m}$. What is $\nabla_{\mathbf{A}} \mathbf{x}^T \mathbf{A} \mathbf{y}$?

(d) (1 points) Let $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{n \times n}$, and let $f = \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x}$. What is $\nabla_{\mathbf{x}} f$?

(e) (1 points) Let $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times n}$ and $f = \text{tr}(\mathbf{AB})$. What is $\nabla_{\mathbf{A}} f$?

(f) (2 points) Let $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times n}$ and $f = \text{tr}(\mathbf{BA} + \mathbf{A}^T \mathbf{B} + \mathbf{A}^2 \mathbf{B})$. What is $\nabla_{\mathbf{A}} f$?

(g) (3 points) Let $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times n}$ and $f = \|\mathbf{A} + \lambda \mathbf{B}\|_F^2$. What is $\nabla_{\mathbf{A}} f$?

(a) $\nabla_x x^T A y$

$(1 \times n)(n \times m)(m \times 1) = 1 \times 1$ scalar $\therefore \nabla_x x^T A y = Ay$

(b) $\nabla_y x^T A y = \nabla_y \text{tr}(x^T A y) = \nabla_y \text{tr}(y x^T A)$

we can introduce the trace bc trace of scalar is a scalar

$\nabla_y \text{tr}(y x^T A) = (x^T A)^T = A^T x$

For any $f = \text{tr}(Cy)$ where C is a matrix independent of y, derivative w.r.t y will be $\nabla_y \text{tr}(Cy) = C^T$

$\therefore \nabla_y (x^T A y) = (x^T A)^T = A^T x$

(c) $\nabla_A x^T A y = \nabla_A \text{tr}(x^T A y) = \nabla_A \text{tr}(y x^T A) = (y x^T)^T = x y^T$ $\therefore \nabla_A x^T A y = (y x^T)^T = x y^T$

(we apply the trace; use the $\nabla_y \text{tr}(Cy) = C^T$)

(d) $\nabla_x f$ where $f = x^T A x + b^T x$

$\nabla_x f = \nabla_x (x^T A x + b^T x) = \nabla_x (x^T A x) + \nabla_x (b^T x) = (A + A^T) x + b$

(e) ∇f where $f = \text{tr}(AB) \rightarrow \nabla_A \text{tr}(AB) = B^T$ (use the rule $\nabla_y \text{tr}(Cy) = C^T$)

(f) $\nabla_A f$ where $f = \text{tr}(BA + A^T B + A^2 B)$

$\nabla_A f = \nabla_A \text{tr}(BA + A^T B + A^2 B) = \underset{①}{\nabla_A \text{tr}(BA)} + \underset{②}{\nabla_A \text{tr}(A^T B)} + \underset{③}{\nabla_A \text{tr}(A^2 B)}$

① $\nabla_A \text{tr}(A^T B) = B^T$ ② $\nabla_A \text{tr}(A^T B)^T = \text{tr}(B^T A) = (B^T)^T = B$ ③ $\nabla_A \text{tr}(A^2 B) = AB^T + BA^T$ (symmetric $\rightarrow 2AB$)

$\nabla_A f = B^T + B + 2AB$

(g) $\nabla_A f$ where $f = \|A + \lambda B\|_F^2$

$\nabla_A f = \nabla_A \|A + \lambda B\|_F^2 = \nabla_A \text{tr}((A + \lambda B)^T (A + \lambda B)) = \text{tr}(A^T A + \lambda A^T B + \lambda B^T A + \lambda^2 B^T B)$

$\nabla_A f = 2A + \lambda B + \lambda B + 0 \rightarrow \nabla_A f = 2A + 2\lambda B$

Problem #4

4. (10 points) **Deriving least-squares with matrix derivatives.**
In least-squares, we seek to estimate some multivariate output $\mathbf{y}$ via the model

$$\hat{\mathbf{y}} = \mathbf{W}\mathbf{x}$$

In the training set we're given paired data examples $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ from $i = 1, \ldots, n$. Least-squares is the following quadratic optimization problem:

$$\min_{\mathbf{W}} \quad \frac{1}{2}\sum_{i=1}^{n} \left\| \mathbf{y}^{(i)} - \mathbf{W}\mathbf{x}^{(i)} \right\|^2$$

Derive the optimal $\mathbf{W}$.

Where $\mathbf{W}$ is a matrix, and for each example in the training set, both $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ $\forall i = 1, \ldots, n$ are vectors.
Hint: you may find the following derivatives useful:

$$\begin{aligned} \frac{\partial \mathrm{tr}(\mathbf{WA})}{\partial \mathbf{W}} &= \mathbf{A}^T \\ \frac{\partial \mathrm{tr}(\mathbf{WAW}^T)}{\partial \mathbf{W}} &= \mathbf{WA}^T + \mathbf{WA} \end{aligned}$$

$$\min_W \frac{1}{2}\sum_{i=1}^{n} \|y^{(i)} - Wx^{(i)}\|^2 = \frac{1}{2}\sum_{i=1}^{n} (y^{(i)} - Wx^{(i)})^T (y^{(i)} - Wx^{(i)})$$

$$= \frac{1}{2}\sum_{i=1}^{n} y^{(i)T}y^{(i)} - x^{(i)T}W^T y^{(i)} - y^{(i)T}Wx^{(i)} + (Wx^{(i)})^T(Wx^{(i)})$$

$$= \frac{1}{2}\sum_{i=1}^{n} (-2y^{(i)T}Wx^{(i)} + x^{(i)T}W^TWx^{(i)})$$

Terms dependent on w
[$y^{(i)T}y^{(i)}$ isn't dependent on w at all]

Written in matrix form

$$\min_W \frac{1}{2}\sum_{i=1}^{n} \|y^{(i)} - Wx^{(i)}\|^2 = \min \frac{1}{2}\|Y - WX\|_F^2$$

Y = matrix of target vectors $y^{(i)}$
X = matrix of target vectors $x^{(i)}$

$$\min \frac{1}{2}\|Y - WX\|_F^2 = \mathrm{tr}((Y - WX)^T(Y - WX)) = \mathrm{tr}(Y^TY - Y^TWX - X^TW^TY + (WX)^T(WX))$$

$$= \mathrm{tr}(Y^TY) - 2\mathrm{tr}(Y^TWX) + \mathrm{tr}((WX)^T(WX))$$

$$\frac{\partial}{\partial W}[\mathrm{tr}(Y^TY) - 2\mathrm{tr}(Y^TWX) + \mathrm{tr}((WX)^T(WX))] = 0 - YX^T + 2WXX^T \stackrel{!}{=} 0$$

$$-YW^T + WXX^T = 0$$

$$WXX^T = YW^T \longrightarrow W = YX^T(XX^T)^{-1}$$

Problem #5

5. (10 points) **Regularized least squares**

In lecture, we worked through the following least squares problem

$$\arg\min_{\theta} \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - \theta^T \hat{\mathbf{x}}^{(i)})^2$$

However, the least squares has a tendency to overfit the training data. One common technique used to address the overfitting problem is regularization. In this problem, we work through one of the regularization techniques namely ridge regularization which is also known as the regularized least squares problem. In the regularized least squares we solve the following optimization problem

$$\arg\min_{\theta} \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - \theta^T \hat{\mathbf{x}}^{(i)})^2 + \frac{\lambda}{2}\|\theta\|_2^2$$

where λ is a tunable regularization parameter. From the above cost function it can be observed that we are seeking least squares solution with a smaller 2-norm. Derive the solution to the regularized least squares problem, i.e Find θ^*.

$$\arg\min_{\theta} \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - \theta^T x^{(i)})^2 + \frac{\lambda}{2}\|\theta\|_2^2$$

↳ vectorization $\quad \mathcal{L}(\theta) = \frac{1}{2}\|Y - X\theta\|_2^2 + \frac{\lambda}{2}\|\theta\|_2^2$

$$= \frac{1}{2}(Y - X\theta)^T(Y - X\theta) + \frac{\lambda}{2}\theta^T\theta$$

$$= \frac{1}{2}(Y^TY - 2Y^TX\theta + \theta^TX^TX\theta) + \frac{\lambda}{2}\theta^T\theta$$

$$\nabla_\theta \mathcal{L}(\theta) = \frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \frac{\partial}{\partial \theta}\left[\frac{1}{2}(Y^TY) - Y^TX\theta + \frac{1}{2}\theta^TX^TX\theta + \frac{\lambda}{2}\theta^T\theta\right]$$

$$= -X^TY + X^TX\theta + \lambda\theta \; [=] \; 0$$

$$-X^TY + X^TX\theta + \lambda\theta = 0$$

$$(X^TX + \lambda I)\theta = X^TY$$

$$\theta^* = \hat{\theta} = (X^TX + \lambda I)^{-1}X^TY$$

linear_regression

January 17, 2025

0.1 Linear regression workbook

This workbook will walk you through a linear regression example. It will provide familiarity with Jupyter Notebook and Python. Please print (to pdf) a completed version of this workbook for submission with HW #1.

ECE C147/C247, Winter Quarter 2025, Prof. J.C. Kao, TAs: B. Qu, K. Pang, S. Dong, S. Rajesh, T. Monsoor, X. Yan

[34]:
```
import numpy as np
import matplotlib.pyplot as plt

#allows matlab plots to be generated in line
%matplotlib inline
```

0.1.1 Data generation

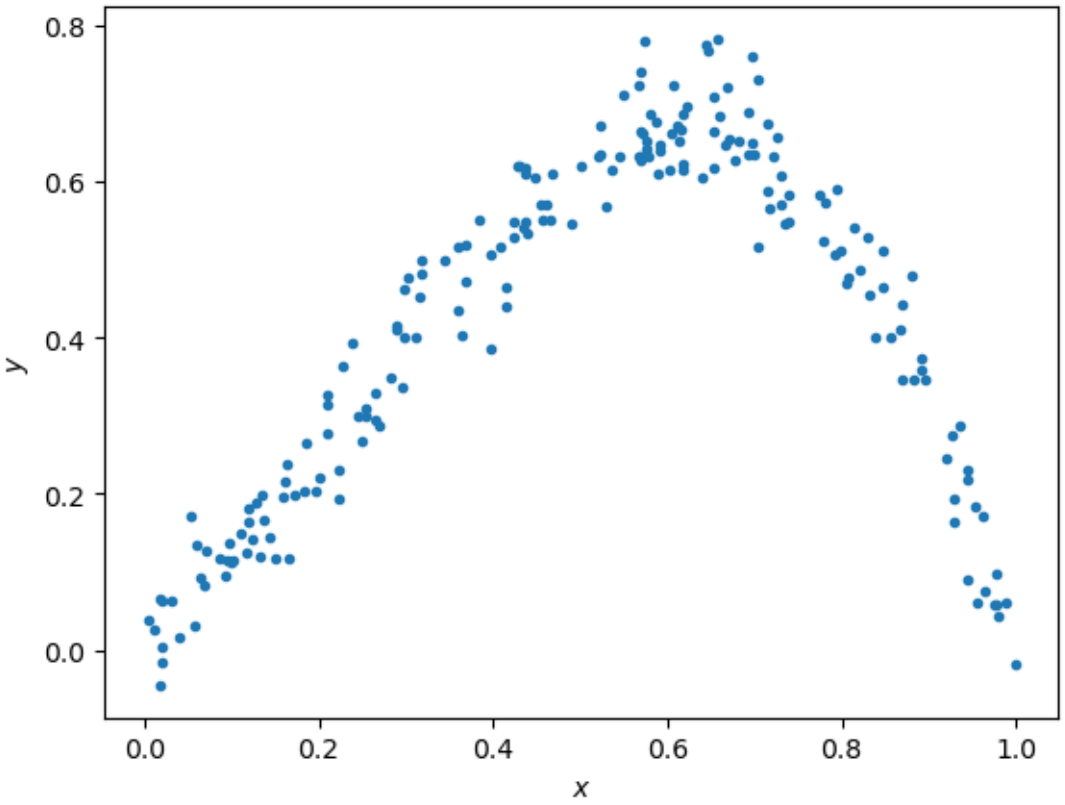
For any example, we first have to generate some appropriate data to use. The following cell generates data according to the model: $y = x + 2x^2 - 3x^3 + \epsilon$

[38]:
```
np.random.seed(0)  # Sets the random seed.
num_train = 200      # Number of training data points

# Generate the training data
x = np.random.uniform(low=0, high=1, size=(num_train,))
y = x + 2*x**2 - 3*x**3 + np.random.normal(loc=0, scale=0.05, size=(num_train,))
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
```

[38]:
```
Text(0, 0.5, '$y$')
```

[32]:
```
#added line by me
plt.show(f)
```



0.1.2 QUESTIONS:

Write your answers in the markdown cell below this one:

(1) What is the generating distribution of x?

(2) What is the distribution of the additive noise ϵ?

0.1.3 ANSWERS:

(1) The code generates a uniform distribution, as we can see in the above graph and the average for the data points. Additionally, according to the code for x, we use "np.random.uniform" for generating our distribution therefore we generate a uniform distribution with values between 0 and 1.

(2) The additive noise generated has a normal distribution, with a mean of 0 and a standard deviation of 0.05. The np.random library did add this as noise to the y output, howver, the general distribution of x on y is x + 2x^2 - 3x^3.

0.1.4 Fitting data to the model (5 points)

Here, we'll do linear regression to fit the parameters of a model $y = ax + b$.

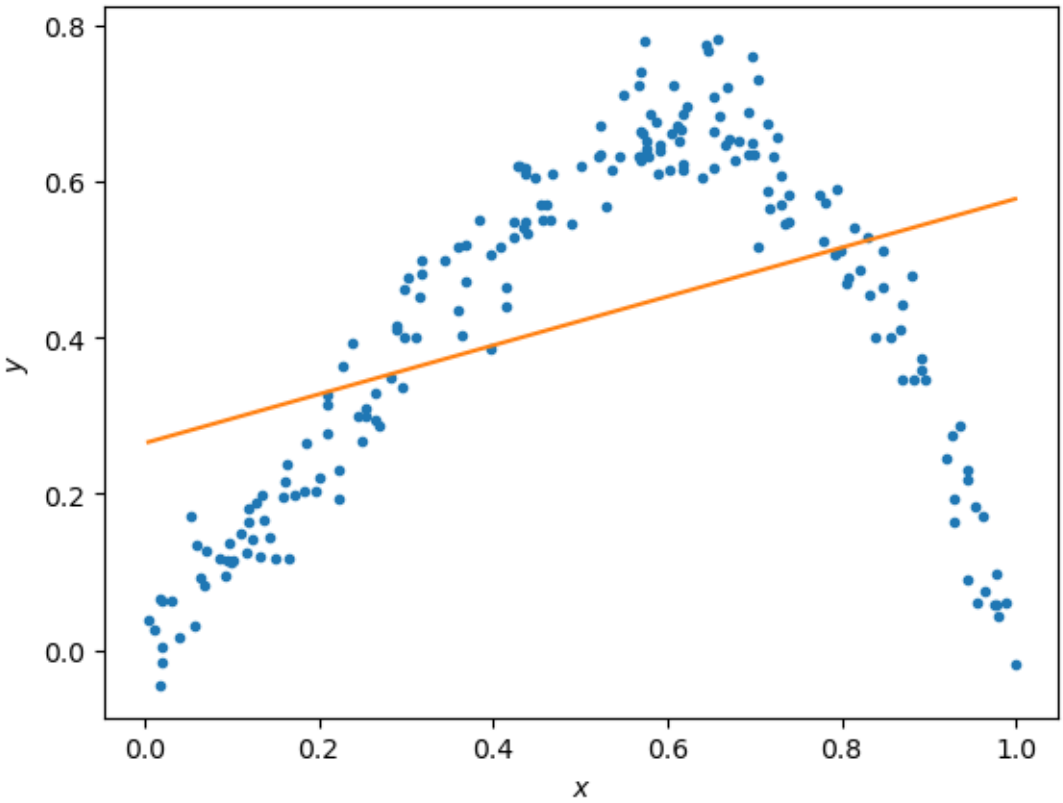
[80]:
```
# xhat = (x, 1)
xhat = np.vstack((x, np.ones_like(x)))

# ==================== #
# START YOUR CODE HERE #
# ==================== #
# GOAL: create a variable theta; theta is a numpy array whose elements are [a,␣
 ↪b]

#theta = np.zeros(2) # please modify this line
theta = np.linalg.inv((xhat.T).T.dot(xhat.T)).dot((xhat.T).T.dot(y))
# ================== #
# END YOUR CODE HERE #
# ================== #
```

[82]:
```
# Plot the data and your model fit.
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')

# Plot the regression line
xs = np.linspace(min(x), max(x),50)
xs = np.vstack((xs, np.ones_like(xs)))
plt.plot(xs[0,:], theta.dot(xs))
plt.show() # had to add
```



0.1.5 QUESTIONS

(1) Does the linear model under- or overfit the data?

(2) How to change the model to improve the fitting?

0.1.6 ANSWERS

(1) The linear model very clearly underfits the data. Based on observation, there is an inverse parabolic path with the general distribution of the data points indicating that a polynomial model would be better. Thus, the linear regression model underfits the data and doesn't capture the shape of the data points very well.

(2) We can improve the fitting of the model by changing the complexity. We can include more polynomial terms. By increasing the number of terms within theta to fit the number of polynomial terms we want, we can minimize the loss and have a better fit to the data.

0.1.7 Fitting data to the model (5 points)

Here, we'll now do regression to polynomial models of orders 1 to 5. Note, the order 1 model is the linear model you prior fit.

[91]:
```
N = 5
xhats = []
thetas = []

# ==================== #
# START YOUR CODE HERE #
# ==================== #

# GOAL: create a variable thetas.
# thetas is a list, where theta[i] are the model parameters for the polynomial␣
 ↪fit of order i+1.
#   i.e., thetas[0] is equivalent to theta above.
#   i.e., thetas[1] should be a length 3 np.array with the coefficients of the␣
 ↪x^2, x, and 1 respectively.
#   ... etc.
for i in range(1, N+1):
    xhat = np.ones_like(x)
    #NOTE: ones_like() function is used to return an array of ones, 1 , with␣
 ↪the same shape as a given array
    for j in range(1, i+1):
        xhat = np.vstack((x**j, xhat))
    x_transpose = xhat.T
    #theta = np.linalg.inv((xhat.T).T.dot(xhat.T)).dot((xhat.T).T.dot(y))
    theta = np.linalg.inv(x_transpose.T.dot(x_transpose)).dot(x_transpose.T.
 ↪dot(y))
    xhats.append(x_transpose)
    thetas.append(theta)

#pass

# ==================== #
# END YOUR CODE HERE #
# ==================== #
```

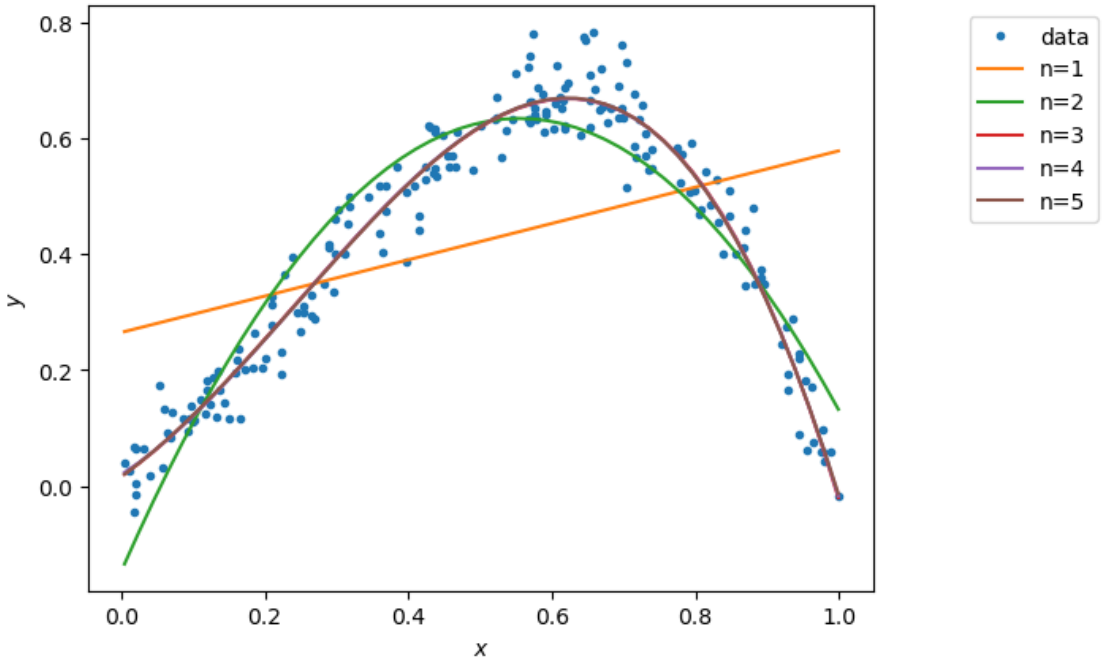
[99]:
```
# Plot the data
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')

# Plot the regression lines
plot_xs = []
for i in np.arange(N):
    if i == 0:
        plot_x = np.vstack((np.linspace(min(x), max(x),50), np.ones(50)))
    else:
```

```
        plot_x = np.vstack((plot_x[-2]**(i+1), plot_x))
    plot_xs.append(plot_x)

for i in np.arange(N):
    ax.plot(plot_xs[i][-2,:], thetas[i].dot(plot_xs[i]))

labels = ['data']
[labels.append('n={}'.format(i+1)) for i in np.arange(N)]
bbox_to_anchor=(1.3, 1)
lgd = ax.legend(labels, bbox_to_anchor=bbox_to_anchor)
plt.show()
```



0.1.8 Calculating the training error (5 points)

Here, we'll now calculate the training error of polynomial models of orders 1 to 5.

[101]:
```
training_errors = []

# ==================== #
# START YOUR CODE HERE #
# ==================== #

# GOAL: create a variable training_errors, a list of 5 elements,
# where training_errors[i] are the training loss for the polynomial fit of␣
 ↪order i+1.
```

```
#pass
for i in range(N):
    difference = y - xhats[i].dot(thetas[i])
    training_errors.append(difference.T.dot(difference)/num_train)

# ================== #
# END YOUR CODE HERE #
# ================== #

print ('Training errors are: \n', training_errors)
```

```
Training errors are:
 [0.041899148752619, 0.005860281754804519, 0.002269334389195937,
0.0022681538153602717, 0.0022670775543125817]
```

0.1.9 QUESTIONS

(1) What polynomial has the best training error?

(2) Why is this expected?

0.1.10 ANSWERS

(1) The 5th order polynomial has the best training error based on the values. Even if it is only marginally better by approximately 0.000001 compared to the other lower order polynomials.

(2) The 5th order polynomial will have the lowest training error in comparison to the lower order polynomials because as the complexity increases, the extra order will allow us to reduce the loss. However, this can make the model more prone to overfitting and do worse with generalization. With each added order, we reduce the loss by increasing the complexity but at the risk of overfitting.

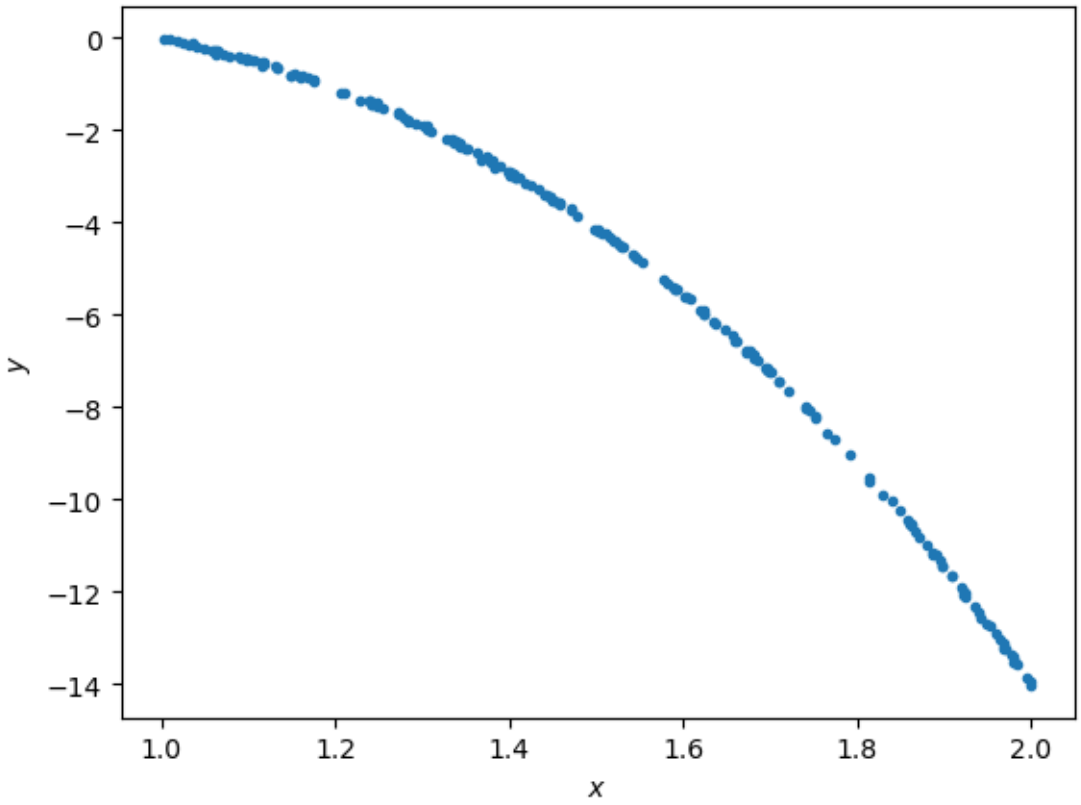
0.1.11 Generating new samples and testing error (5 points)

Here, we'll now generate new samples and calculate testing error of polynomial models of orders 1 to 5.

[104]:
```
x = np.random.uniform(low=1, high=2, size=(num_train,))
y = x + 2*x**2 - 3*x**3 + np.random.normal(loc=0, scale=0.03, size=(num_train,))
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
```

[104]:
```
Text(0, 0.5, '$y$')
```

[106]:
```
plt.show()
```



[108]:
```
xhats = []
for i in np.arange(N):
    if i == 0:
        xhat = np.vstack((x, np.ones_like(x)))
        plot_x = np.vstack((np.linspace(min(x), max(x),50), np.ones(50)))
    else:
        xhat = np.vstack((x**(i+1), xhat))
        plot_x = np.vstack((plot_x[-2]**(i+1), plot_x))

    xhats.append(xhat)
```

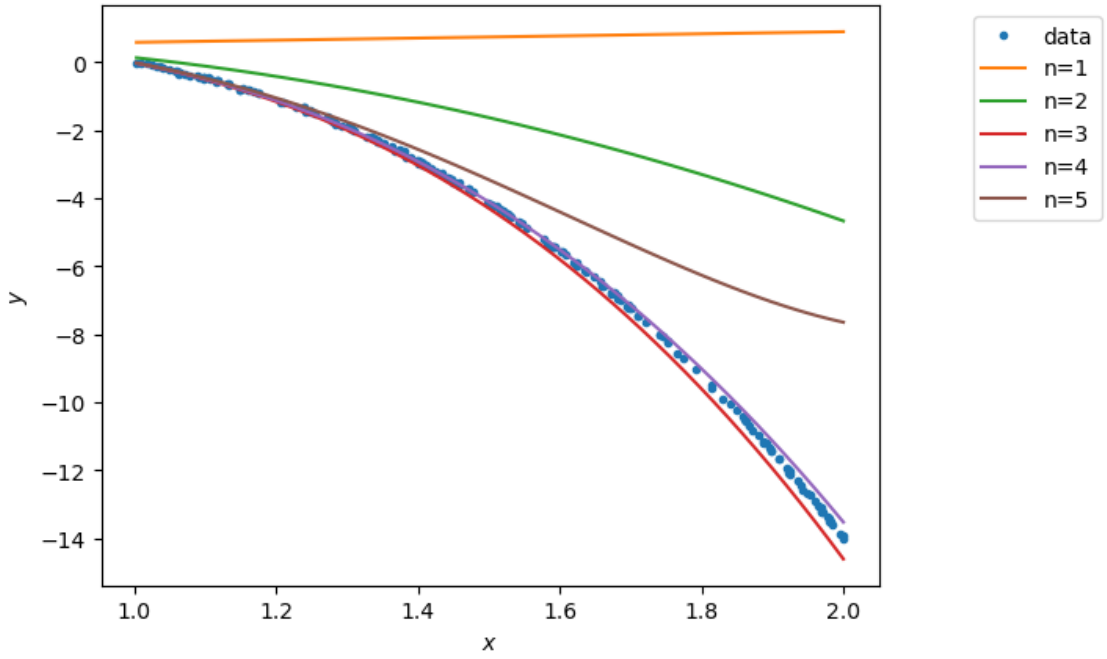
[114]:
```
# Plot the data
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')

# Plot the regression lines
plot_xs = []
for i in np.arange(N):
```

```
    if i == 0:
        plot_x = np.vstack((np.linspace(min(x), max(x),50), np.ones(50)))
    else:
        plot_x = np.vstack((plot_x[-2]**(i+1), plot_x))
    plot_xs.append(plot_x)

for i in np.arange(N):
    ax.plot(plot_xs[i][-2,:], thetas[i].dot(plot_xs[i]))

labels = ['data']
[labels.append('n={}'.format(i+1)) for i in np.arange(N)]
bbox_to_anchor=(1.3, 1)
lgd = ax.legend(labels, bbox_to_anchor=bbox_to_anchor)
#added line
plt.show()
```



[120]:
```
testing_errors = []

# ==================== #
# START YOUR CODE HERE #
# ==================== #

# GOAL: create a variable testing_errors, a list of 5 elements,
# where testing_errors[i] are the testing loss for the polynomial fit of order
  ↪i+1.
```

```
#pass
for i in range(N):
    difference = y - xhats[i].T.dot(thetas[i])
    testing_errors.append((difference.T.dot(difference))/num_train)
# ================== #
# END YOUR CODE HERE #
# ================== #

print ('Testing errors are: \n', testing_errors)
```

```
Testing errors are:
 [54.246317716012214, 18.91115921752976, 0.08693862570503448,
0.030429668694778193, 5.954356338564789]
```

0.1.12 QUESTIONS

(1) What polynomial has the best testing error?

(2) Why polynomial models of orders 5 does not generalize well?

0.1.13 ANSWERS

(1) The 4th order polynomial has the best testing error of 0.030429668694778193 compared to all of the other polynomial orders.

(2) The polynomial order of 5 doesn't generalize well because the extra order of magnitude overfitted to the training data. With higher order polynomials, the training performance will eventually plateau, but the testing performance may decrease as higher complexity models have a tendency to overfit the training data. This means that higher order models are prone to not generalize well to unseen data. The polynomial of order 4 may try to also fit the noise into the training data which can also contribute to the overfitting and poor generalization.