Margaret Capetz
147 HW1

1) Linear alg

a) Let Q be a real orthogonal matrix.

i) Show Q^T and Q^{-1} are also orthogonal.
To test if a matrix is orthogonal, see if it satisfies $AA^T = I$
Definition of orthogonal matrix A: $A^T = A^{-1}$.

Pf.
$(Q^{-1})^{-1} = Q$, $(Q^T)^T = Q$ by definition of inverse, transpose.

$(Q^T)^T = Q = (Q^{-1})^{-1} = (Q^T)^{-1}$
$\therefore (Q^T)^T = (Q^T)^{-1} \Rightarrow Q^T$ is orthogonal.

$(Q^{-1})^{-1} = Q = (Q^T)^T = (Q^{-1})^T$
$\therefore (Q^{-1})^{-1} = (Q^{-1})^T \Rightarrow Q^{-1}$ is orthogonal. □.

ii) Show that Q has eigenvalues with norm 1.

Pf.
By definition, Q is comprised of orthonormal vectors as columns & rows.
Thus the norms of each orthonormal vector, by definition, is 1.

Consider $Qx = \lambda x$, where λ is an eigenvalue and x is the corresponding eigenvector.

$$\begin{aligned}
\|Qx\|^2 &= \|\lambda x\|^2 \\
(Qx)^T(Qx) &= |\lambda|^2 \|x\|^2 \text{ by def of length} \\
x^T Q^T Q x &= |\lambda|^2 \|x\|^2 \text{ by transpose distribution} \\
x^T x &= |\lambda|^2 \|x\|^2 \text{ by def of orthonormal} \\
\|x\|^2 &= |\lambda|^2 \|x\|^2 \text{ by def of length} \\
1 &= |\lambda|^2 \\
1 &= \lambda
\end{aligned}$$

$\therefore$ all eigenvalues of Q have norm 1. □.

iii) Show the determinant of Q is either +1 or -1.

Pf.
Take $QQ^T = I$ by definition of orthogonal.
$\det(QQ^T) = \det I$ by taking the det of both sides
$\det(Q)\det(Q^T) = 1$
$(\det Q)^2 = 1$ since $\det(Q) = \det(Q^T)$
$\det Q = \pm 1$ square rooting both sides

$\therefore \det Q = \pm 1$. □.

length preserving transformation: $\|Qv\|_2^2 = \|v\|_2^2$ length preserved

iv) show that Q defines a length preserving transformation.

can ask on piazza

Pf.

$$\begin{aligned} &\|Qv\|_2^2 \\ &= (Qv)^T(Qv) \\ &= v^T Q^T Q v \\ &= v^T v \\ &= \|v\|_2^2 \end{aligned}$$

$\therefore$ Q transformation preserves length. $\square$.

b) Let A be a matrix.

singular vectors

for $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, $A = U\Sigma V^T$

singular values

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

eigendecomp: A square, diag

SVD basics: A not necessarily square, AA^T always symmetric

$A = U\Sigma V^T$

left right

$AA^T = U \Lambda U^T$

$A^TA = V \Lambda V^T$

Λ: eigenvalues; V: orthogonal

$$\begin{aligned} AA^T &= U\Sigma V^T (U\Sigma V^T)^T \\ &= U\Sigma V^T V^{TT} \Sigma^T U^T \\ &= U\Sigma V^T V \Sigma^T U^T \\ &= U \underbrace{\Sigma\Sigma^T} U^T \end{aligned}$$

$$\Sigma\Sigma^T = \sigma^2$$

$$= U\sigma^2 U^T$$

> U is the eigenvector of AA^T.
>
> V is the eigenvector of A^TA.

ii) What about values of A & eigenvalues of AA^T, A^TA

singular values (σ) are the square of the eigenvalues (λ), for both AA^T and A^TA.

$\sigma^2 = \lambda$

see part (i) for work.

c) True or false.

i) Every linear operator (linear map.) in an n-dimensional vector space has n distinct eigenvalues.
False, should be at most n distinct eigenvalues.

ii) A non-zero sum of two eigenvectors of a matrix A is an eigenvector.
False, only if they share an eigenvalue.

iii) If a matrix A has the positive semidefinite property, ie. $x^T A x \geq 0 \ \forall x$, then its eigenvalues must be non-negative.
True. Similar to discussion question.

iv) The rank of a matrix can exceed the number of distinct non-zero eigenvalues.
True, if there is repetition of distinct eigenvalues. 1 eigenvalue for I matrix.

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

2) Probability

a) Jar of coins. 'H50' coins : $\mathbb{P}(H) = 0.5$
'H60' coins : $\mathbb{P}(H) = 0.6$

i) Flip a coin, lands tails. Posterior probability it is an H50 coin?

$\mathbb{P}(H50) = 0.5$
$\mathbb{P}(H60) = 0.5$
$\mathbb{P}(H \mid H50) = 0.5$
$\mathbb{P}(H \mid H60) = 0.6$

$$\mathbb{P}(T) = \mathbb{P}(T \cap H50) + \mathbb{P}(T \cap H60)$$
$$= \mathbb{P}(H50)\,\mathbb{P}(T \mid H50) + \mathbb{P}(H60)\,\mathbb{P}(T \mid H60)$$
$$= 0.5 \cdot 0.5 + 0.5 \cdot 0.4$$
$$= 0.45$$

$$\mathbb{P}(H50 \mid T) = \frac{\mathbb{P}(H50)\,\mathbb{P}(T \mid H50)}{\mathbb{P}(T)} = \frac{0.5 \cdot 0.5}{0.45} = \boxed{0.556}$$

ii) Flip a coin 4 time, landing THHH.
How likely is the coin to be type H50?

$$\mathbb{P}(THHH) = \mathbb{P}(THHH \cap H50) + \mathbb{P}(THHH \cap H60)$$
$$= \mathbb{P}(H50)\,\mathbb{P}(THHH \mid H50) + \mathbb{P}(H60)\,\mathbb{P}(THHH \mid H60)$$
$$= 0.5 \cdot (0.5)^4 + 0.5 \cdot (0.4 \cdot 0.6^3)$$
$$= 0.07445$$

$\mathbb{P}(H50 \mid \mathbb{P}(THHH)$

$$= \frac{\mathbb{P}(THHH \mid H50)\,\mathbb{P}(H50)}{\mathbb{P}(THHH)} = \frac{(0.5)^5}{0.07445} = \boxed{0.4197}$$

iii) New jar, with H50, H55, H60.
Take 1 coin, flip it 10 times. It lands heads 9 times.
How likely is the coin of each type?
Let's call the event in question to be A.

Note:
$\binom{10}{9}$ = 10 ways to have 9 heads among 10 flips.

$$\mathbb{P}(A) = \mathbb{P}(A \cap H50) + \mathbb{P}(A \cap H55) + \mathbb{P}(A \cap H60)$$

$$= \mathbb{P}(H50)\mathbb{P}(A|H50) + \mathbb{P}(H55)\mathbb{P}(A|H55) + \mathbb{P}(H60)\mathbb{P}(A|H60)$$

$$= \tfrac{1}{3} \cdot 10 \cdot (\tfrac{1}{2})^{10} + \tfrac{1}{3} \cdot 10 \cdot \tfrac{45}{100} \cdot \left(\tfrac{55}{100}\right)^9 + \tfrac{1}{3} \cdot 10 \cdot \tfrac{40}{100} \cdot \left(\tfrac{60}{100}\right)^9$$

$$= 0.02366$$

$$\mathbb{P}(H50|A) = \frac{\mathbb{P}(A|H50)\mathbb{P}(H50)}{\mathbb{P}(A)} = \frac{\frac{1}{3} \cdot 10 \cdot (\frac{1}{2})^{10}}{0.02366} = \boxed{0.1379}$$

$$\mathbb{P}(H55|A) = \frac{\mathbb{P}(A|H55)\mathbb{P}(H55)}{\mathbb{P}(A)} = \frac{\frac{1}{3} \cdot 10 \cdot \frac{45}{100} \cdot (\frac{55}{100})^9}{0.02366} = \boxed{0.2927}$$

$$\mathbb{P}(H60|A) = \frac{\mathbb{P}(A|H60)\mathbb{P}(H60)}{\mathbb{P}(A)} = \frac{\frac{1}{3} \cdot 10 \cdot \frac{40}{100} \cdot (\frac{60}{100})^9}{0.02366} = \boxed{0.5694}$$

b) UCLA students:

15% science	90% science like lecture
21% healthcare	18% healthcare "
24% liberal arts	0% liberal arts "
40% Engineering	10% engr "

Suppose a student liked lec. What prob they are sci?

Want: $\mathbb{P}(\text{science} \mid \text{liked lecture})$, or $\mathbb{P}(S \mid L)$

Define: S as science
H as healthcare
A as liberal arts
E as engr
L as liked lecture

$$\mathbb{P}(L) = \mathbb{P}(L \cap S) + \mathbb{P}(L \cap H) + \mathbb{P}(L \cap A) + \mathbb{P}(L \cap E)$$
$$= \mathbb{P}(S)\mathbb{P}(L|S) + \mathbb{P}(H)\mathbb{P}(L|H) + \mathbb{P}(A)\mathbb{P}(L|A) + \mathbb{P}(E)\mathbb{P}(L|E)$$
$$= 0.15 \cdot 0.9 + 0.21 \cdot 0.18 + 0.24 \cdot 0 + 0.4 \cdot 0.1$$
$$= 0.2128$$

$$\mathbb{P}(S|L) = \frac{\mathbb{P}(L|S)\mathbb{P}(S)}{\mathbb{P}(L)} = \frac{0.9 \cdot 0.15}{0.2128} = \boxed{0.634}$$

c) Pregnancy test: what is the probability a woman is pregant given a pos. test? Give an explanation.

Define + : positive
P : pregnant

$\mathbb{P}(+|P) = 0.99$
$\mathbb{P}(+|P') = 0.1$
$\mathbb{P}(P) = 0.01$
$\mathbb{P}(P') = 0.99$

$$\mathbb{P}(+) = \mathbb{P}(+ \cap P) + \mathbb{P}(+ \cap P')$$
$$= \mathbb{P}(P)\mathbb{P}(+|P) + \mathbb{P}(P')\mathbb{P}(+|P')$$
$$= 0.01 \cdot 0.99 + 0.99 \cdot 0.1$$
$$= 0.1089$$

$$\mathbb{P}(P|+) = \frac{\mathbb{P}(+|P)\mathbb{P}(P)}{\mathbb{P}(+)} = \frac{0.99 \cdot 0.01}{0.1089} = \boxed{0.09}$$

There is a 9% probability a woman is pregnant given a positive pregnancy test. This is intuitive since the rate of false positives is high at 10%.
The majority of women who take the test receive a false positive, because very little women are actually pregnant.

d) Let $x_1, x_2, \ldots, x_n$ be identically distributed random vars.
A random vector, x, is defined as

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

what is $\mathbb{E}(Ax+b)$ in terms of $\mathbb{E}(x)$, given that A, b are deterministic?

↳ at every single point, you know the exact value

$\mathbb{E}$ is expectation $\mathbb{E}(x)$ is the mean of x

$y = Wx + b$

$\mathbb{E}(b) = b$, $\mathbb{E}(A) = A$, $\mathbb{E}$ is linear

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

$$\begin{bmatrix} a_{11} & & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} =$$

show it is linearly separate

Pf.
Let $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$

Then $$Ax = \underbrace{\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}}_{m \times n} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}}_{n \times 1} = \underbrace{\begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \end{bmatrix}}_{m \times 1}$$

Then $$Ax + b = \underbrace{\begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \end{bmatrix}}_{m \times 1} + \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

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

Thus $Ax + b$ is linearly separable.
$\therefore \mathbb{E}(Ax+b) = \mathbb{E}(A)\,\mathbb{E}(x) + \mathbb{E}(b)$
$= A\,\mathbb{E}(x) + b$ as A, b are deterministic.

Thus $\mathbb{E}(Ax+b) = A\,\mathbb{E}(x) + b$. □

e) Let $\text{cov}(x) = \mathbb{E}((x - \mathbb{E}(x))(x - \mathbb{E}(x))^T)$
What is $\text{cov}(Ax+b)$ in terms of $\text{cov}(x)$ given A, b are deterministic?

$$\text{cov}(Ax+b) = \mathbb{E}((Ax+b - \mathbb{E}(Ax+b))(Ax+b - \mathbb{E}(Ax+b))^T)$$
$$= \mathbb{E}((Ax+b - (A\,\mathbb{E}(x)+b))(Ax+b - (A\,\mathbb{E}(x)+b))^T)$$

$$= \mathbb{E}\left((Ax \cancel{+b} - A\mathbb{E}(x) \cancel{-b})(Ax \cancel{+b} - A\mathbb{E}(x) \cancel{-b})^T\right)$$
$$= \mathbb{E}\left(A(x-\mathbb{E}(x))(A(x-\mathbb{E}(x))^T\right)$$
$$= A \cdot \mathbb{E}\left((x-\mathbb{E}(x))(x-\mathbb{E}(x))^T\right) \cdot A^T$$
$$= A\ cov(x)\ A^T$$

Ref me #

3) Multivariate derivatives.

a) Let $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times m}$. What is $\nabla_x x^T A y$?

$\nabla_x x^T A y = \nabla_x x^T (Ay) = Ay$ by the matrix cookbook

$(\nabla_x x^T A = A)$

alternatively,

$$\frac{\partial f(x,y)}{\partial x_1} = a_{11}y_1 + \sum_{j=2}^{m} a_{1j}y_j + 0$$
$$= \sum_{j=1}^{m} a_{1j}y_j$$
$$= \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$
$$= Ay$$

same!

$$f(x,y) = x^T A y = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} \cdot x_i \cdot y_j$$

$i=1, j=1$ $\quad$ $i=1, j \neq 1$ $\quad$ $i \neq 1, j=1$

$$\frac{\partial a_{11}x_1y_1}{\partial x_1} = a_{11}y_1 \qquad \frac{\partial\left(\sum_{j=2}^{m} a_{1j} \cdot x_1 \cdot y_j\right)}{\partial x_1} = \sum_{j=2}^{m} a_{1j} \cdot y_j \qquad \frac{\partial \sum_{i=2}^{n}(a_{i1} \cdot x_i \cdot y_1)}{\partial x_1} = 0$$

hint: see COOKBOOK

1) derivative of a vector w.r.t. a vector
$\nabla_x x^T A = A$, $x \in \mathbb{R}^n$
$\nabla_x Ax = A^T \rightarrow \nabla_x \theta^T x = 0$
$\nabla_x b^T x = b$
$\nabla_x x^T A x = Ax + A^T x = 2Ax$ if symmetric (A is)

2) derivative of a scalar w.r.t. a matrix
$\nabla_A x^T A y = xy^T$

$\nabla_A tr(AB) = B^T$

hint: $\nabla_x x^T A x = Ax + A^T x$
A could be $B^T B$, $A + B$, $\cdots$, $W^T W$, $\cdots$ substitution

a) $\nabla_x x^T A y = \nabla_x x^T (Ay) = Ay$
b) $\nabla_y x^T A y = \nabla_y (x^T A) y = (x^T A)^T = A^T x$

b)

Let $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times m}$. What is $\nabla_y x^T A y$?

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

$(\nabla_x Ax = A)$

alternatively,

$$\frac{\partial f(x,y)}{\partial y_1} = a_{11}x_1 + \sum_{i=2}^{n} a_{i1} \cdot x_i + 0$$
$$= \sum_{i=1}^{n} a_{i1} \cdot x_i$$
$$= \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}^T \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$$
$$= A^T x$$

same!

$$f(x,y) = x^T A y = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} \cdot x_i \cdot y_j$$

$i \neq 1, j=1$ $\quad$ $i=1, j=1$ $\quad$ $i=1, j \neq 1$

$$\frac{\partial \sum_{i=2}^{n}(a_{i1} \cdot x_i \cdot y_1)}{\partial y_1} = \sum_{i=2}^{n} a_{i1} \cdot x_i \qquad \frac{\partial a_{11}x_1y_1}{\partial y_1} = a_{11}x_1 \qquad \frac{\partial\left(\sum_{j=2}^{m} a_{1j} \cdot x_1 \cdot y_j\right)}{\partial y_1} = 0$$

c) What is $\nabla_A x^T A y$?

By the matrix cookbook,

$\nabla_A x^T A y = x y^T$

d) Let $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$, $f = x^T A x + b^T x$. What is $\nabla_x f$?

$\nabla_x f = \nabla_x x^T A x + \nabla_x b^T x$, we have $\nabla_x x^T A x = Ax + A^T x$ and $\nabla_x b^T x = b$, so

$= Ax + A^T x + b$ using substitutions we showed in class.

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

$\nabla_A tr(AB) = B^T$ by the matrix cookbook.

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

$\nabla_A tr(BA + A^T B + A^2 B)$

$= \nabla_A tr\,AB + \nabla_A tr\,A^T B + \nabla_A tr\,A^2 B \quad (trAB = tr\,BA)$

$= B^T + B + (AB + BA)^T$ by the matrix cookbook.

g) Let $A, B \in \mathbb{R}^{n \times n}$, $f = \|A + \lambda B\|_F^2$. What is $\nabla_A f$?

$\nabla_A \|A + \lambda B\|_F^2 = \nabla_A Tr((A + \lambda B)(A + \lambda B)^T)$

$= \nabla_A Tr((A + \lambda B)(A^T + B^T \lambda^T))$

$= \nabla_A Tr(AA^T + AB^T\lambda^T + \lambda B A^T + \lambda B B^T \lambda^T)$

$= \nabla_A Tr(A\underline{A^T}) + \nabla_A Tr(A\underline{B^T\lambda^T}) + \nabla_A Tr(\underline{\lambda B}A^T) + \nabla_A (\lambda B B^T \lambda^T)$

$= 2A + (B^T\lambda^T)^T + \lambda B + 0$

$= 2A + \lambda B + \lambda B + 0 \quad = 2A + 2\lambda B$

Matrix cookbook:

$\frac{\partial}{\partial X} \|X\|_F^2 = \frac{\partial}{\partial X} Tr(XX^H) = 2X$

$\|A\|_F^2 = Tr(AA^H) = Tr(AA^T)$ for $A \in \mathbb{R}^{m \times n}$ (H: conjugate transpose)

Hints:
F: Frobenius
For real matrices, $A \in \mathbb{R}^{m \times n}$, the conjugate transpose is just the transpose, $A^H = A^T$

4) Deriving least-squares with matrix derivatives.

In least squares, we seek to estimate multivariate output y via model

$\hat{y} = Wx$

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

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

Derive the optimal W.

Where W is a matrix, and for each example in the training set, both $x^{(i)}$ and $y^{(i)}$ $\forall i = 1, \dots, n$ are vectors.

Hint:

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

Take $\frac{1}{2}\sum_{i=1}^{n} \| y^{(i)} - W x^{(i)} \|^2$ * $y^{(i)} \in \mathbb{R}^{m\times 1}$, $x^{(i)} \in \mathbb{R}^{k\times 1}$, $W \in \mathbb{R}^{m\times k}$

"Vectorization" $= \frac{1}{2}\sum_{i=1}^{n} (y^{(i)} - W x^{(i)})^T (y^{(i)} - W x^{(i)})$ by definition of 2-norm

change formula to match dims

$$= \frac{1}{2}\left(\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix} - \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix} W^T \right)^T \left(\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix} - \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix} W^T \right)$$

$Y \in \mathbb{R}^{n\times m}$, $X \in \mathbb{R}^{n\times k}$, $XW^T \in \mathbb{R}^{n\times m}$ ($n\times k$, $k\times m$)

$= \frac{1}{2}\| Y - XW^T \|_F^2$

$= \frac{1}{2} Tr((Y - XW^T)(Y - XW^T)^T)$

$= \frac{1}{2} Tr((Y - XW^T)(Y^T - WX^T))$

$= \frac{1}{2} Tr(YY^T - YWX^T - XW^TY^T + XW^TWX^T)$

$= \frac{1}{2} Tr(YY^T - (XW^TY^T)^T - XW^TY^T + XW^TWX^T)$ (same since scalar^T are equal)

$= \frac{1}{2} Tr(YY^T - 2YWX^T + XW^TWX^T)$, now take the derivative

$\frac{\partial}{\partial W}\left(\frac{1}{2} Tr(YY^T - 2YWX^T + XW^TWX^T)\right)$

$= \frac{1}{2}\left(\frac{\partial}{\partial W} Tr(YY^T) - 2\frac{\partial}{\partial W} Tr(YWX^T) + \frac{\partial}{\partial W} Tr(XW^TWX^T)\right)$ ($Tr(AB) = Tr(BA)$; $A = XW^T$, $B = WX^T$)

$= \frac{1}{2}(0 - 2\frac{\partial}{\partial W} Tr(WX^TY) + \frac{\partial}{\partial W} Tr(WX^TXW^T))$

$= \frac{1}{2}(-2(X^TY)^T + W(X^TX)^T + W(X^TX))$

$= -(X^TY)^T + \frac{1}{2}(W(X^TX)^T + W(X^TX))$

$= -(X^TY)^T + \frac{1}{2}W(X^TX + X^TX)$

$= -(X^TY)^T + W(X^TX)$

Setting this equal to zero,

$0 = -(X^TY)^T + W(X^TX)$

$(X^TY)^T = W(X^TX)$

$$\boxed{W = (X^TY)^T(X^TX^T)^{-1}}$$

See below

5) Regularized least-squares

In lecture, we worked through the regularized least squares problem

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

Regularization addresses overfitting, we'll consider ridge regularization aka regularized least-squares problem.

Solve the following optimization problem:

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

where λ is a tunable regularization param.
Observe we are seeking the least-squares sol w/ the smaller 2-norm.
Derive the solution, i.e. find θ^*

hints:

For $\sum_{i=1}^{N} (y^{(i)} - \theta^T \hat{x}^{(i)})^2$ ⟶ express as a vector

⇓

not $y^{(i)}$ is a scalar ∴ $Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$ is a vector

for the second term $\frac{\lambda}{2}\|\theta\|_2^2$, try expressing it as the product of θ and θ^T

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

$y^{(i)} \in \mathbb{R}^{m \times 1}$

$\hat{x}^{(i)} \in \mathbb{R}^{n \times 1}$

$\theta \in \mathbb{R}^{N \times N}$

↓ by vectorization,

$$= \frac{1}{2} \left(\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix} - \theta^T \begin{bmatrix} \hat{x}^{(1)} \\ \hat{x}^{(2)} \\ \vdots \\ \hat{x}^{(N)} \end{bmatrix} \right)^2 + \frac{\lambda}{2} \|\theta\|_2^2$$

$Y \in \mathbb{R}^{N \times n}$ $\quad X \in \mathbb{R}^{N \times n}$

$\theta^T X \in \mathbb{R}^{N \times n}$

$N \times N \quad N \times n$

the square of a matrix can be expressed as its innerproduct

$$= \frac{1}{2} (Y - \theta^T X)^2 + \frac{\lambda}{2} \|\theta\|_2^2$$

$$= \frac{1}{2} (Y - \theta^T X)^T (Y - \theta^T X) + \frac{\lambda}{2} \|\theta\|_2^2$$

$$= \frac{1}{2} (Y^T - X^T \theta)(Y - \theta^T X) + \frac{\lambda}{2} \|\theta\|_2^2$$

$$= \frac{1}{2} (Y^T Y - Y^T \theta^T X - X^T \theta Y + X^T \theta \theta^T X) + \frac{\lambda}{2} \|\theta\|_2^2$$

then, take the derivative w.r.t. θ

$$\frac{\partial}{\partial \theta} \left(\frac{1}{2} (Y^T Y - Y^T \theta^T X - X^T \theta Y + \theta^T X X^T \theta) + \frac{\lambda}{2} \|\theta\|_2^2 \right)$$

two-norm

$$\frac{\partial \|x\|_2^2}{\partial x} = \frac{\partial \|x^T x\|_2}{\partial x} = 2x$$

$= \frac{1}{2}\left(0 - \mathbf{Y}\mathbf{X}^T - \mathbf{X}\mathbf{Y}^T + (\mathbf{X}\mathbf{X}^T)(\mathbf{X}\mathbf{X}^T)^T + \lambda 2\theta \right)$

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

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

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

Try Again:

$y^{(i)} \in \mathbb{R}$ ← scalar!

$\theta \in \mathbb{R}^{m \times 1}$

$x \in \mathbb{R}^{m \times 1}$

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

↓ by vectorization

$$= \frac{1}{2} \left(\begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix} - \theta^T \begin{bmatrix} \hat{x}^{(1)} \\ \vdots \\ \hat{x}^{(n)} \end{bmatrix} \right)^2 + \frac{\lambda}{2} \|\theta\|_2^2$$

$\mathbf{Y} \in \mathbb{R}^{1 \times n}$ $\quad \theta^T \in \mathbb{R}^{1 \times m}$

$\mathbf{X} \in \mathbb{R}^{m \times n}$

$\theta^T \mathbf{X} \in \mathbb{R}^{1 \times n}$

$= \frac{1}{2} \left(\mathbf{Y} - \mathbf{X}\theta \right)^T \left(\mathbf{Y} - \mathbf{X}\theta \right) + \frac{\lambda}{2} \|\theta\|_2^2$

$= \frac{1}{2} \left((\mathbf{Y}^T - \theta^T\mathbf{X}^T)(\mathbf{Y} - \mathbf{X}\theta) + \lambda \|\theta\|_2^2 \right)$

$= \frac{1}{2} \left(\mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{X}\theta - \theta^T\mathbf{X}^T\mathbf{Y} + \theta^T\mathbf{X}^T\mathbf{X}\theta + \lambda \|\theta\|_2^2 \right)$

$\nabla_x x^T A x = Ax + A^T x$

$\nabla_x Ax = A^T$

$\nabla_x x^T A = A$

Now, let's take the derivative w.r.t. θ

$$\frac{\partial}{\partial\theta} \left(\frac{1}{2} \left(\mathbf{Y}^T\mathbf{Y} - \underline{\mathbf{Y}^T\mathbf{X}\theta} - \underline{\theta^T\mathbf{X}^T\mathbf{Y}} + \theta^T\mathbf{X}^T\mathbf{X}\theta + \lambda \|\theta\|_2^2 \right) \right)$$

scalars - so can combine as

$$\mathbf{Y}\mathbf{X}\theta^T = \theta^T\mathbf{X}^T\mathbf{Y}$$

$$= \frac{1}{2} \frac{\partial}{\partial\theta} \left(\mathbf{Y}^T\mathbf{Y} - 2\underline{\mathbf{Y}^T\mathbf{X}\theta} + \theta^T\underline{\mathbf{X}^T\mathbf{X}}\theta + \lambda \|\theta\|_2^2 \right)$$

$= \frac{1}{2}\left(0 - 2\mathbf{X}^T\mathbf{Y} + (\mathbf{X}^T\mathbf{X} + (\mathbf{X}^T\mathbf{X})^T)\theta + \lambda 2\theta \right)$

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

$= -\mathbf{X}^T\mathbf{Y} + \mathbf{X}^T\mathbf{X}\theta + \lambda\mathbf{I}\theta$

Now, setting this equal to zero, we can solve for θ

$0 = -\mathbf{X}^T\mathbf{Y} + \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I} \right)\theta$

$\mathbf{X}^T\mathbf{Y} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I} \right)\theta$

$$\therefore \boxed{\theta^* = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I} \right)^{-1}\left(\mathbf{X}^T\mathbf{Y} \right)}.$$

$= \frac{1}{2}\left(0 - (\mathbf{Y}\mathbf{X})^T - (\mathbf{X}^T\mathbf{Y}) + (\mathbf{X}^T\mathbf{X})^T(\mathbf{X}^T\mathbf{X}) + \lambda 2\theta \right)$

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

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

Solving for θ when setting this equal to zero,

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

4) Take $\frac{1}{2}\sum_{i=1}^{n} \| y^{(i)} - Wx^{(i)} \|^2$ $\quad y^{(i)} \in \mathbb{R}^{m\times 1}$, $x^{(i)} \in \mathbb{R}^{n\times 1}$

$= \frac{1}{2}\sum_{i=1}^{n} (y^{(i)} - Wx^{(i)})^T (y^{(i)} - Wx^{(i)})$ ↑ flip transpose order for scalar

"vectorization" ↓

$= \frac{1}{2}\left(\begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix} - \begin{bmatrix} x^{(1)} \\ \vdots \\ x^{(n)} \end{bmatrix}^T W^T \right)^T \left(\begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix} - \begin{bmatrix} x^{(1)} \\ \vdots \\ x^{(n)} \end{bmatrix} W^T \right)$

$= \frac{1}{2} \| Y - X^T W^T \|_F^2$

$= \frac{1}{2} \mathrm{Tr}((Y - X^T W^T)(Y - X^T W^T)^T)$

$= \frac{1}{2} \mathrm{Tr}((Y - X^T W^T)(Y^T - WX))$

$= \frac{1}{2} \mathrm{Tr}(YY^T - \underbrace{Y^T W X - X^T W^T Y}_{\text{same since scalar}^T \text{ are equal}} + X^T W^T W X)$

$= \frac{1}{2} \mathrm{Tr}(YY^T - 2Y^T W X + X^T W^T W X)$

Now take the derivative

$\frac{\partial}{\partial W}\left(\frac{1}{2} \mathrm{Tr}(YY^T - 2Y^T W X + X^T W^T W X) \right)$

$= 0 - \frac{\partial}{\partial W}(Y^T W X) + \frac{1}{2}\frac{\partial}{\partial W}(X^T W^T W X)$

$= -\frac{\partial}{\partial W}(W \underline{X Y^T}) + \frac{1}{2}\frac{\partial}{\partial W}(W \underline{X X^T} W)$

$= -(XY^T)^T + \frac{1}{2}(W(XX^T)^T + W(XX^T))$

$= -YX^T + W(XX^T)$

$\boxed{W = YX^T(XX^T)^{-1}}$

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 2023, Prof. J.C. Kao, TAs: T.M, P.L, R.G, K.K, N.V, S.R, S.P, M.E

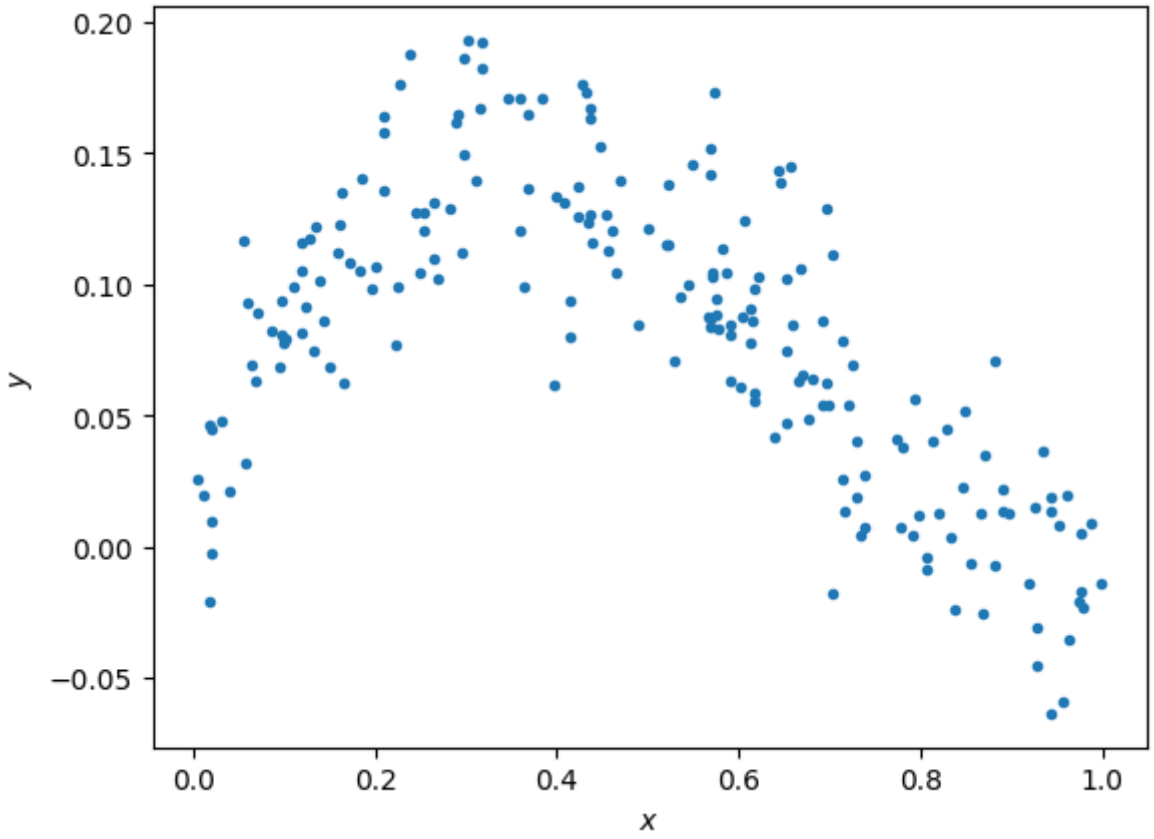
In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt

#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 [ ]:
```
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[ ]:
```
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 probability distribution.

(2) Normal (Gaussian) distribution.

Fitting data to the model (5 points)

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

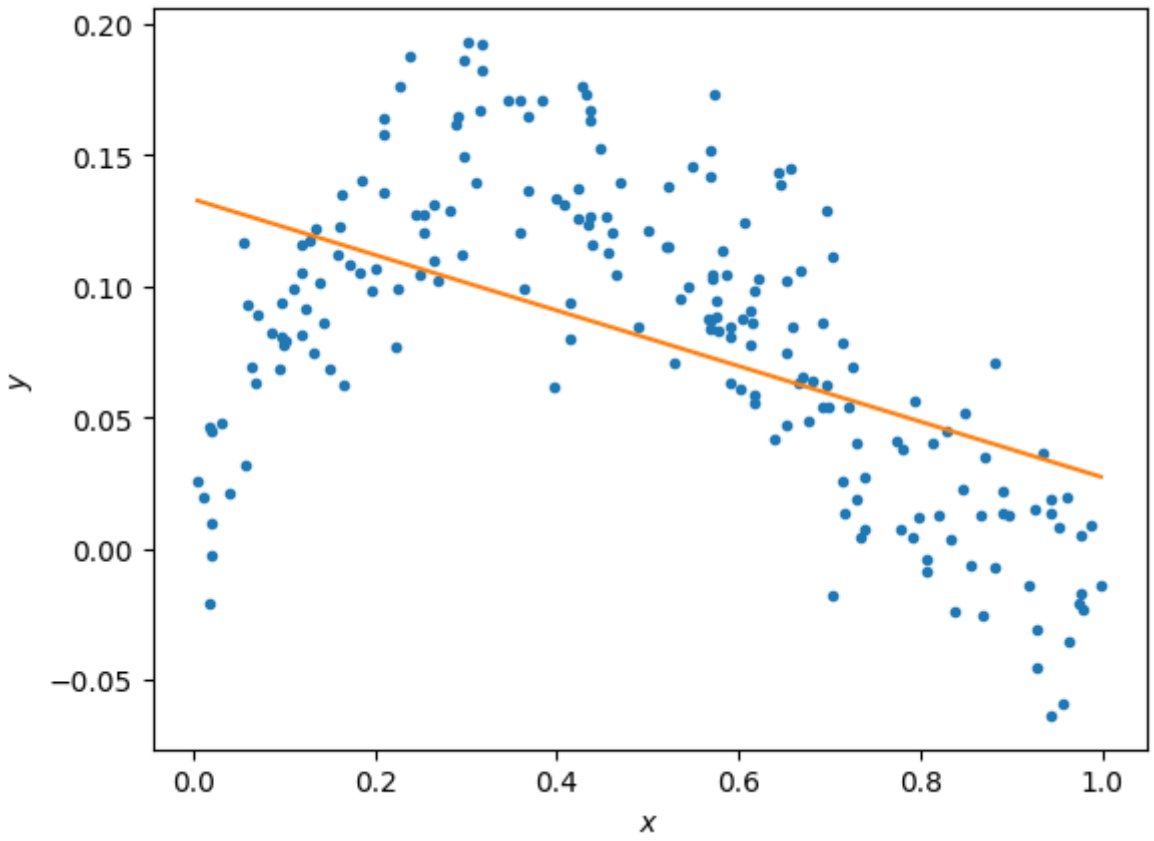
In [ ]:
```
# xhat = (x, 1)
xhat = np.vstack((x, np.ones_like(x)))

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

theta = np.zeros(2) # please modify this line
```

```
theta = ((np.linalg.inv((xhat).dot(xhat.T)).dot(xhat.dot(y)))).T
# print((xhat.dot(y)))
# print(xhat.shape, y.shape)
print(theta)

# use the transpose of what we got in class because in class x was horizontally
# try to match dims

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

```
[-0.10599633  0.13315817]
```

In [ ]:
```
# 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))
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x12c8a99f0>]
```



QUESTIONS

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

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

ANSWERS

(1) Underfit.

(2) Change the order of the polynomial, as a higher order polynomial model will better fit the data due to its curvature.

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.

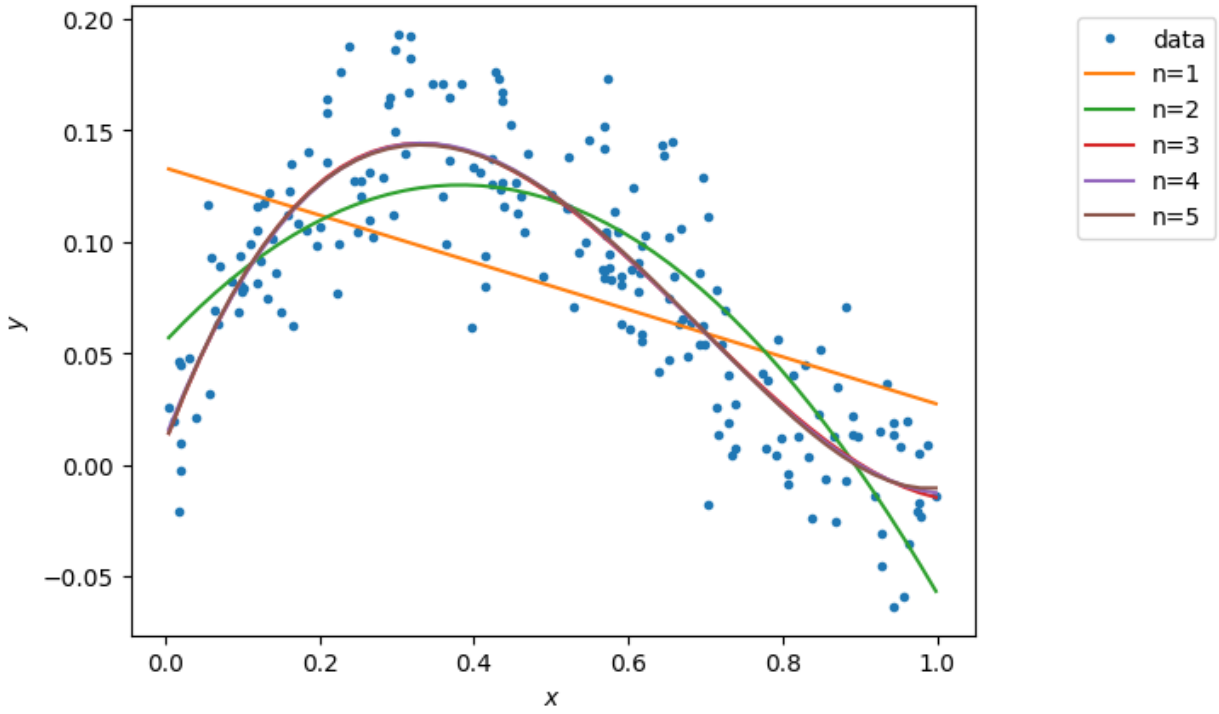
In [ ]:
```
N = 5
xhats = []
thetas = []

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

# GOAL: create a variable thetas.
# thetas is a list, where theta[i] are the model parameters for the polynomial
#   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
#   ... etc.

# hint: add to the vstack

# vstack xhats with x^3, will add to the back
# vstack(x^n, xhat)

# 2 then 3 then 4 then 5 using a loop

xhats.append(xhat)
temp = np.zeros(2)
temp = ((np.linalg.inv((xhat).dot(xhat.T)).dot(xhat.dot(y)))).T
thetas.append(temp)

for i in range(1, N):
    xhat = np.vstack((x**(i+1), xhat))
    xhats.append(xhat)
    thetas.append(((np.linalg.inv((xhat).dot(xhat.T)).dot(xhat.dot(y)))).T)

    print(i, thetas[i])
    # print("xhat shape", xhats.shape)
    print("theta shape", thetas[i].shape)

pass

# 0 : x, 1
# 1: x^2, x, 1
# 2: x^3, x^2, x, 1
# 3: x^4, x^3, x^2, x, 1
```

```
# 4: x^5, x^4, x^3, x^2, x, 1

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

```
1 [-0.48023061  0.36743967  0.05521084]
theta shape (3,)
2 [ 0.8843808  -1.82077417  0.91178032  0.00979068]
theta shape (4,)
3 [ 0.14080037  0.60466289 -1.64250929  0.87250485  0.01175321]
theta shape (5,)
4 [ 0.52432592 -1.164568    1.76052438 -2.07430275  0.93373916  0.009716  ]
theta shape (6,)
```

In [ ]:

```
# 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 (5 points)

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

In [ ]:
```
# ==================== #
# 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 orde
training_errors = [0] * 5
# MSE

for i in range(5):
    training_errors[i] = 0.5*(y.T.dot(y) - 2*(y.T).dot(xhats[i].T).dot(thetas[i


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

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

```
Training errors are:
 [0.23799610883626965, 0.10924922209268251, 0.08169603801102243, 0.08165353735
294645, 0.08161479195520172]
```

QUESTIONS

(1) What polynomial has the best training error?

(2) Why is this expected?

ANSWERS

(1) Polynomial with degree 5.

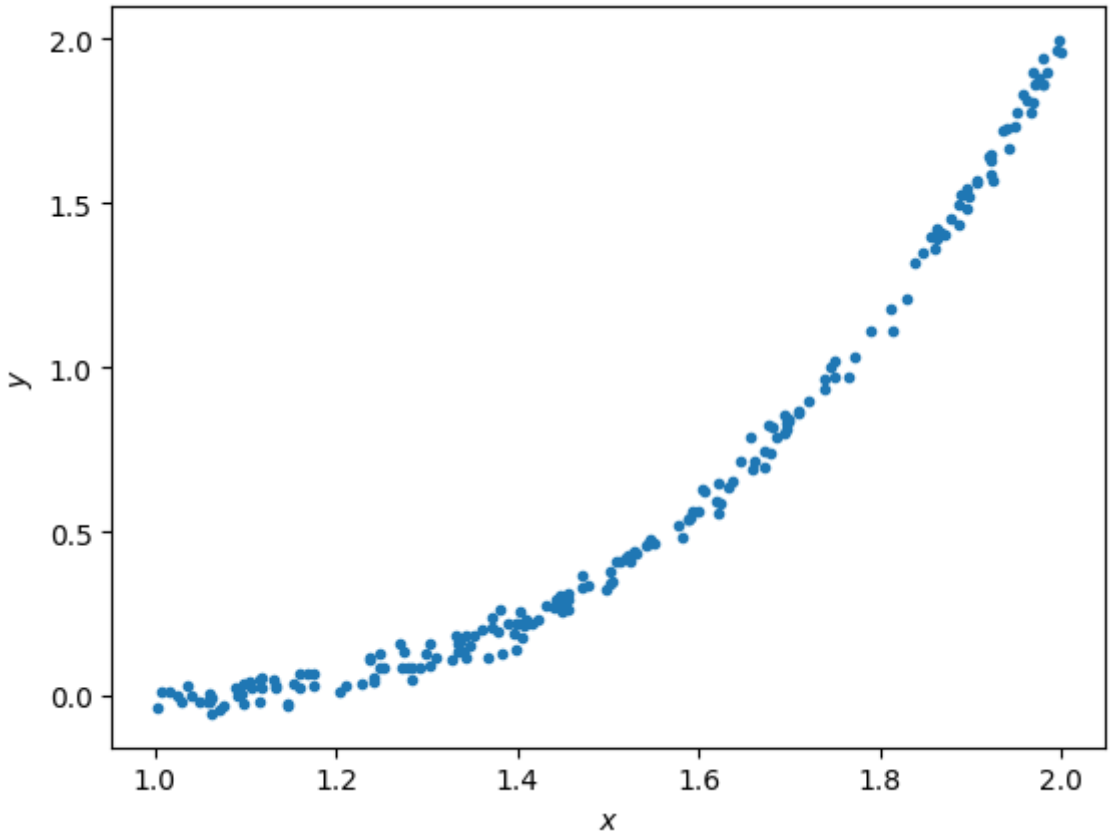
(2) As the polynomial order increases, the model can better fit the data, eventually passing through each point to make the training data equal to zero. However, this may lead to overfitting.

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.

In [ ]:
```
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('$y$')
```

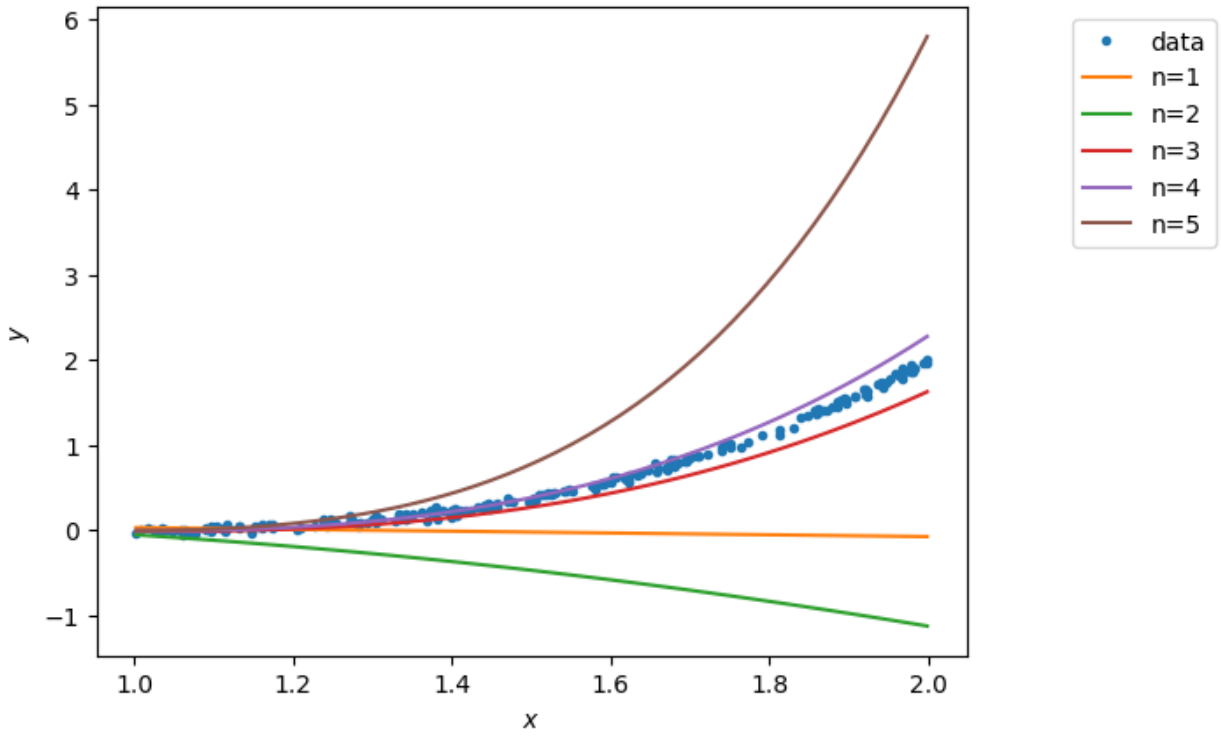
Out[ ]:
```
Text(0, 0.5, '$y$')
```



In [ ]:
```
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 [ ]:
```
# 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 [ ]:
```
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

testing_errors = [0] * 5
# MSE

for i in range(5):
    testing_errors[i] = 0.5*(y.T.dot(y) - 2*(y.T).dot(xhats[i].T).dot(thetas[i]

pass

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

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

```
Testing errors are:
 [80.86165184550593, 213.19192445058508, 3.125697108313929, 1.187076519555475
3, 214.91021831758638]
```

QUESTIONS

(1) What polynomial has the best testing error?

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

ANSWERS

(1) Polynomial of degree 4.

(2) Order 5 polynomial models do not generalize well because such a high order polynomial is prone to overfitting.