HW1 - Q1

Saturday, January 8, 2022 12:52 PM

Q1 (a) A is a square matrix

$AA^T = I$

(i) Let $A = \begin{bmatrix} \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$

$AA^T = \begin{bmatrix} \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$

$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I_2$

$Av = \lambda v$

$\Rightarrow (A - \lambda I)v = 0$

$\Rightarrow |A - \lambda I| = 0$ $\quad \left| \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right| = 0$

$\begin{vmatrix} -\frac{1}{\sqrt{2}} - \lambda & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} - \lambda \end{vmatrix} = 0$

$-\left(\frac{1}{\sqrt{2}} + \lambda\right)\left(\frac{1}{\sqrt{2}} - \lambda\right) - \frac{1}{2} = 0$

$-\left(\frac{1}{2} - \lambda^2\right) - \frac{1}{2} = 0$

$\lambda^2 = 1$

Eigen values : $\boxed{\lambda_1 = 1}$ $\boxed{\lambda_2 = -1}$

$(A - \lambda I)v = 0$

Taking $\lambda_1 = 1$,

$\begin{bmatrix} \frac{-1}{\sqrt{2}} - 1 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} - 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$

$-\left(\frac{1}{\sqrt{2}} + 1\right)v_1 + \frac{1}{\sqrt{2}}v_2 = 0 \quad \Rightarrow -(1+\sqrt{2})v_1 + v_2 = 0$

$\frac{1}{\sqrt{2}}v_1 + \left(\frac{1}{\sqrt{2}} - 1\right)v_2 = 0 \quad \Rightarrow v_1 + (1-\sqrt{2})v_2 = 0$

$\Rightarrow \frac{v_1}{\sqrt{2}} = \frac{1}{(\sqrt{2}+1)}$ $\quad \Rightarrow$ eigen vector $= K \begin{bmatrix} 1 \\ \sqrt{2}+1 \end{bmatrix}$

Taking $\lambda_2 = -1$,

$\begin{bmatrix} -\frac{1}{\sqrt{2}} + 1 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} + 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$

$\left(\frac{-1}{\sqrt{2}} + 1\right)v_1 + \frac{1}{\sqrt{2}}v_2 = 0 \quad \Rightarrow (-1+\sqrt{2})v_1 + v_2 = 0$

$\frac{1}{\sqrt{2}}v_1 + \left(\frac{1}{\sqrt{2}} + 1\right)v_2 = 0 \quad \Rightarrow v_1 + (1+\sqrt{2})v_2 = 0$

$\Rightarrow \frac{v_1}{\sqrt{2}} = \frac{1}{(1-\sqrt{2})}$ $\Rightarrow$ eigen vector $= k\begin{bmatrix} 1 \\ 1-\sqrt{2} \end{bmatrix}$

We observe that eigenvalues have norm $= 1$

Also, $\begin{bmatrix} 1 & \sqrt{2}+1 \end{bmatrix}\begin{bmatrix} 1 \\ 1-\sqrt{2} \end{bmatrix} = 1 + 1 - 2 = 0 \Rightarrow$ eigenvectors are orthogonal

(ii) To show: A has eigenvalues with norm 1

$AA^T = I \quad \Rightarrow A^TA = I$

$Av = \lambda v$

Taking norm of both sides

$\|Av\|^2 = \|\lambda v\|^2$

$\Rightarrow (Av)^T Av = \|\lambda\|^2 \|v\|^2$

$\Rightarrow v^T A^T A v = \|\lambda\|^2 \|v\|^2$

$\Rightarrow v^T v = \|\lambda\|^2 \|v\|^2$

$\Rightarrow \|v\|^2 = \|\lambda\|^2 \|v\|^2$

$\Rightarrow \|\lambda\|^2 = 1$

$\Rightarrow \boxed{\|\lambda\| = 1}$ as it cant be negative

$\Rightarrow$ Eigenvalues of A have norm $= 1$

(iii) To show: eigenvectors of A corresponding to distinct eigenvalues are orthogonal

Let 2 distinct eigenvalues of A be λ_1 & λ_2 with corresponding eigenvectors v_1 & v_2

$\Rightarrow Av_1 = \lambda_1 v_1 \quad$ — (1)

$Av_2 = \lambda_2 v_2 \quad$ — (2)

Taking transpose of (1)

$v_1^T A^T = \lambda_1 v_1^T$

$\Rightarrow v_1^T A^T v_2 = \lambda_1 v_1^T v_2$

Multiplying A^T to (2)

$A^T A v_2 = \lambda_2 A^T v_2$

$\Rightarrow v_2 = \lambda_2 A^T v_2$

$\Rightarrow A^T v_2 = \frac{v_2}{\lambda_2}$

$\Rightarrow v_1^T \frac{v_2}{\lambda_2} = \lambda_1 v_1^T v_2$ $\quad \boxed{v_1^T v_2 = \lambda_1 \lambda_2 v_1^T v_2}$

$\lambda_1 \lambda_2$ cannot be 1 as $\lambda_1 \neq \lambda_2$ & both have unit norm

$\Rightarrow \boxed{v_1^T v_2 = 0}$ Hence they are orthogonal

0

(iv) When a vector x is multiplied by A, where A is orthogonal, the vector is rotated or reflected but lengths and angles are preserved.

There fore norm of a vector is invariant under multiplication by an orthogonal matrix

$$\|Ax\|^2 = (Ax)^T Ax = x^T A^T A x = x^T x = \|x\|^2$$

$$\Rightarrow \boxed{\|Ax\| = \|x\|}$$

Thus, we see that the length is preserved and it will only rotate or reflect

(b)(i) Relationship between singular vectors of A & eigenvectors of AA^T & A^TA

⇒ The left singular vectors of A are eigenvectors of AA^T
⇒ The right singular vectors of A are eigenvectors of A^TA

Writing the SVD of A

$A = UDV^T$, where U & V are orthonormal

$$AA^T = UDV^T VD^TU^T$$
$$= UDD^TU^T$$
$$= U \operatorname{diag}(d) U^{-1}$$ ⇒ left singular vectors of A are eigenvectors of AA^T

$$A^TA = VD^TU^TUDV^T$$
$$= VD^TDV^T$$
$$= V \operatorname{diag}(d) V^{-1}$$ ⇒ right singular vectors of A are eigenvectors of A^TA

(ii) Relationship between singular value of A and eigenvalues of AA^T & A^TA

⇒ For both AA^T & A^TA, the non zero singular values of A are square roots of eigenvalues of A^TA.

For AA^T, $\operatorname{diag}(d) = DD^T$

$$= \underbrace{\begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_m \end{bmatrix}}_{n \times m} \underbrace{\begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \\ & & & \end{bmatrix}}_{m \times n}$$

$$= \begin{bmatrix} d_1^2 & & \\ & d_2^2 & \\ & & \ddots \end{bmatrix}$$

For A^TA, $\operatorname{diag}(d) = D^TD$

$$= \underbrace{\begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \\ & & & \end{bmatrix}}_{m \times n} \underbrace{\begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_m \end{bmatrix}}_{n \times m}$$

$$= \begin{bmatrix} d_1^2 & & \\ & d_2^2 & \\ & & \ddots \end{bmatrix}$$

$$\begin{bmatrix} & & \\ & & d_n^2 \end{bmatrix}_{n\times n}$$

$\Rightarrow d_i^2 = \lambda_i$

$$\begin{bmatrix} & & d_n^2 \\ & & 0 \cdots 0 \end{bmatrix}_{m\times m}$$

$\Rightarrow d_i^2 = \lambda_i$

(c) (i) False

We can take an example to prove this is false

Let $A = I_2$

$|A - \lambda I| = 0$

$\Rightarrow \begin{vmatrix} 1-\lambda & 0 \\ 0 & 1-\lambda \end{vmatrix} = 0 \qquad \Rightarrow (1-\lambda)^2 = 0$

$\Rightarrow \boxed{\lambda = 1} \rightarrow$ only 1 distinct eigenvalue

(ii) False

Let v_1 & v_2 be 2 eigen vectors of A, corresponding to eigenvalues λ_1 & λ

$A(\alpha v_1 + \beta v_2) = A\alpha v_1 + A\beta v_2$

$= \alpha\lambda_1 v_1 + \beta\lambda_2 v_2$

which is not a constant scaling of $(\alpha v_1 + \beta v_2)$

(iii) True

As A is positive semidefinite

$\Rightarrow x^T A x \geq 0 \quad \forall x$

For eigenvalues,

$Ax = \lambda x$

$x^T A x = \lambda x^T x \geq 0$

$\Rightarrow \lambda \|x\|^2 \geq 0 \qquad \Rightarrow \lambda \geq 0$

(iv) True.

Rank of a matrix can exceed the number of distinct non-zero eigenv

example $\rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ we know rank $= 2$

& it has only one distinct eigenvalue $= 1$

(v) True

Let two eigen vectors corresponding to same eigenvalue λ be v_1 & v_2

$$\Rightarrow \quad Av_1 = \lambda v_1$$
$$Av_2 = \lambda v_2$$

$$A(\alpha v_1 + \beta v_2) = \lambda \alpha v_1 + \lambda \beta v_2$$
$$= \lambda(\alpha v_1 + \beta v_2)$$

HW1 - Q2

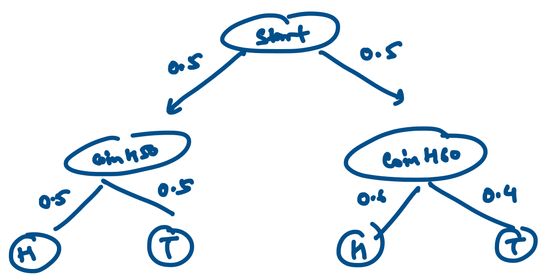
Thursday, January 13, 2022 6:33 PM

Q2 (a)

(i)

Coin H50: $P(H) = 0.5$, $P(T) = 0.5$

Coin H60: $P(H) = 0.6$, $P(T) = 0.4$



$P(\text{coin H50}) = P(\text{coin H60}) = 0.5$ as jar is equally populated with both types of coins

Using Bayes Rule,

$$P(\text{Coin H50} \mid \text{Tail}) = \frac{P(\text{Tail} \mid \text{Coin H50})\, P(\text{Coin H50})}{P(\text{Tail})}$$

$$= \frac{P(\text{Tail} \mid \text{Coin H50})\, P(\text{Coin H50})}{P(\text{Tail} \mid \text{Coin H50})\, P(\text{Coin H50}) + P(\text{Tail} \mid \text{Coin H60})\, P(\text{Coin H60})}$$

$$= \frac{(0.5)(0.5)}{(0.5)(0.5) + (0.4)(0.5)}$$

$$= \frac{25}{25+20} = \boxed{\frac{5}{9}} \approx \boxed{0.556}$$

(ii) To find : $P(\text{Coin H50} \mid THHH)$

Using Bayes Rule,

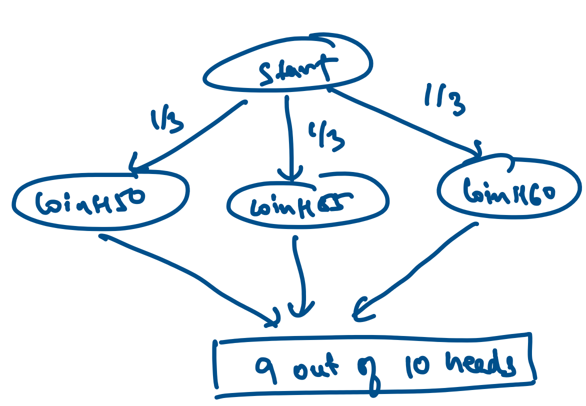
$$P(\text{Coin H50} \mid THHH) = \frac{P(THHH \mid \text{Coin H50})\, P(\text{Coin H50})}{P(THHH)}$$

$$= \frac{P(THHH \mid \text{Coin H50})\, P(\text{Coin H50})}{P(THHH \mid \text{Coin H50})\, P(\text{Coin H50}) + P(THHH \mid \text{Coin H60})\, P(\text{Coin H60})}$$

$$= \frac{(0.5)(0.5)(0.5)(0.5) \times (0.5)}{(0.5)(0.5)(0.5)(0.5) \times (0.5) + (0.4)(0.6)(0.6)(0.6) \times 0.5}$$

$$= \frac{5^4}{5^4 + 4 \times 6^3} = \frac{625}{625 + 864} = \boxed{\frac{625}{1489}} \approx \boxed{0.4197}$$

(iii) $P(\text{Coin H50}) = P(\text{Coin H55}) = P(\text{Coin H60}) = 1/3$



Let X be the event of getting 9 heads out of 10 coin tosses

$$\Rightarrow \boxed{\begin{aligned} P(X \mid \text{Coin H50}) &= {}^{10}C_9 (0.5)^9 (0.5) = 10 \times (0.5)^{10} \\ P(X \mid \text{Coin H55}) &= {}^{10}C_9 (0.55)^9 (0.45) = 10 \times (0.55)^9 (0.45) \\ P(X \mid \text{Coin H60}) &= {}^{10}C_9 (0.6)^9 (0.4) = 10 \times (0.6)^9 \times 0.4 \end{aligned}}$$

$$P(\text{coin } H50 \mid x) = \frac{P(x \mid \text{coin } H50)\, P(\text{coin} H50)}{P(x)}$$

$$= \frac{P(x \mid \text{coin } H50)\, P(\text{coin } H50)}{P(x \mid \text{coin } H50)\, P(\text{coin} H50) + P(x \mid \text{coin} H55)\, P(\text{coin } H55) + P(x \mid \text{coin} H60)\, P(\text{coin} H60)}$$

$$= \frac{(0.5)^{10}}{(0.5)^{10} + (0.55)^9 (0.45) + (0.6)^9 (0.4)} = \boxed{0.1379}$$

$$P(\text{coin } H55 \mid x) = \frac{P(x \mid \text{coin } H55)\, P(\text{coin } H55)}{P(x)}$$

$$= \frac{P(x \mid \text{coin } H55)\, P(\text{coin } H55)}{P(x \mid \text{coin } H50)\, P(\text{coin} H50) + P(x \mid \text{coin} H55)\, P(\text{coin } H55) + P(x \mid \text{coin} H60)\, P(\text{coin} H60)}$$

$$= \frac{(0.55)^9 (0.45)}{(0.5)^{10} + (0.55)^9 (0.45) + (0.6)^9 (0.4)} = \boxed{0.2927}$$

$$P(\text{coin } H60 \mid x) = \frac{P(x \mid \text{coin } H60)\, P(\text{coin } H60)}{P(x)}$$

$$= \frac{P(x \mid \text{coin } H60)\, P(\text{coin } H60)}{P(x \mid \text{coin } H50)\, P(\text{coin} H50) + P(x \mid \text{coin} H55)\, P(\text{coin } H55) + P(x \mid \text{coin} H60)\, P(\text{coin} H60)}$$

$$= \frac{(0.6)^9\, 0.4}{(0.5)^{10} + (0.55)^9 (0.45) + (0.6)^9 (0.4)} = \boxed{0.5693}$$

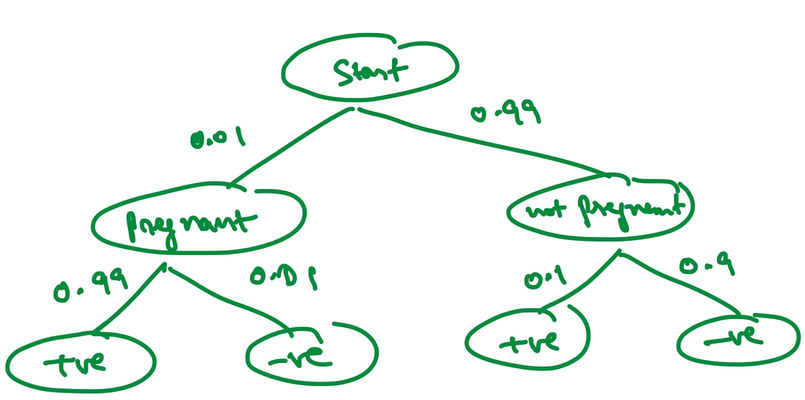
(b) Given :

$P(+ve \mid \text{Pregnant}) = 0.99 \Rightarrow P(-ve \mid \text{pregnant}) = 0.01$

$P(+ve \mid \text{not pregnant}) = 0.10 \rightarrow P(-ve \mid \text{not pregnant}) = 0.9$

$P(\text{not pregnant}) = 0.99 \rightarrow P(\text{pregnant}) = 0.01$

To Find : $P(\text{pregnant} \mid +ve)$



$$P(\text{pregnant} \mid +ve) = \frac{P(+ve \mid \text{pregnant})\, P(\text{pregnant})}{P(+ve)}$$

$$= \frac{P(+ve \mid \text{pregnant})\, P(\text{pregnant})}{P(+ve \mid \text{pregnant})\, P(\text{pregnant}) + P(+ve \mid \text{not pregnant})\, P(\text{not pregnant})}$$

$$= \frac{(0.99)(0.01)}{(0.99)(0.01) + (0.1)(0.99)} = \frac{99}{99 + 99\times 10}$$

$$= \frac{1}{11}$$

⇒ Intuitive sense : There are a lot of false positives in the test. It gives 10% +ve for not pregnant cases which is much greater than the pregnant population. Hence the true positive rate is very low. Intuitively if 100 people tested, 99 were not pregnant but test would say ≈ 10 of them were +ve. & assuming almost 100% accuracy on pregnant cases the 1 pregnant would also get +ve, Hence only 1/11 would be pregnant given +ve

(C) Given :

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

where $x_1, x_2, \dots x_n$ are identically distributed random variables

$$E(x) = \begin{bmatrix} E(x_1) \\ E(x_2) \\ \vdots \\ E(x_n) \end{bmatrix}$$

To find: $E(Ax+b)$ where A & b are deterministic

$$Ax + b =$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{1n} \\ & & \\ a_{m1} & a_{m2} & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{j=1}^{n} a_{1j} x_j + b_1 \\ \sum_{j=1}^{n} a_{2j} x_j + b_2 \\ \vdots \\ \sum_{j=1}^{n} a_{mj} x_j + b_m \end{bmatrix}$$

$$E(Ax+b) = \begin{bmatrix} E\left(\sum_{j=1}^{n} a_{1j} x_j + b_1\right) \\ E\left(\sum_{j=1}^{n} a_{2j} x_j + b_2\right) \\ \vdots \\ E\left(\sum_{j=1}^{n} a_{mj} x_j + b_m\right) \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{n} a_{1j} E(x_j) + b_1 \\ \sum_{j=1}^{n} a_{2j} E(x_j) + b_2 \\ \vdots \\ \sum_{j=1}^{n} a_{mj} E(x_j) + b_m \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{j=1}^{n} a_{1j} E(x_j) \\ \sum_{j=1}^{n} a_{2j} E(x_j) \\ \vdots \\ \sum_{j=1}^{n} a_{mj} E(x_j) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} E(x_1) \\ E(x_2) \\ \vdots \\ E(x_n) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

$$= \boxed{A\,E(x) + b}$$

(d) $\operatorname{cov}(x) = E\left((x - Ex)(x - Ex)^T\right)$

To find: $\operatorname{cov}(Ax+b)$

$$\begin{aligned}
\operatorname{cov}(Ax+b) &= E\left(\left[Ax+b - E(Ax+b)\right]\left[Ax+b - E(Ax+b)\right]^T\right) \\
&= E\left(\left[Ax+b - AE(x) - b\right]\left[Ax+b - AE(x) - b\right]^T\right) \\
&= E\left(\left[Ax - AE(x)\right]\left[Ax - AE(x)\right]^T\right) \\
&= E\left(A(x - E(x))\,(x - E(x))^T A^T\right) \\
&= A\; E\left((x - E(x))(x - E(x))^T\right) A^T \\
&= \boxed{A \operatorname{cov}(x) A^T}
\end{aligned}$$

HW1 - Q3

Saturday, January 15, 2022 1:56 PM

Q3 (a) $x \in \mathbb{R}^n$

$y \in \mathbb{R}^m$

$A \in \mathbb{R}^{n \times m}$

To find: $\nabla_x x^T A y$

$$x^T A y = \underset{1 \times n}{\begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}} \underset{n \times m}{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}} \underset{m \times 1}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}}$$

$$= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} \sum_{j=1}^m a_{1j} y_j \\ \sum_{j=1}^m a_{2j} y_j \\ \\ \sum_{j=1}^m a_{nj} y_j \end{bmatrix}$$

$$= x_1 \sum_{j=1}^m a_{1j} y_j + x_2 \sum_{j=1}^m a_{2j} y_j + \cdots + x_n \sum_{j=1}^m a_{nj} y_j$$

$$\nabla_x x^T A y = \begin{bmatrix} \frac{\partial}{\partial x_1} x^T A y \\ \frac{\partial}{\partial x_2} x^T A y \\ \vdots \\ \frac{\partial}{\partial x_n} x^T A y \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^m a_{1j} y_j \\ \sum_{j=1}^m a_{2j} y_j \\ \vdots \\ \sum_{j=1}^m a_{nj} y_j \end{bmatrix} = Ay$$

(b) To find: $\nabla_y x^T A y$

$$x^T A y = \underset{1 \times n}{\begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}} \underset{n \times m}{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}} \underset{m \times 1}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}}$$

$$= \begin{bmatrix} \sum_{i=1}^n x_i a_{i1} & \sum_{i=1}^n x_i a_{i2} & \cdots & \sum_{i=1}^n x_i a_{im} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$$= \left(\sum_{i=1}^m x_i a_{i1}\right) y_1 + \left(\sum_{i=1}^n x_i a_{i2}\right) y_2 + \cdots + \left(\sum_{i=1}^n x_i a_{im}\right) y_m$$

$$\nabla_y x^T A y = \begin{bmatrix} \frac{\partial}{\partial y_1} x^T A y \\ \frac{\partial}{\partial y_2} x^T A y \\ \vdots \\ \frac{\partial}{\partial y_m} x^T A y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n x_i a_{i1} \\ \sum_{i=1}^n x_i a_{i2} \\ \vdots \\ \sum x_i a_{im} \end{bmatrix} = A^T x$$

(c) To find: $\nabla_A x^T A y$

$$x^TAy = \left(\sum_{i=1}^{n} x_i a_{i1}\right) y_1 + \left(\sum_{i=1}^{n} x_i a_{i2}\right) y_2 + \cdots\cdots + \left(\sum_{i=1}^{n} x_i a_{im}\right) y_m$$

$$\nabla_A x^TAy = \begin{bmatrix} \frac{\partial}{\partial a_{11}} x^TAy & \frac{\partial}{\partial a_{12}} x^TAy & \cdots & \frac{\partial}{\partial a_{1m}} x^TAy \\ \frac{\partial}{\partial a_{21}} x^TAy & \frac{\partial}{\partial a_{22}} x^TAy & \cdots & \frac{\partial}{\partial a_{2m}} x^TAy \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial a_{n1}} x^TAy & \frac{\partial}{\partial a_{n2}} x^TAy & \cdots & \frac{\partial}{\partial a_{nm}} x^TAy \end{bmatrix}$$

$$= \begin{bmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_m \\ x_2y_1 & x_2y_2 & \cdots & x_2y_m \\ \vdots & \vdots & \ddots & \vdots \\ x_ny_1 & x_ny_2 & \cdots & x_ny_m \end{bmatrix}$$

$$= xy^T$$

(d) $A \in R^{n\times n}$

$f = x^TAx + b^Tx$

To find : $\nabla_x f$

$$x^TAx + b^Tx = [x_1\ x_2\ \cdots\ x_n] \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + [b_1\ b_2\ \cdots\ b_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$= [x_1\ x_2\ \cdots\ x_n] \begin{bmatrix} \sum_{j=1}^{n} a_{1j} x_j \\ \sum_{j=1}^{n} a_{2j} x_j \\ \vdots \\ \sum_{j=1}^{n} a_{nj} x_j \end{bmatrix} + \sum_{i=1}^{n} b_i x_i$$

$$= \left(\sum_{j=1}^{n} a_{1j} x_j\right) x_1 + \left(\sum_{j=1}^{n} a_{2j} x_j\right) x_2 \cdots \left(\sum_{j=1}^{n} a_{nj} x_j\right) x_n + \sum_{i=1}^{n} b_i x_i$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} x_i x_j + \sum_{i=1}^{n} b_i x_i$$

$$\nabla_x f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_m} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} x^T A x + b^T x \\ \frac{\partial}{\partial x_2} x^T A x + b^T x \\ \vdots \\ \frac{\partial}{\partial x_n} x^T A x + b^T x \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} a_{i1} x_i + \sum_{j=1}^{n} a_{1j} x_j + b_1 \\ \sum_{i=1}^{n} a_{i2} x_i + \sum_{j=1}^{n} a_{2j} x_j + b_2 \\ \vdots \\ \sum_{i=1}^{n} a_{in} x_i + \sum_{j=1}^{n} a_{nj} x_j + b \end{bmatrix}$$

$$= Ax + A^T x + b$$

$$= (A + A^T)x + b$$

(e) Let $f = tr(AB)$

To find: $\nabla_A f$

Let $A \in \mathbb{R}^{n \times m}$
$B \in \mathbb{R}^{m \times n}$

$$A = \begin{array}{c} n \times m \\ \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \end{array} \qquad B = \begin{array}{c} m \times n \\ \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & & \ddots & \vdots \\ b_{m1} & \cdots & & b_{mn} \end{bmatrix} \end{array}$$

$$AB = \begin{bmatrix} \sum_{j=1}^{m} a_{1j} b_{j1} & \sum_{j=1}^{m} a_{1j} b_{j2} & \cdots & \sum_{j=1}^{m} a_{1j} b_{jn} \\ \sum_{j=1}^{m} a_{2j} b_{j1} & \sum_{j=1}^{m} a_{2j} b_{j2} & \cdots & \sum_{j=1}^{m} a_{2j} b_{jn} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^{m} a_{nj} b_{j1} & \sum_{j=1}^{m} a_{nj} b_{j2} & \cdots & \sum_{j=1}^{m} a_{nj} b_{jn} \end{bmatrix}$$

$$tr(AB) = \sum_{j=1}^{m} a_{1j} b_{j1} + \sum_{j=1}^{m} a_{2j} b_{j2} \cdots \sum_{j=1}^{m} a_{nj} b_{jn}$$

$$\nabla_A tr(AB) = \begin{bmatrix} \frac{\partial\, tr(AB)}{\partial a_{11}} & \frac{\partial\, tr(AB)}{\partial a_{12}} & \cdots & \frac{\partial\, tr(AB)}{\partial a_{1m}} \\ \frac{\partial}{\partial a_{21}} tr(AB) & \frac{\partial}{\partial a_{22}} tr(AB) & \cdots & \frac{\partial}{\partial a_{2m}} tr(AB) \end{bmatrix}$$

$$
\begin{bmatrix}
\vdots & \ddots & & \vdots \\
\frac{\partial}{\partial a_{m1}} tr(AB) & \frac{\partial}{\partial a_{m2}} tr(AB) & \cdots & \frac{\partial}{\partial a_{mn}} tr(AB)
\end{bmatrix}
$$

$$
= \begin{bmatrix}
b_{11} & b_{21} & \cdots & b_{m1} \\
b_{12} & b_{22} & \cdots & b_{m2} \\
\vdots & \vdots & \ddots & \vdots \\
b_{1n} & b_{2n} & \cdots & b_{mn}
\end{bmatrix}
$$

$$= B^T$$

HW1 - Q4

Saturday, January 15, 2022 6:46 PM

Q4 Given model:

$$\hat{y} = Wx \qquad x \in \mathbb{R}^p, \quad y \in \mathbb{R}^m, \quad W \in \mathbb{R}^{m \times p}$$

least-square loss is given by:

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

optimization:

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

Taking derivative:

$$\Rightarrow \frac{\partial}{\partial W} \frac{1}{2} \sum_{i=1}^{n} \| y^{(i)} - Wx^{(i)} \|^2 = 0$$

let $Y = \begin{bmatrix} — y^{(1)T} — \\ — y^{(2)T} — \\ \vdots \\ — y^{(n)T} — \end{bmatrix}$

$X = \begin{bmatrix} — x^{(1)T} — \\ — x^{(2)T} — \\ \vdots \\ — x^{(n)T} — \end{bmatrix}$

vectorizing the problem to the following.

$$\Rightarrow \frac{\partial}{\partial W} \frac{1}{2} \| Y - XW^T \|_F^2 = 0$$

Using $\|A\|_F^2 = tr(A^T A)$, we get

$$\Rightarrow \frac{1}{2} \frac{\partial}{\partial W} tr\left[(Y - XW^T)^T (Y - XW^T) \right] = 0$$

$$\Rightarrow \frac{1}{2} \frac{\partial}{\partial W} tr\left(Y^T Y - W X^T Y - Y^T X W^T - W X^T X W^T \right) = 0$$

Using $\frac{\partial}{\partial W} tr(WA) = A^T$ &

$\frac{\partial}{\partial W} tr(WAW^T) = WA^T + WA$, we get

$$\frac{1}{2}\left[-y^Tx - y^Tx + wx^Tx + wx^Tx\right] = 0$$

$$\Rightarrow \quad wx^Tx = y^Tx$$

$$\Rightarrow \quad \boxed{w = y^Tx\,(x^Tx)^{-1}}$$

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 2022, Prof. J.C. Kao, TAs Y. Li, P. Lu, T. Monsoor, T. wang

In [133…

```
import numpy as np
import matplotlib.pyplot as plt

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

Data generation

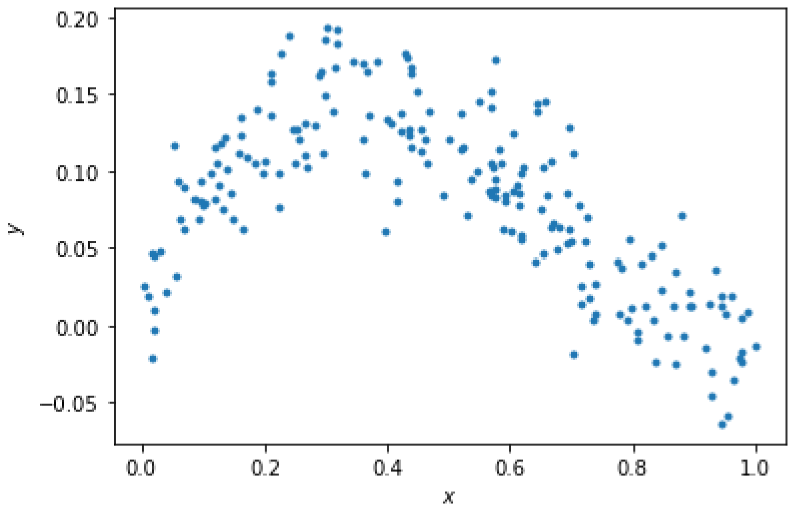
For any example, we first have to generate some appropriate data to use. The following cell generates data according to the model: $y = x - 2x^2 + x^3 + \epsilon$

In [134…

```
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 + 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('$y$')
```

Out[134… Text(0, 0.5, 'y')



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?

ANSWERS:

(1) Uniform Distribution between 0 and 1

(2) Normal Distribution with mean 0 and standard deviation 0.03

Fitting data to the model (5 points)

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

In [135…

```
# 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.linalg.inv(xhat.dot(xhat.T)).dot(xhat.dot(y)) # please modify this line

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

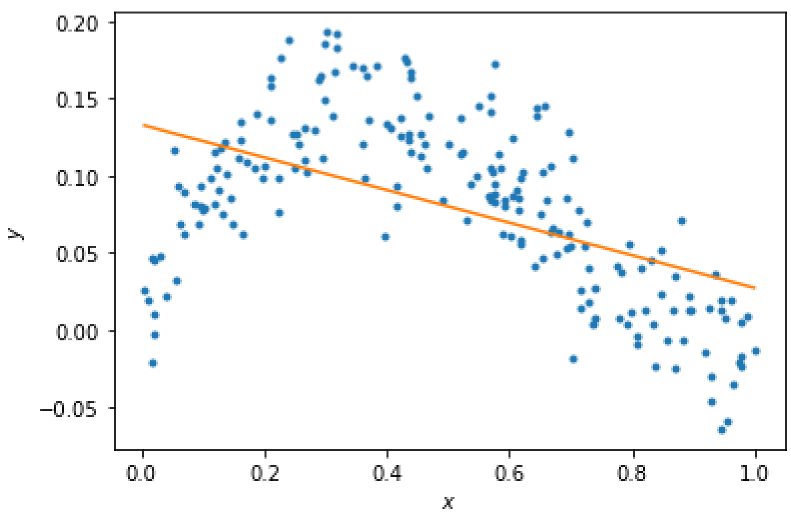
In [136…

```
# Plot the data and your model fit.
f = plt.figure()
ax = f.gca()
```

```python
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))
```

Out[136… [<matplotlib.lines.Line2D at 0x7fc344729100>]



QUESTIONS

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

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

ANSWERS

(1) The linear model underfits the data

(2) The model is very simple to capture the relationship between input and output, we need to increase complexity of model by adding higher order polynomial terms

Fitting data to the model (10 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.

In [137…

```python
N = 5
xhats = []
thetas = []

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


xhats = [np.vstack((x, np.ones_like(x))), np.vstack((x*x, x, np.ones_like(x))),
         np.vstack((x*x*x, x*x, x, np.ones_like(x))), np.vstack((x*x*x*x, x*x*x, x*x, x, np.ones_like(x))),
         np.vstack((x*x*x*x*x, x*x*x*x, x*x*x, x*x, x, np.ones_like(x)))]
thetas = [np.linalg.inv(xhat.dot(xhat.T)).dot(xhat.dot(y)) for xhat in xhats]
# 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.

pass

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

In [138…

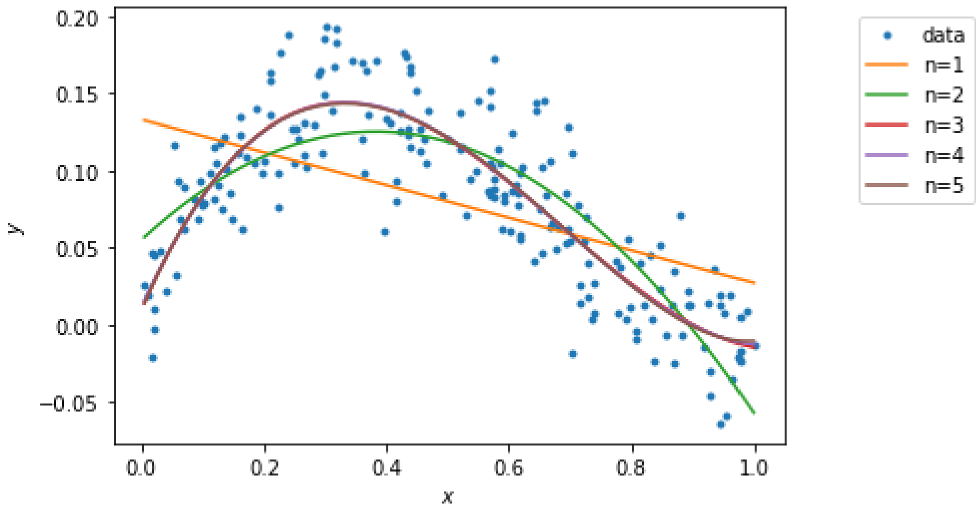
```python
# Plot the data
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')

# 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)
```



Calculating the training error (10 points)

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

$L(\theta) = \frac{1}{2}\sum_j (\hat{y}_j - y_j)^2$

In [139…

```
training_errors = []

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

training_errors = [0.5* np.sum(np.square(y-(thetas[i]@xhats[i]))) for i in range(N)]

# 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

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

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

```
Training errors are:
 [0.2379961088362701, 0.10924922209268531, 0.08169603801105371, 0.08165353735296985, 0.08161479195525291]
```

QUESTIONS

(1) Which polynomial model has the best training error?

(2) Why is this expected?

ANSWERS

(1) Polynomial model of degree 5

(2) Becuase it has more complexity and hance more degrees of freedom to fit the training data.

Generating new samples and validation error (5 points)

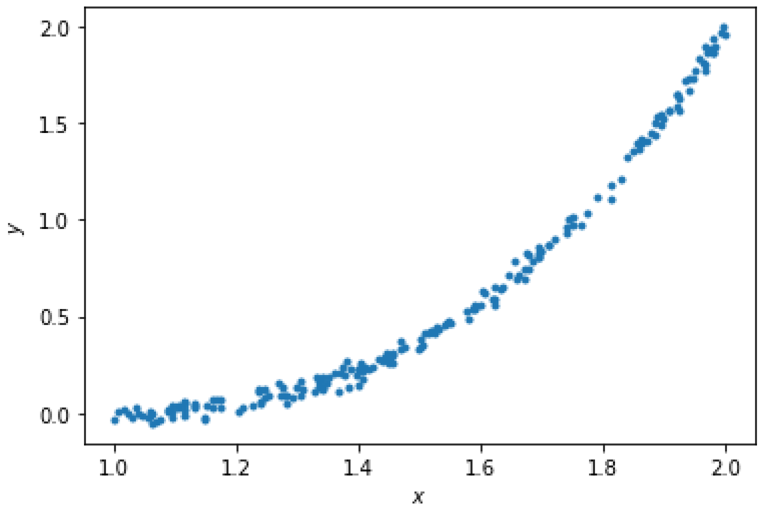
Here, we'll now generate new samples and calculate the validation error of polynomial models of orders 1 to 5.

In [140…

```
x = np.random.uniform(low=1, high=2, size=(num_train,))
y = x - 2*x**2 + 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('$y$')
```

Out[140…

```
Text(0, 0.5, '$y$')
```



In [141…

```
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)
```

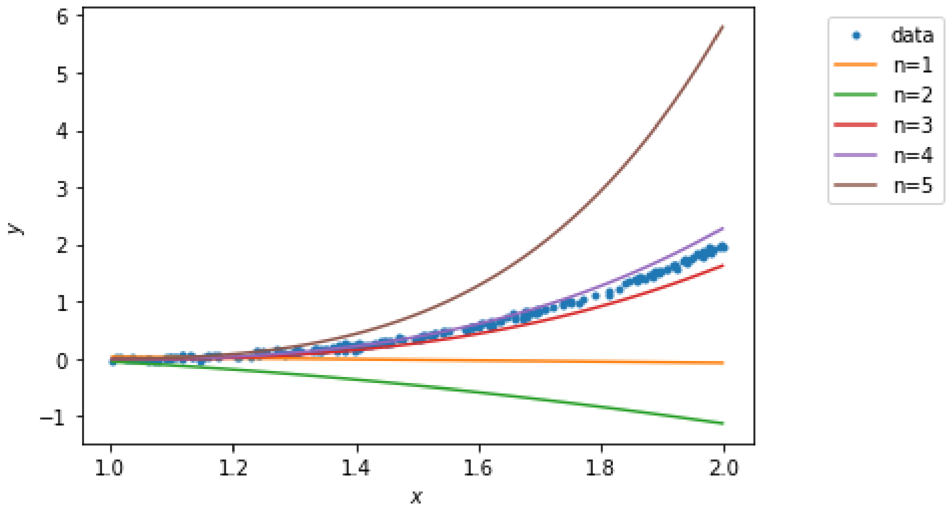
In [142…

```
# 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)
```



In [143…

```
validation_errors = []

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

validation_errors = [0.5* np.sum(np.square(y-(thetas[i]@xhats[i]))) for i in range(N)]
# GOAL: create a variable validation_errors, a list of 5 elements,
# where validation_errors[i] are the validation loss for the polynomial fit of order i+1.
pass

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

print ('Validation errors are: \n', validation_errors)
```

```
Validation errors are:
 [80.86165184550586, 213.19192445057962, 3.1256971082784704, 1.1870765198488837, 214.91021806189653]
```

QUESTIONS

(1) Which polynomial model has the best validation error?

(2) Why does the order-5 polynomial model not generalize well?

ANSWERS

(1) Polynomial model of degree 4

(2) It overfits the training data by capturing the noise in it and hence is not able to generalize well

In [ ]:
```

```