UCLA Samueli Computer Science


## CS145 Discussion: Week 1

# Course logistics, Math Prep, Linear Regression and Programming Prep

Junheng Hao

Friday, 10/02/2020

# Roadmap

- Course Logistics
- Math Prep: Calculus, linear algebra, probability and optimization
  - In separate slides
- Review: Linear regression
- Programming Prep: Python and Jupyter Notebook
- Q & A

UCLA
Engineer Change.

# Course Information

- Course homepage:
  - https://sites.google.com/view/cs145fall2020/home
  - Please find all the relevant course information there, e.g. schedule, slides, and etc.
- CCLE:
  - Homework, recordings lectures and discussion sessions
- Piazza (for QA):
  - Link: piazza.com/ucla/fall2020/cs145
  - Please ask your question on Piazza before email the professor or any TAs, so others will also benefit from your question.
- Important dates (Mark in your calendar now!)
  - Midterm: Nov 16th, In class
  - Final exam: Dec 16th, 8-10am PT

UCLA
Engineer Change.

# Course logistics

- Office hours & Zoom links
  - Yizhou Sun (yzsun@cs.ucla.edu) Mondays 3:30-5:30pm @Zoom https://ucla.zoom.us/j/93620232458
  - **Junheng Hao (haojh.ucla@gmail.com) Thursdays 8:30-10:30 AM @Zoom https://ucla.zoom.us/j/8391795219**
  - Zijie Huang (zijiehuang@cs.ucla.edu) Mondays 1:00pm-3:00pm @Zoom https://ucla.zoom.us/j/9506978812
  - Yue (Ariel) Wu (arielwu@cs.ucla.edu) Tuesdays 8:30-10:30 am @Zoom https://ucla.zoom.us/j/3698552706
  - Shichang Zhang (shichang@cs.ucla.edu) Wednesdays 1:30-3:30pm @Zoom https://ucla.zoom.us/j/93549332503
- Discussion 1C by Junheng Hao:
  - Link: https://www.haojunheng.com/teaching/cs145-fall20/

UCLA
Engineer Change.

# Course Grading

- **Homework: 30%**
  - 6 assignments (5 highest counted)
  - Submitted by GradeScope (tentative)
  - Late submission penalty:
- **Exam: 35% → Midterm exam: 20%, Final exam: 15%**
  - Open book, no internet access and no communication with others
- **Course project: 25%**
  - Group project (4 or 5 students per group)
  - Topics: COVID-19 Prediction
  - Project details announced next Monday (Oct 12)
- **Participation: 10%**
  - In-class participation
  - Quizzes (valid for 24 hours after release)
  - Piazza

UCLA
Engineer Change.

# Yes, all online!

- As required by UCLA chancellor office, CS145 is entirely online this quarter.
- All teaching activities (lectures, discussion sessions, office hours) will be held virtually through Zoom.
- Please **DO NOT** share the zoom links outside or record the lectures!
- Please **DO NOT** enter the meeting room outside the regular teaching time!

Zoom
University of California

UCLA
Engineer Change.

# Other Questions?

- PTE
- Group formation
- Grading option
- CS145 by Yizhou or CS146 by Kai-Wei?

UCLA
Engineer Change.

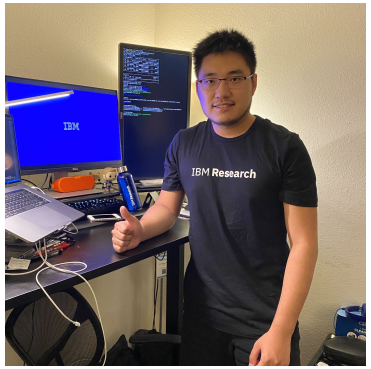
# Announcements

- Oct. 12 (Monday, Week 2): Homework #1 released.
- Oct. 12 (Monday, Week 2): Project introduction and guidelines release.
- Oct. 16 (Friday, Week 2): Group information submission.

Other homework/project/exam deadlines will be announced in class, course website, Piazza, CCLE and my discussion webpage.

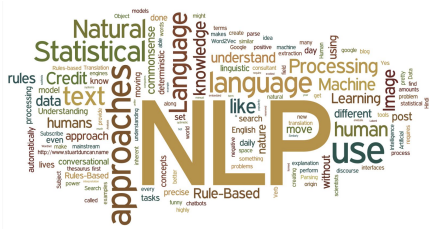
# About TA (Myself)

- Fourth-year Ph.D. candidate
- UCLA Advisors: Yizhou Sun, Wei Wang (UCLA ScAi Institute, UCLA Data Mining Group)
- Past work experiences: @NEC Labs, @Amazon, @IBM Research AI
- Research interests: Knowledge Graphs, Graph mining, NLP, Bioinformatics, etc.
- Hobbies: Languages (beginner for Spanish and German), tennis, …
- More about myself: https://www.haojunheng.com/




# About My PhD Research

- Foundational to knowledge-driven AI systems
- Enable many downstream applications (NLP tasks, Recommender, Bioinformatics, etc)

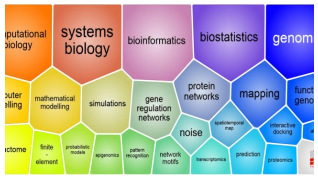



*Natural Language Processing*


*QA & Dialogue systems*

***Knowledge Graphs***


*Recommender Systems*




*Computational Biology*

UCLA
Engineer Change.

# Math Review

- What kind of math is needed for CS145?
  - Just a bit (or a lot) from calculus, linear algebra, probability and optimization…
- Detailed math overview in the slides named “Week 1 Math”.

UCLA
Engineer Change.

# Math Review: Probability

- **Checklist:**
  - Properties of probability
  - Probability spaces (discrete/continuous)
  - Probability distributions (discrete/continuous)
  - Random variables
  - Multivariate probability distributions
  - Marginal probability and conditional probability
  - Expectation, variance, covariance
  - Rules of probability
  - Independence and Bayes rule

# Math Review: Linear Algebra

- **Checklist:**
  - Vector, matrix
  - Multiplication
  - Useful (special) matrices
  - Rank
  - Inverse
  - Eigenvalues and eigenvectors

# Math Review: Optimization

- Checklist:
  - Convex set and convex functions
  - Gradients
  - Gradient descent
- Note: We will learn and implement about gradient descent in this course!

UCLA
Engineer Change.

# Math Review: Linear Algebra

- Checklist:
  - Basic calculus $y = x^2, \frac{\delta y}{\delta x} = ?$
  - Gradient calculation in matrix format

UCLA
Engineer Change.

# Math Review: Reading List

From the class schedule website:

- Review of probability from a course by David Blei from Princeton U.
  - Link: https://www.cs.princeton.edu/courses/archive/spring07/cos424/scribe_notes/0208.pdf
- Machine Learning Math Essentials by Jeff Howbert from Washington U.
  - Link: http://courses.washington.edu/css490/2012.Winter/lecture_slides/02_math_essentials.pdf
- Probability by Arian Maleki and Tom Do from Stanford
  - Link: http://cs229.stanford.edu/section/cs229-prob.pdf
- Optimization
  - Link: http://web.cs.ucla.edu/~yzsun/classes/2019Winter_CS145/Slides/optimization.pdf

# Review: Linear Regression

## Formalization

- Data: n independent data points $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$
  - $y_i$, *dependent variable*
  - $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})^T$, *explanatory variables*
- Model:
  - For any data point $(\boldsymbol{x}, y)$
    - $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^T$: *weight vector*
    - $y = \boldsymbol{x}^T\boldsymbol{\beta} + \beta_0 = \beta_0 + x_1\beta_1 + x_2\beta_2 + \cdots + x_p\beta_p$
  - For convenience, include bias term $\beta_0$ into $\boldsymbol{\beta}$
    - $\boldsymbol{x} = (1, x_1, x_2, \dots, x_p)^T$
    - $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)^T$
    - $y = \boldsymbol{x}^T\boldsymbol{\beta}$

# Review: Linear Regression

## A 3-step Process

- Model Construction
  - Use training data to find the best parameter $\boldsymbol{\beta}$, denoted as $\hat{\boldsymbol{\beta}}$
- Model Selection
  - Use validation data to select the best model
    - E.g., Feature selection
- Model Usage
  - Apply the model to the unseen data (test data):

$$\hat{y}_{new} = \boldsymbol{x}_{new}^T \hat{\boldsymbol{\beta}}$$

# Review: Linear Regression

## Least Square Estimation

- Cost function (Mean Square Error):
  - $J(\boldsymbol{\beta}) = \frac{1}{2}\sum_i \left(\boldsymbol{x}_i^T \boldsymbol{\beta} - y_i\right)^2 / n$
- Matrix form:
  - $J(\boldsymbol{\beta}) = (\mathrm{X}\boldsymbol{\beta} - \boldsymbol{y})^T (X\boldsymbol{\beta} - \boldsymbol{y})/2n$

$$or \left|\left|\mathrm{X}\boldsymbol{\beta} - \boldsymbol{y}\right|\right|^2 /2n$$

$$\begin{bmatrix} 1, x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ 1, x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ 1, x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix} \qquad \begin{pmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix}$$

$\boldsymbol{X}$: $\boldsymbol{n \times (p+1)}$ matrix

y: $\boldsymbol{n \times 1}$ vector

# Review: Linear Regression

## Ordinary Least Squares (OLS)

- Goal: find $\widehat{\boldsymbol{\beta}}$ that minimizes $J(\boldsymbol{\beta})$
  - $J(\boldsymbol{\beta}) = \frac{1}{2n}(\mathrm{X}\boldsymbol{\beta} - y)^T(X\boldsymbol{\beta} - y)$
    $= \frac{1}{2n}(\boldsymbol{\beta}^T X^T X\boldsymbol{\beta} - y^T X\boldsymbol{\beta} - \boldsymbol{\beta}^T X^T y + y^T y)$
- Ordinary least squares
  - Set first derivative of $J(\boldsymbol{\beta})$ as 0
  - $\frac{\partial J}{\partial \boldsymbol{\beta}} = (X^T X\boldsymbol{\beta} - X^T y)/n = 0$
  - $\Rightarrow \widehat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T y$

| $z$ | $\frac{\partial z}{\partial x}$ |
|---|---|
| $\mathbf{Ax}$ | $\mathbf{A}^T$ |
| $\mathbf{x}^T\mathbf{A}$ | $\mathbf{A}$ |
| $\mathbf{x}^T\mathbf{x}$ | $2\mathbf{x}$ |
| $\mathbf{x}^T\mathbf{Ax}$ | $\mathbf{Ax} + \mathbf{A}^T\mathbf{x}$ |

More about matrix calculus:
https://atmos.washington.edu/~dennis/MatrixCalculus.pdf

# Review: Linear Regression

## Batch Gradient Descent

- Move in the direction of steepest descend

Repeat until converge {

$$\boldsymbol{\beta}^{(t+1)} := \boldsymbol{\beta}^{(t)} - \eta \frac{\partial J}{\partial \boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(t)}}, \quad \text{e.g., } \eta = 0.01$$

}

Where $J(\boldsymbol{\beta}) = \frac{1}{2}\sum_i \left(\boldsymbol{x}_i^T \boldsymbol{\beta} - y_i\right)^2 / n = \sum_i J_i(\boldsymbol{\beta})/n$ *and*

$$\frac{\partial J}{\partial \boldsymbol{\beta}} = \sum_i \frac{\partial J_i}{\partial \boldsymbol{\beta}} / n = \sum_i \boldsymbol{x}_i \left(\boldsymbol{x}_i^T \boldsymbol{\beta} - y_i\right) / n$$

# Review: Linear Regression

- Closed form solution

$$\Rightarrow \widehat{\boldsymbol{\beta}} = \left(X^T X\right)^{-1} X^T y$$

- Three types of gradient descent

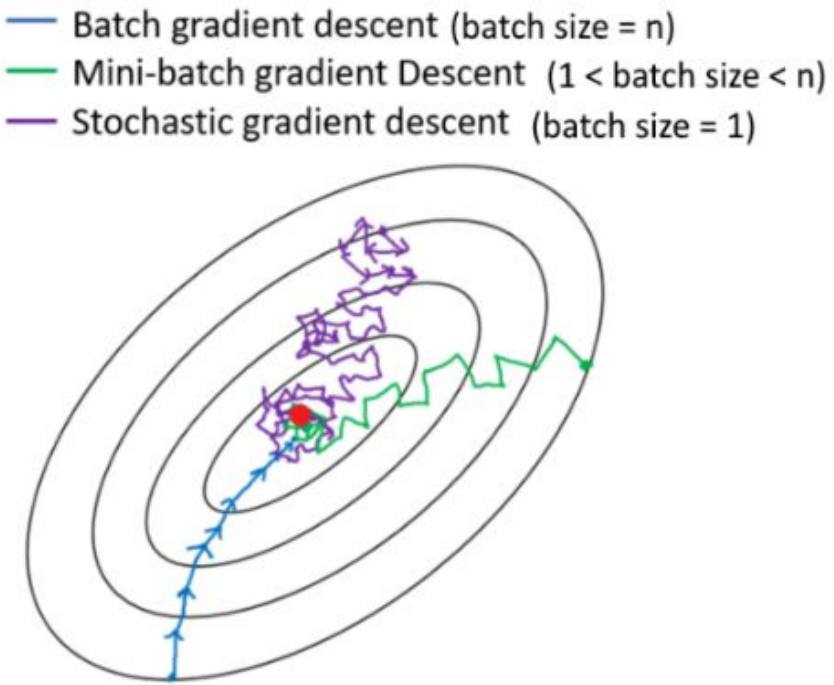



Figure credit:
https://datascience.stackexchange.com/questions/52884/possible-for-batch-size-of-neural-network-to-be-too-small

UCLA
Engineer Change.

# Review: Linear Regression

**Quick questions:**

1. Why do we design various types of gradient descent for linear regression, instead of using closed-form solution?
2. Does gradient descent guarantee convergence in linear regression?
3. What if the learning rate $\eta$ is too large or too small?

UCLA
Engineer Change.

# Python & Jupyter Notebook

- **Step 1:** Install Anaconda (with Python 3.X and Jupyter Notebooks)
- **Step 2:** Try out Python in command line and open Jupyter Notebooks
- **Step 3:** Familiarize yourself with Python 3
- **Step 4:** Use Jupyter Notebooks for coding and writing together
- **Step 5:** Customize your Python environment and install Python packages
    - Example packages: Numpy, Pandas, Matplotlib

UCLA
Engineer Change.

# Where is your Python?

- Install Conda/Anaconda
  - Conda: https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html
  - Anaconda: https://docs.anaconda.com/anaconda/install/mac-os/
- Install Jupyter Notebook from anaconda (this step may be skipped once Anaconda is installed)
  - Link: https://jupyter.org/install
  - Command Line: `conda install -c conda-forge notebook`
- Check out Python and Jupyter notebook
  - Command Line: `python` or `ipython`
  - Version/Source: `python --version` or `which python`
  - Open Jupyter Notebook: `jupyter notebook` (automatically into something URL like: `http://localhost:8888/tree`)
- Demo in class

***Alert:*** *For student using Windows, we may not be able to identify and easily solve the python setting problem.*

UCLA
Engineer Change.

# Jupyter Notebook

- Checklist:
  - Open Jupyter Notebook in your web browser (Chrome)
  - Identify **Markdown** cell and **Code** cell
  - Learn how to use markdown and latex to input math formula
  - Run Python code

- Markdown tutorial → *It is a notebook interface!*
  - Checklist: paragraph, bold, italic, list, code (courier), math formula (in latex)
  - Link: https://www.markdowntutorial.com/
- Latex → *It is for typing math symbols and equations!*
  - No need to install Tex or Mactex
  - Cheatsheet: http://tug.ctan.org/info/undergradmath/undergradmath.pdf
- Demo in class

UCLA
Engineer Change.

# Create customized Python environment

- Checklist:
  - Create a customized virtual environment
  - Activate/Deactivate your environment
  - Install packages for your virtual environment
- Helpful links:
  - Managing conda environment: https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html
- Demo in class

UCLA Samueli Computer Science


# Thank you!

## Q & A