ECE 29595
Introduction to Data Science

Instructors: Milind Kulkarni, Stanley Chan

Wednesdays, 10:30–11:20

a parable of purdue professors

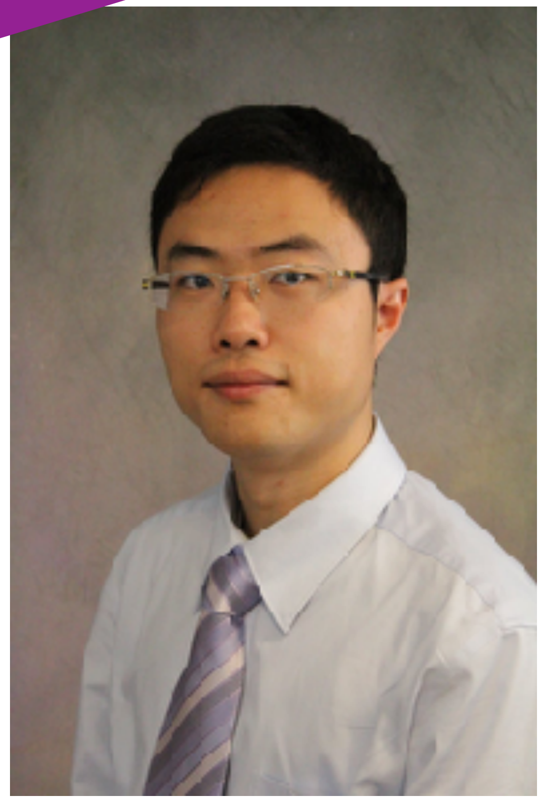
Prof. Seungyoon Lee (Comm) analyzes social media behavior to understand how social networks help people process information

Prof. Bryan Pijanows
sound recordings fro
ecological change

Neville (CS) builds
earning tools
phs and networks

Are they doing data science?

Prof. Milind Kulkarni (ECE) builds systems to make data analyses run faster



Prof. Stanley Chan (ECE and Stats) develops new algorithms for extracting data and signals from noisy images

what is data science?

- Collecting data from a wide variety of sources and putting them into a consistent format?
- Making observations about patterns in da
- Visualizing trends in data?
- Making predictions abo uture?
- Identifying similarity bet points?
- Developing new machine learning and data mining algorithms?
- Accelerating analysis algorithms?

Yes!

data science is a lot of things

using analyses to make predictions

identifying patterns in data

building systems for data analysis

privacy concerns

visualizing data

collecting/organizing data

interpreting data

analyzing data

ethics

writing data analyses

data science is a lot of things

using analyses to make predictions

identifying patterns in data

building systems for data analysis

privacy concerns

visualizing data

collecting/organizing data

interpreting data

analyzing data

ethics

writing data analyses

landscape

- This is the first class in a larger curricular effort in the college: **data mind**
  - Giving students the tools they need to navigate a world of data
- **Foundations**: data literacy, ethics of data science, data collection/curation/organization
- **Methods**: visualization, analysis, machine learning
- **Applications**: using these tools to solve engineering problems

what will you learn in this class?

- Statistics
  - Histograms, probability distributions
  - Sampling, confidence intervals
  - Regression and modeling
  - Classification (naïve Bayes, kNN)
  - Clustering (kmeans)

- Programming (in Python)
  - Higher order functions
  - SciPy stack
  - pandas data structures for analysis
  - Data structures to speed up analyses
  - Basic neural nets

syllabus break!

data analysis in “practice”

- Lets say we have a data set of applicants to Purdue

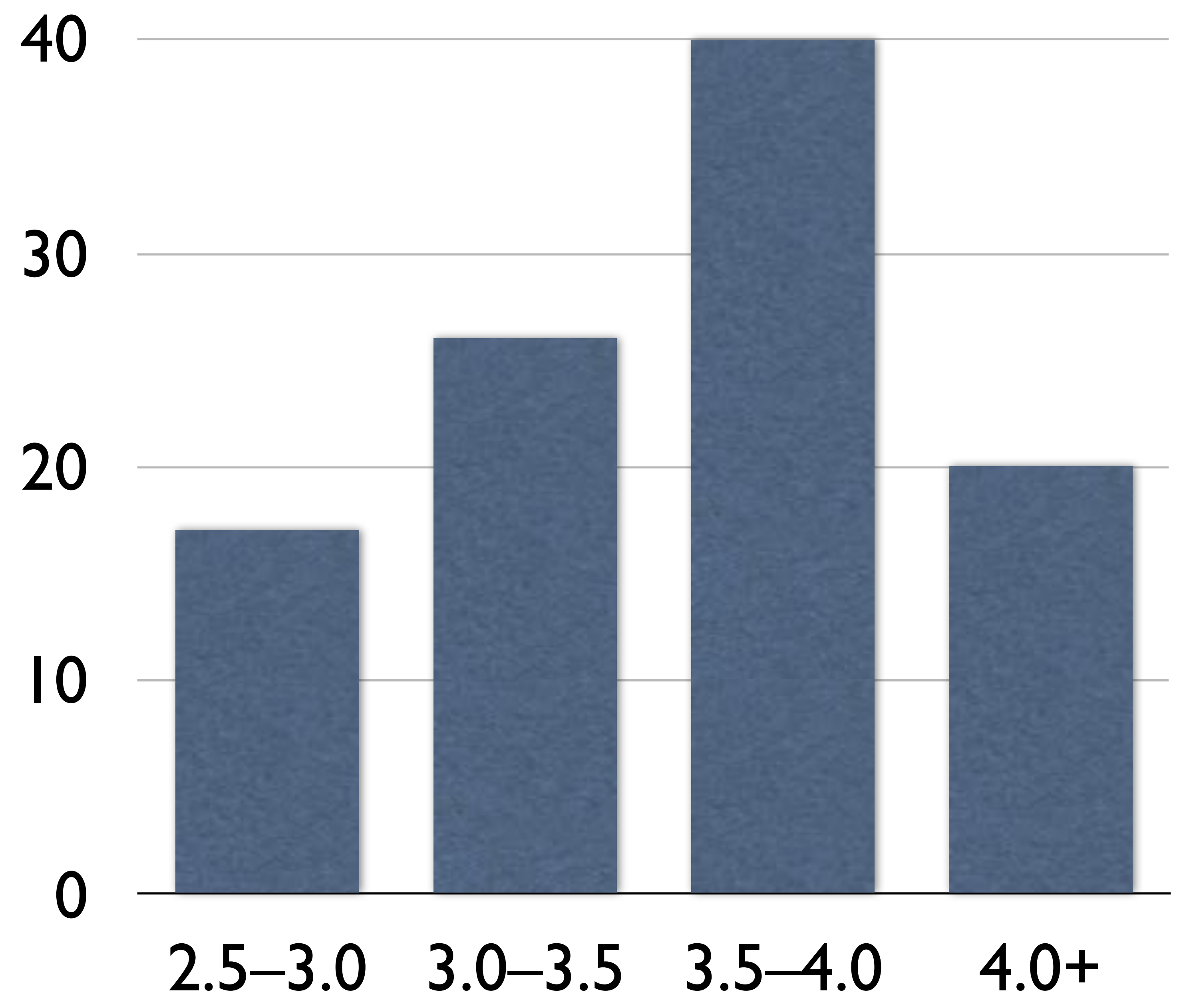
Name	High school GPA	SAT Math	SAT R/W	Residence
Jane Doe	4.7	760	700	Indiana
Purdue Pete	3.5	680	620	Indiana
B. O. ller	3.0	800	650	Michigan
Engy Neer	4.2	750	590	N.C.
…	…	…	…	…

- What might we want to learn about them?

descriptive statistics

- Which students come from which states?
- What is the distribution of GPAs? SAT scores?

- Can build histograms — but how do we know how big to make the buckets?



reasoning about data

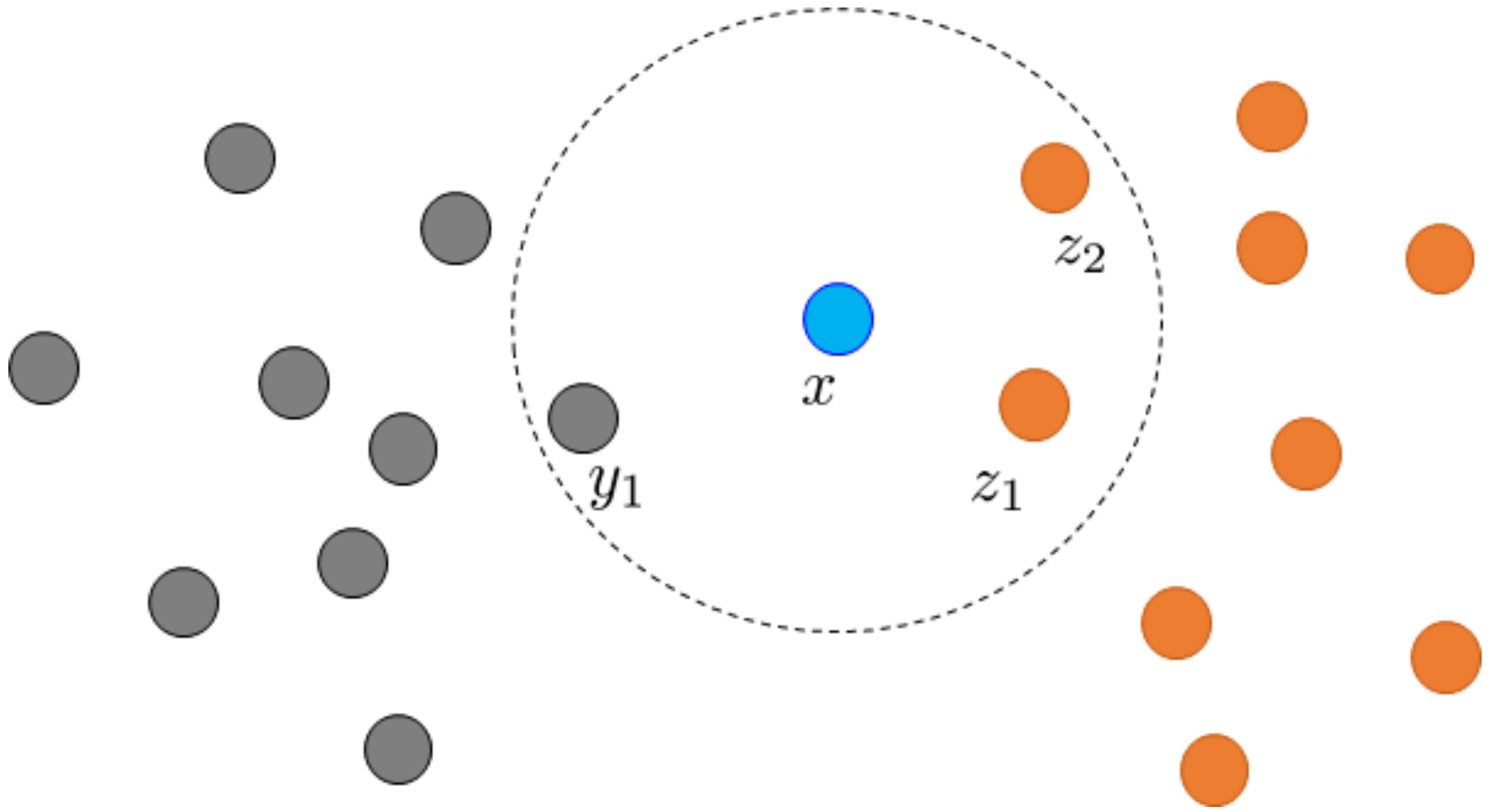
- How do Purdue applicants compare to the national average?
  - Mean GPA of applicants: 3.6
- Is this high or low?
  - Can *sample* GPA of all high school students (randomly collect 1000 GPAs)
  - Mean GPA is 3.4
- Does this mean Purdue students have a higher GPA on average?
- Need more information!
  - Need to know about *variance* of the data (what is the spread of GPAs)
  - Need to know the *confidence interval* (what is the likely range of the true mean GPA?)

making predictions

- Can we predict how successful a particular applicant might be at Purdue?
- Idea: look at the application statistics of the current *seniors* and see if there is a relationship between their statistics and their Purdue GPA
- One way to find a relationship is using *linear regression*
    - Might tell you something like: "a Purdue student's GPA is predicted mostly by their high school GPA, and not very much by their SAT score"

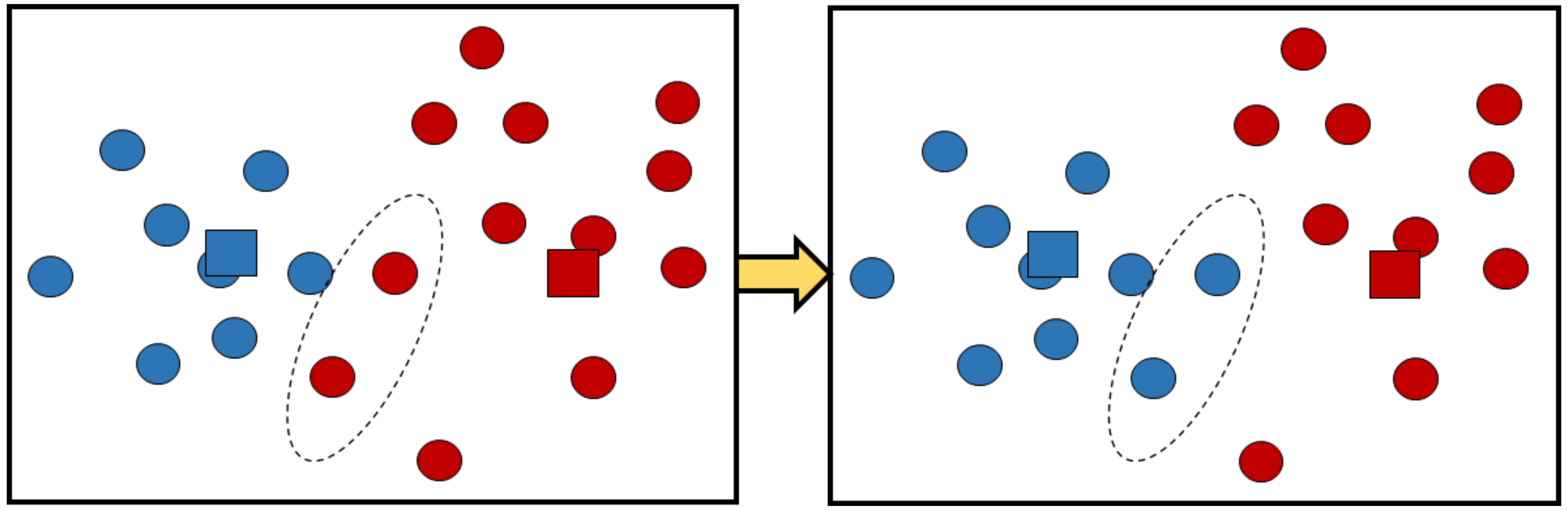
classifying students

- What if I want to make admissions decisions more quickly
- Predict whether a student should be accepted or not
- Idea: compare each applicant to past applicants *that were admitted and those that were rejected*
  - See whether this applicant is more similar to other admitted applicants, or to rejected applicants
  - This is a *k-nearest neighbor* classifier



grouping students

- What if I just want to know if there are different groups of students
- Idea: see if students are clustered together in some way
  - Some students look more like “nearby” students than students that are “far away”
  - Questions: what *features* of students should you consider (e.g., maybe don’t consider something like hair color!)
- This is *k-means clustering*



environment setup

- Setting up github
- Setting up python environment (on ecegrid)
- Jupyter notebook demo