LECTURE 5

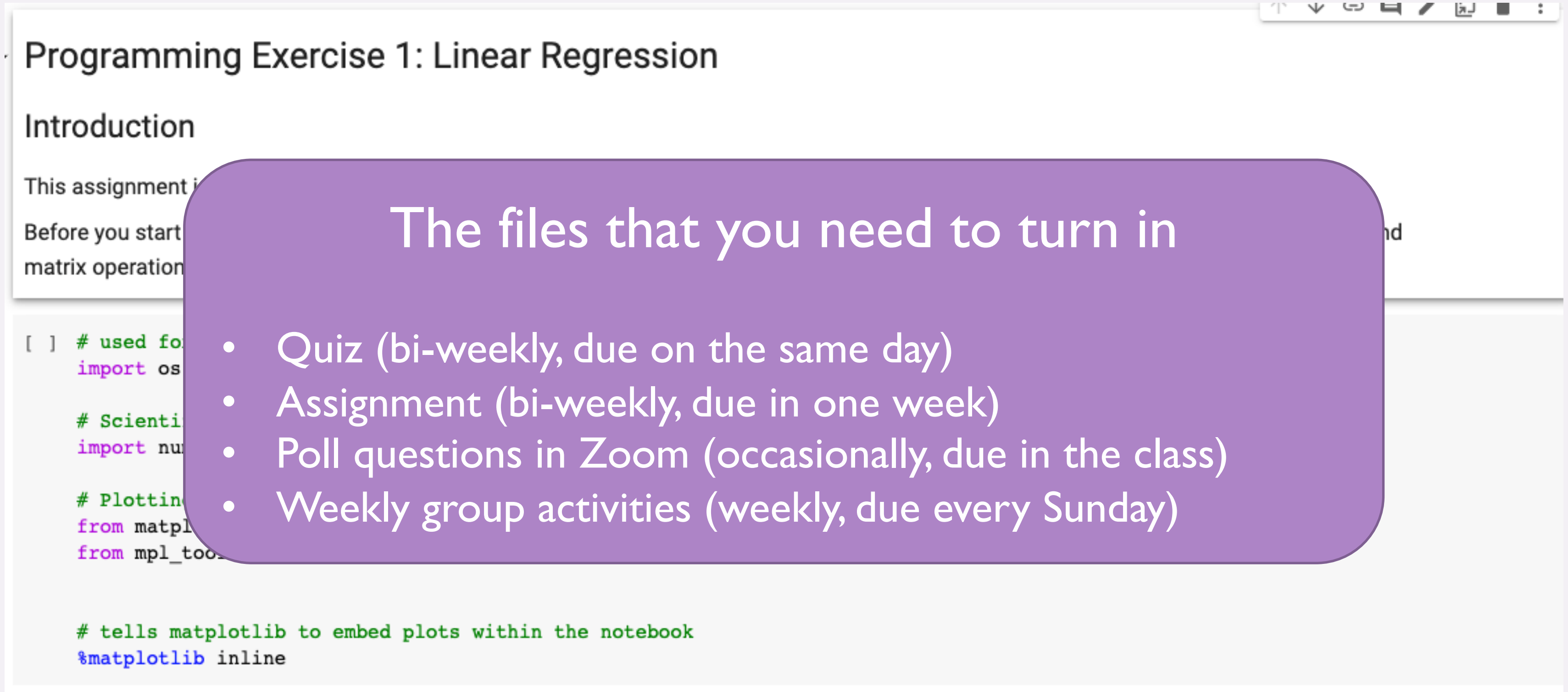
SPRING 2021
APPLIED MACHINE LEARNING
CIHANG XIE

SLIDE CREDIT:
NARGES NOROUZI

EXERCISES

- Google Colab Exercises (distributed during the lecture)

NO NEED TO SUBMIT! JUST TO HELP YOU UNDERSTAND LECTURE



The files that you need to turn in

- Quiz (bi-weekly, due on the same day)
- Assignment (bi-weekly, due in one week)
- Poll questions in Zoom (occasionally, due in the class)
- Weekly group activities (weekly, due every Sunday)

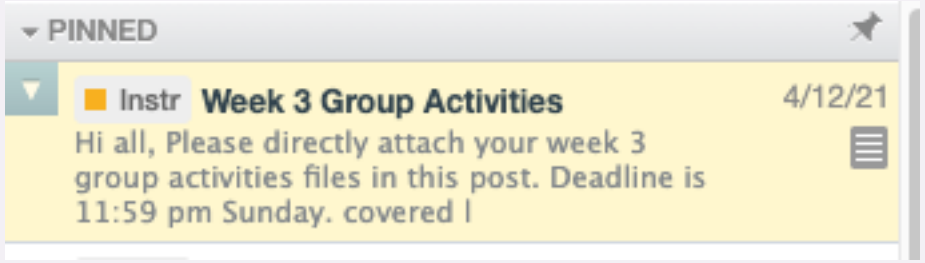
GROUP ACTIVITIES

- ALL groups have submitted the files

  https://piazza.com/class/kmmw9butjod4ay?cid=13

- Please read notes & exercises from other groups

- Week 3 (lecture 4-5)



TODAY

- Review of Gradient Descent Algorithm
- Choosing Learning Rate α
- Basis Functions

LINEAR REGRESSION

GRADIENT DESCENT

- Initialize θ
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial Cost(\theta)}{\partial \theta_j} \qquad \text{(simultaneous update for } \theta_0, \theta_1, \dots, \theta_d\text{)}$$

- For linear regression:

$$\frac{\partial Cost(\theta)}{\partial \theta_j} = \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

With $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_d x_d = \sum_{j=0}^{d} \theta_j x_j$

GRADIENT DESCENT

$$\frac{\partial Cost(\theta)}{\partial \theta_j} = \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$$= \frac{1}{2n} \frac{\partial}{\partial \theta_j} \sum_{i=1}^{n} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

Scalar multiple rule

$$= \frac{1}{2n} \sum_{i=1}^{n} \frac{\partial}{\partial \theta_j} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

Sum rule

$$= \frac{1}{2n} \sum_{i=1}^{n} 2\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \frac{\partial}{\partial \theta_j} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

Power rule

$$= \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)}\right) \frac{\partial}{\partial \theta_j} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)}\right)$$

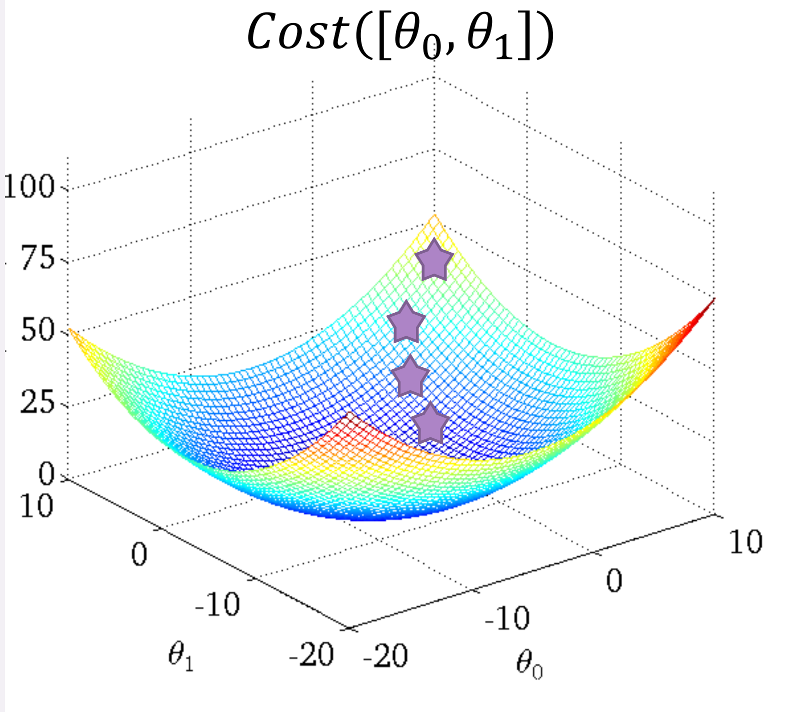
$$= \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)}\right) x_j^{(i)}$$

GRADIENT DESCENT

- Initialize θ
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial Cost(\theta)}{\partial \theta_j} \qquad \text{(simultaneous update for } \theta_0, \theta_1, \ldots, \theta_d\text{)}$$

IN-CLASS QUIZ - WHICH ONE IS CORRECT?



Left:

Temp0 $\leftarrow \theta_0 - \alpha \frac{\partial Cost\ (\theta_0,\theta_1)}{\partial \theta_0}$

Temp1 $\leftarrow \theta_1 - \alpha \frac{\partial Cost\ (\theta_0,\theta_1)}{\partial \theta_1}$

$\theta_0 \leftarrow$ Temp0

$\theta_1 \leftarrow$ Temp1

Right:

Temp0 $\leftarrow \theta_0 - \alpha \frac{\partial Cost\ (\theta_0,\theta_1)}{\partial \theta_0}$

$\theta_0 \leftarrow$ Temp0

Temp1 $\leftarrow \theta_1 - \alpha \frac{\partial Cost\ (\theta_0,\theta_1)}{\partial \theta_1}$

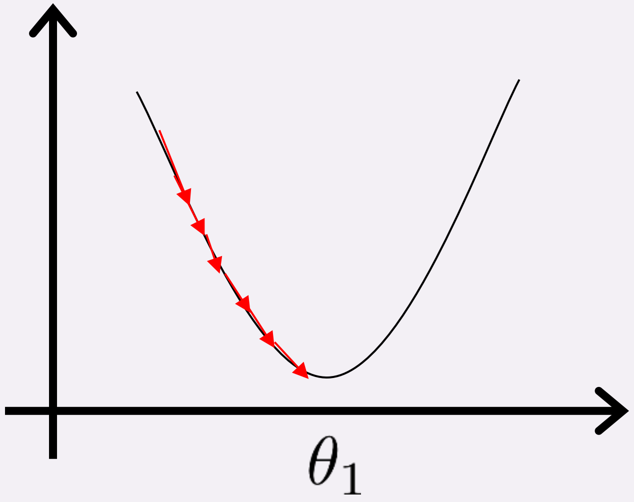
$\theta_1 \leftarrow$ Temp1

CHOOSE α

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial Cost(\theta)}{\partial \theta_j}$$

(simultaneous update for $\theta_0, \theta_1, \ldots, \theta_d$)

Learning rate

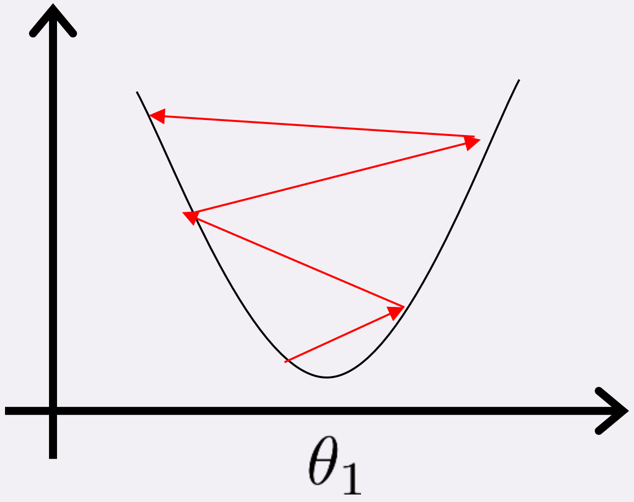


If α is too small, gradient descent can be very slow.

CHOOSE α

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial Cost(\theta)}{\partial \theta_j} \qquad \text{(simultaneous update for } \theta_0, \theta_1, \dots, \theta_d\text{)}$$

Learning rate



If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.

CHOOSE α

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial Cost(\theta)}{\partial \theta_j} \qquad \text{(simultaneous update for } \theta_0, \theta_1, \dots, \theta_d\text{)}$$

Learning rate

For certain functions $Cost(\theta)$, we can theoretically guarantee the convergence of gradient descent by choosing a appropriate α

If interested, please read machine learning course at UBC: lecture 4, starting from page 9
https://www.cs.ubc.ca/~schmidtm/Courses/540-W18/L4.pdf

EXTENDING LINEAR REGRESSION TO MORE COMPLEX MODELS

- The inputs $\boldsymbol{X}$ for linear regression can be:
  - Original quantitative inputs
  - Transformation of quantitative inputs (log, exp, square, etc.)
  - Polynomial transformation (example: $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$)
  - Interactions between variables (example: $x_3 = x_1 \times x_2$)

- This allows use of linear regression techniques to fit non-linear datasets.

LINEAR BASIS FUNCTION MODEL

- Generally,

$$h_\theta(x) = \sum_{j=0}^{d} \theta_j \phi_j(x)$$

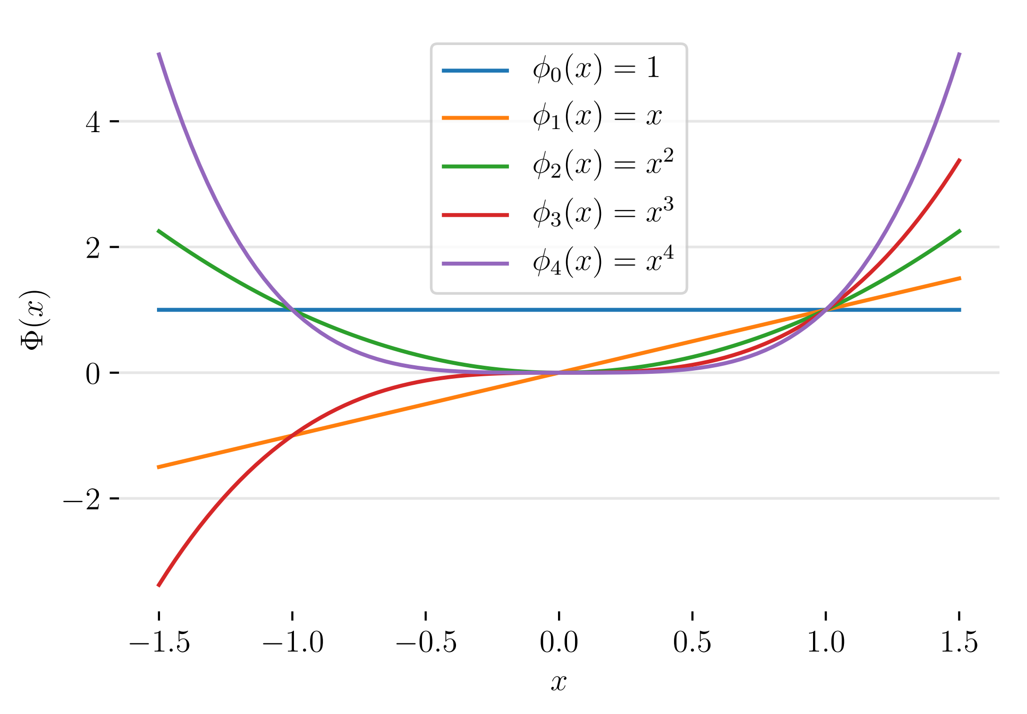
Basis Function

- Typically, $\phi_0(x) = 1$ so that θ_0 acts as a bias.
- In the simplest case, we can use linear basis function:

$$\phi_j(x) = x_j$$

- Polynomial basis function: $\phi_j(x) = x^j$
- Gaussian basis function: $\phi_j(x) = e^{-\frac{\left(x-\mu_j\right)^2}{2s^2}}$

EXAMPLE - POLYNOMIAL BASIS FUNCTION

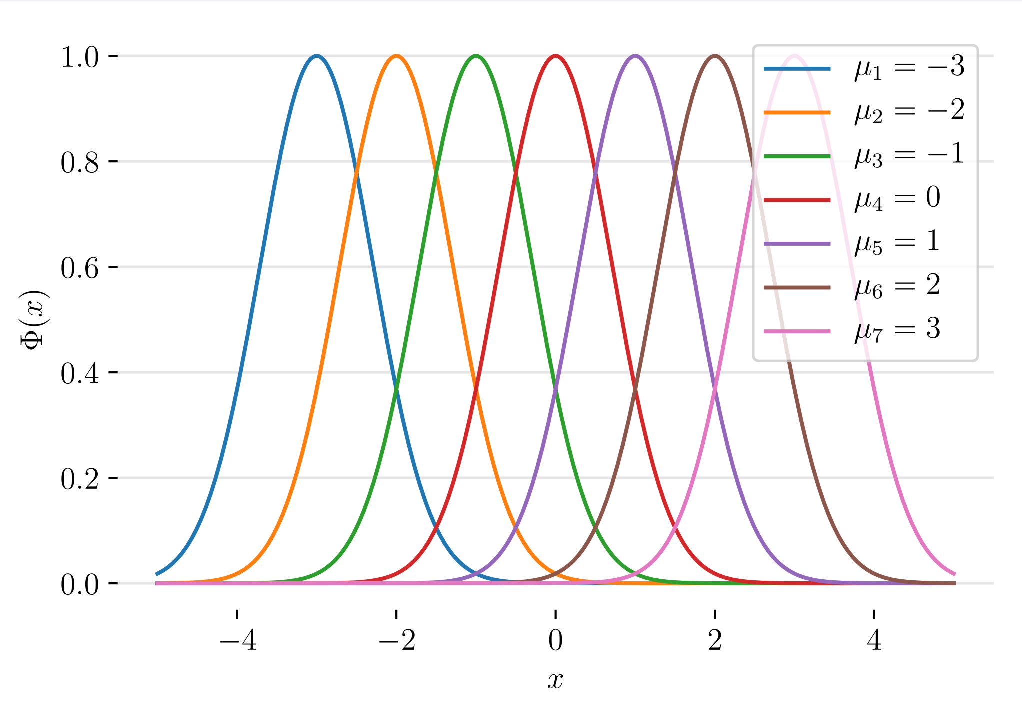


(a) Polynomial basis out to degree 4.

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 = \sum_{j=0} \theta_j x^j$$

Image credit: https://www.cs.princeton.edu/courses/archive/fall18/cos324/files/basis-functions.pdf

EXAMPLE - GAUSSIAN BASIS FUNCTION



(a) Examples of Gaussian-type radial basis functions.

$$y = \theta_0 + \theta_1 e^{-\frac{(x-\mu_1)^2}{2s^2}} + \cdots + \theta_7 e^{-\frac{(x-\mu_7)^2}{2s^2}}$$

Image credit: https://www.cs.princeton.edu/courses/archive/fall18/cos324/files/basis-functions.pdf

EXERCISE

HTTPS://COLAB.RESEARCH.GOOGLE.COM/DRIVE/IVIFN_VBCFAXXAPHZW-WUGJRRGXGLFVFQ?USP=SHARING

QUESTIONS?

HW1 (DUE 4/20)