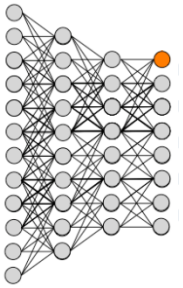
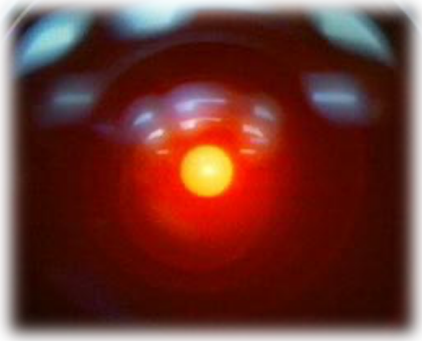
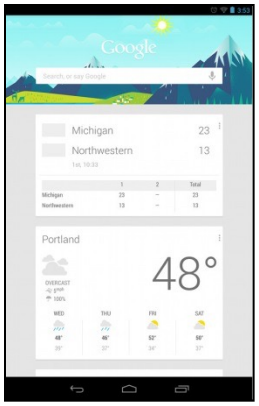




Machine Learning

CSE 142

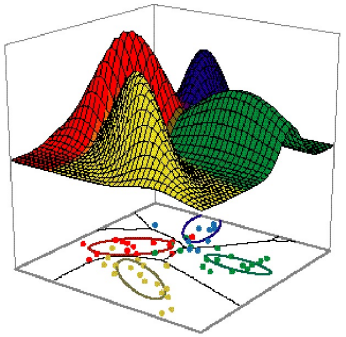
Fall 2021



Xin (Eric) Wang

Assistant Professor

Computer Science and Engineering



Instructor and Venue

- Teaching Staff:
  - Instructor: Xin (Eric) Wang
  - TA: Jing Gu
  - Readers: Tongzhou Jiang, Diji Yang
  - Tutors: Furqan Mohammed, Winson Chen
- Lectures: M W F 2:40-3:45pm PT
- Discussion Sessions: Tue 12:30-1:30 *or* Thu 11:30-12:30
- Tutorial Sessions:
  - Furqan: Wed 10:30-11:30am PT
  - Winson: Thu 9:00-10:00am PT
- Office Hours:
  - Instructor: Wed 3:50-4:50pm PT
  - TA: Mon 9:30-10:30am PT
  - Reader: Thu 2:00-3:00pm PT

Instructor and Venue

- Lectures, discussions, and office hours are all Zoom meetings.
- Lectures:
  - You are expected to attend all lectures (random in-class pop quizzes).
  - Camera on is encouraged.
  - Reading must be done **before** lectures.
  - Slides will be posted after lectures.
- Discussion Sessions:
  - Required.
  - Will be recorded and posted on Canvas.
  - Covering supplementary materials and answering general questions.
- Tutorial Sessions:
  - Optional.
  - Assisting students with understanding course concepts; developing study strategies and methods for independent work; developing writing and critical thinking skills through reviewing; preparing for exams.

Course Websites

- Course website is on Canvas: https://canvas.ucsc.edu/courses/46649
  - Policy, Logistics, Syllabus, Zoom meetings, Announcements, Assignments, etc.

- Piazza used for discussions and questions.
  - Sign up Piazza at piazza.com/ucsc/fall2021/cse142fall2021 (access code: CSE142)

- Check Canvas and Piazza daily or every other day

- Emailing the instructor will **not** be the fastest way to get a response

Grading

- Grades will be based on
  - Participation (e.g., in-class quizzes and Piazza) (5%)
    - Bonus points to the top answerers on Piazza
  - 4 Assignments (40%) → Individually done
  - Midterm Exam (20%) → In class
  - Final Exam (35%)
  - (This allocation may change during the quarter)

Assignments

- Writing & programming assignments on Machine Learning
- Must be done individually
- Assignment due dates are **strict** (no extensions)
- Late submission policy:
  - 4 late days in total are allowed with no penalty
    - Freely allocate to any assignments
    - Counted by days (not hours/minutes)
  - 50% will be deducted if it is within 4 days after the due day
  - Zero credits for assignment submissions that are 5 days late

Exams

- A midterm and a final
- Based on lectures and rest of the course material
- More details will be released one week before exams

Syllabus

- **https://canvas.ucsc.edu/courses/46649**
- **Required textbook**: *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*, by Peter Flach, Cambridge University Press, 2012
  - Available at Amazon, GitHub, etc.
- This course is about understanding machine learning principles
  - "Teaching you to fish," "walk before running," "milk before meat," etc.
- From here, you can continue to learn ML more deeply
- Goal: learn to think about ML correctly, to deeply understand key ML concepts
  - And several basic ML techniques

- P.S. It won't be all about deep learning!

How to Succeed in CSE 142

- Do the assigned reading
  - The lectures will overlap somewhat with the reading, but nowhere near 100% – both lecture and assigned reading are critical
  - The textbook supplements the lecture (providing additional depth, examples, and topics not mentioned in class)
  - The lecture supplements the textbook (providing additional depth, examples, and topics not mentioned in the textbook)
- Attend the discussion sessions
  - Various topics: more details on lecture material, algorithms examples, homework help, questions, etc.
- Be engaged in class
  - Ask questions, offer feedback…
  - Review the lecture notes after class (posted soon after class ends)
- Get started on the homework assignments early

DRC and Course Enrollment

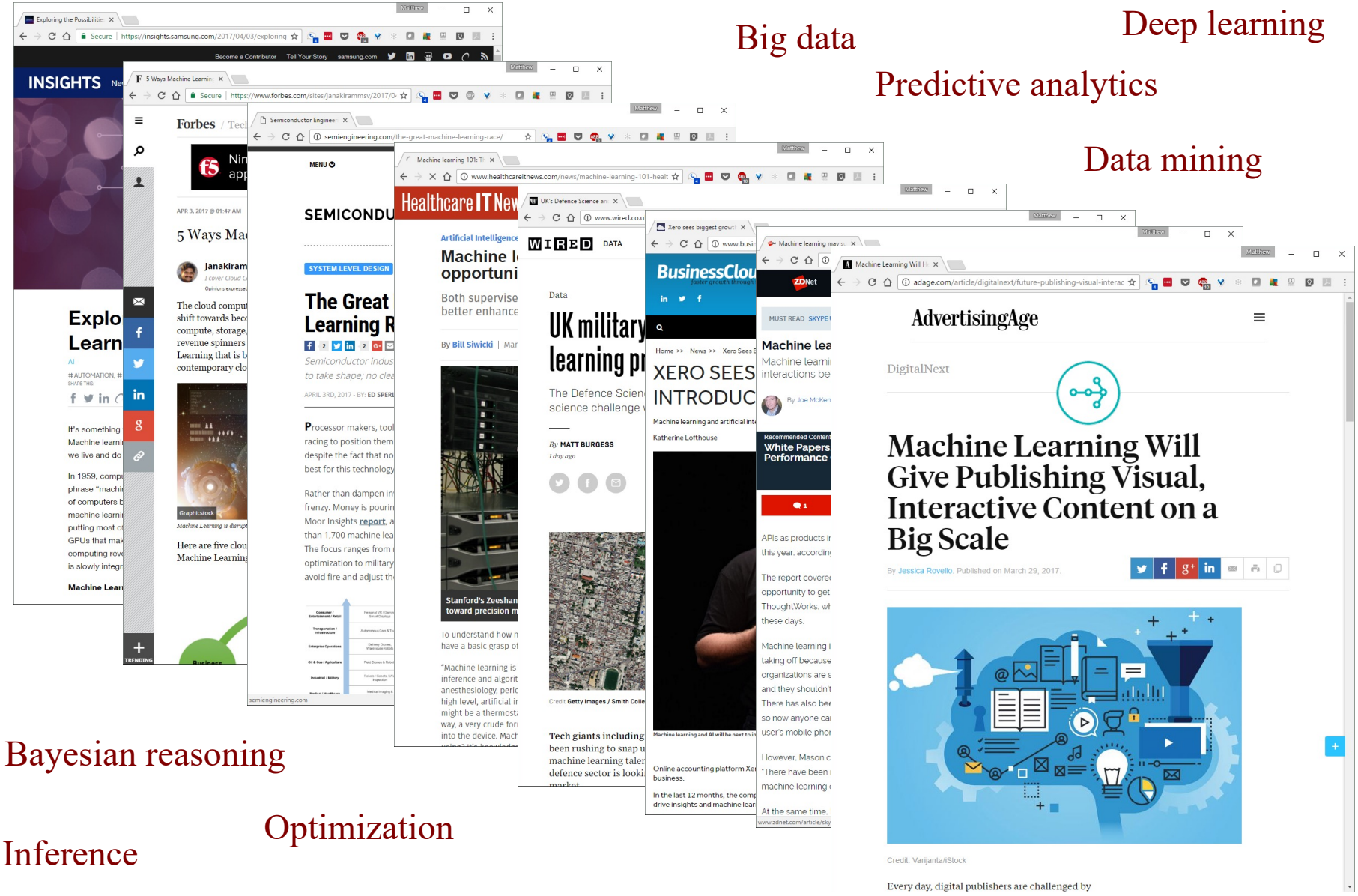
- If you have special needs, contact the Disability Resource Center and send your Accommodation Authorization form to me **during the first two weeks**

- Students requiring permission codes
  - Send me an email shortly with "CSE142 Enrollment – Your Name" as the header to state your interest and require the permission code
  - No time conflict with other courses
  - Highest priority to CSE undergraduates in case of space concern

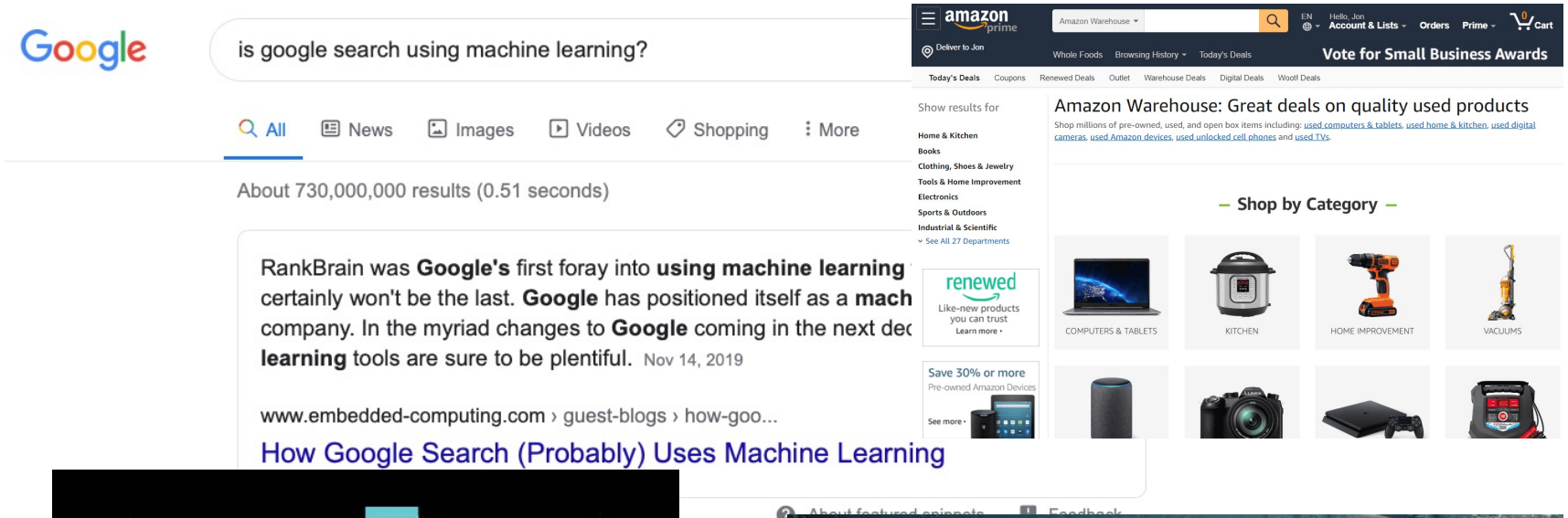
Academic Integrity

- Read the UCSC Academic Integrity Policy on the Canvas course website: https://canvas.ucsc.edu/courses/46649
- Cannot copy code or answers to homework questions or exams
- May discuss course materials and assignments, but **cannot write anything down**
- Must write down answers or code individually
- The presence or absence of any form of help or collaboration must be explicitly stated by all involved
- Academic dishonesty will be reported and students who engage in it will receive an F automatically

Machine learning is taking over the world!



Machine learning in your daily life

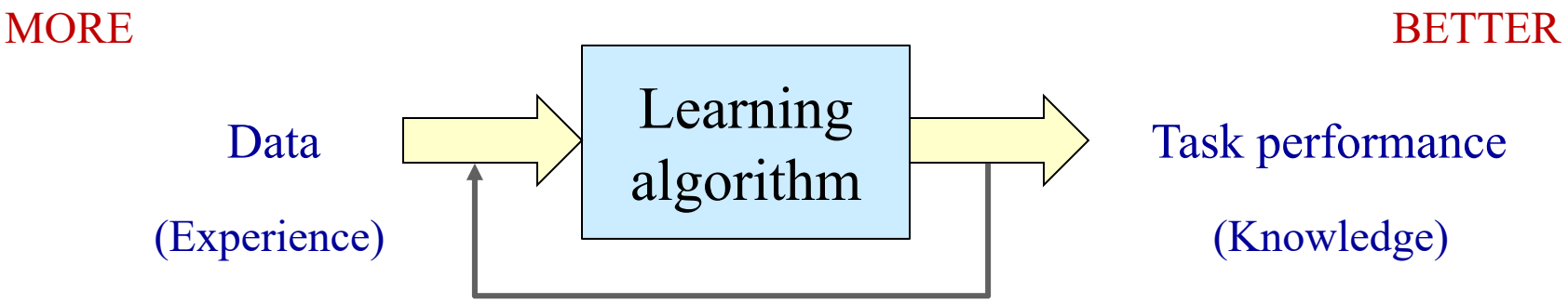


Applications of machine learning

- Search engines
- Data mining
- Computer vision, including object recognition
- Natural language processing
- Bioinformatics
- Syntactic pattern recognition
- Medical diagnosis
- Brain-machine interfaces
- Cheminformatics
- Detecting credit card fraud
- Stock market analysis
- Classifying DNA sequences
- Sequence mining
- Robotics
- Speech and handwriting recognition
- Game playing
- Software engineering
- Adaptive websites
- Computational advertising
- Computational finance
- Structural health monitoring
- Sentiment analysis (or opinion mining)
- Affective computing
- Information retrieval
- Recommender systems

Introduction to machine learning

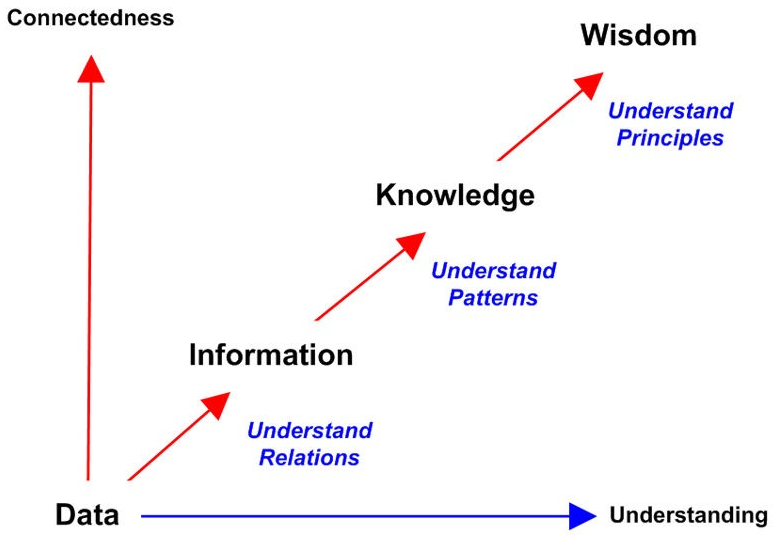
Machine learning is *the design and analysis of algorithms that improve their performance at some task with experience.*



The learning may take place offline (before the execution of the task) –
e.g., face detection, spam detection
It may take place online (as a task progresses) – e.g., an adaptive interface
It may do both – e.g., speech recognition systems

The big picture

- We wish to develop methods and tools for building learning machines that can solve problems in combination with available data sets of training examples
- Ultimately, to move up the "DIKW pyramid"
  - To be able to reveal principles, reveal directions, answer questions, make decisions, determine action
  - From understanding the past to empowering the future



Data science!

How do we learn?

When a machine improves its performance at a given task over time, without reprogramming, it can be said to have learned something

Some ways of learning for people:

- Rote learning, i.e., memorization or "muscle learning"
- Conditioning (associative learning)
- Learning from specific instructions
- Learning from explanations
- Learning from advice
- Learning from examples
- Learning by doing
- Learning by exploration and discovery
- Learning by analogy

The typical machine learning problem

- Given a collection of examples (the training data), we want to build a model to predict something about novel examples

- For example:
  - Spam filtering
    - Is this email spam or ham?
  - Medical diagnosis
    - Do these test results indicate malignant or benign?
  - Natural language processing
    - What does the sentence ask for?
  - Face recognition
    - Who is this a picture of?
  - Go
    - What move to make now?

Machine learning?

The output depends on the learning problem!

- What's the next number in the sequence:
  - 1, 2, 4, 8, … **20**
  - "A big dog had the run of the land and fiercely protected it and the animals on it."

Best Friends:
A Bedtime Story

- How about this sequence:
  - 3, 4, 5, 2, … **5**
  - 3.141592653589793
- Is this image in the same class as these images?

No, it's indoors



- What's the outlier in this group:
  - California, Washington, Hawaii, Canada, Mississippi, New Orleans, Minnesota, Nevada, Utah, New York
  - **Mississippi** – I've never been there!

What is a learning problem?

- Learning involves improving performance
  - at some task ***T***
  - with experience ***E*** (i.e., data)
  - evaluated in terms of performance measure ***P***

- Example: learn to play checkers
  - Task ***T***: playing checkers well
  - Experience ***E***: game database, playing against itself
  - Performance ***P***: percent of games won against humans

- What exactly should be learned?
  - How might this be represented?
  - What specific algorithm(s) should be used?

Components of a learning problem

- Task: the behavior or task that's being improved; e.g., classification, object recognition, acting in an environment
- Data: the experiences that are being used to improve performance in the task
- Measure of performance: How can the improvement be measured? Examples:
  - Provide more accurate solutions (e.g., increasing the accuracy in prediction)
  - Cover a wider range of problems
  - Obtain answers more economically (e.g., improved speed)
  - Simplify codified knowledge
  - New skills that were not presented initially

Thank you!

- Sign up Piazza and vote for the discussion session time today
- Read Prologue
- Discussion sessions and office hours start next week