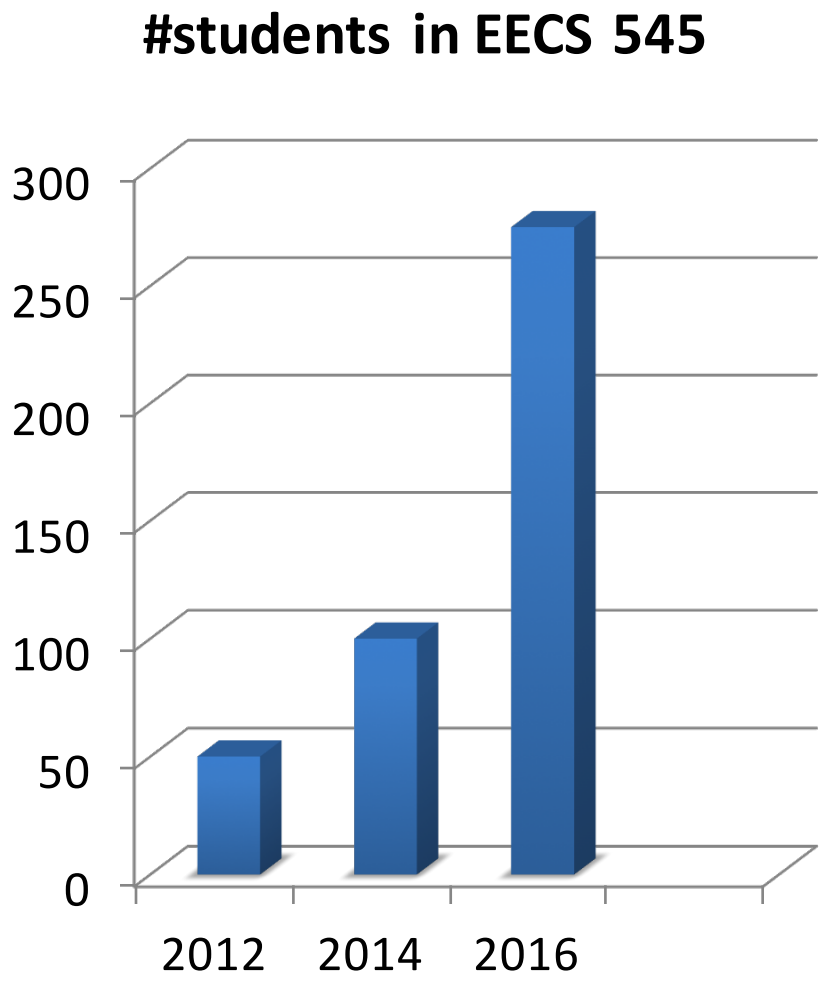
EECS 545: Machine Learning
Lecture 1. Introduction

Jake Abernethy

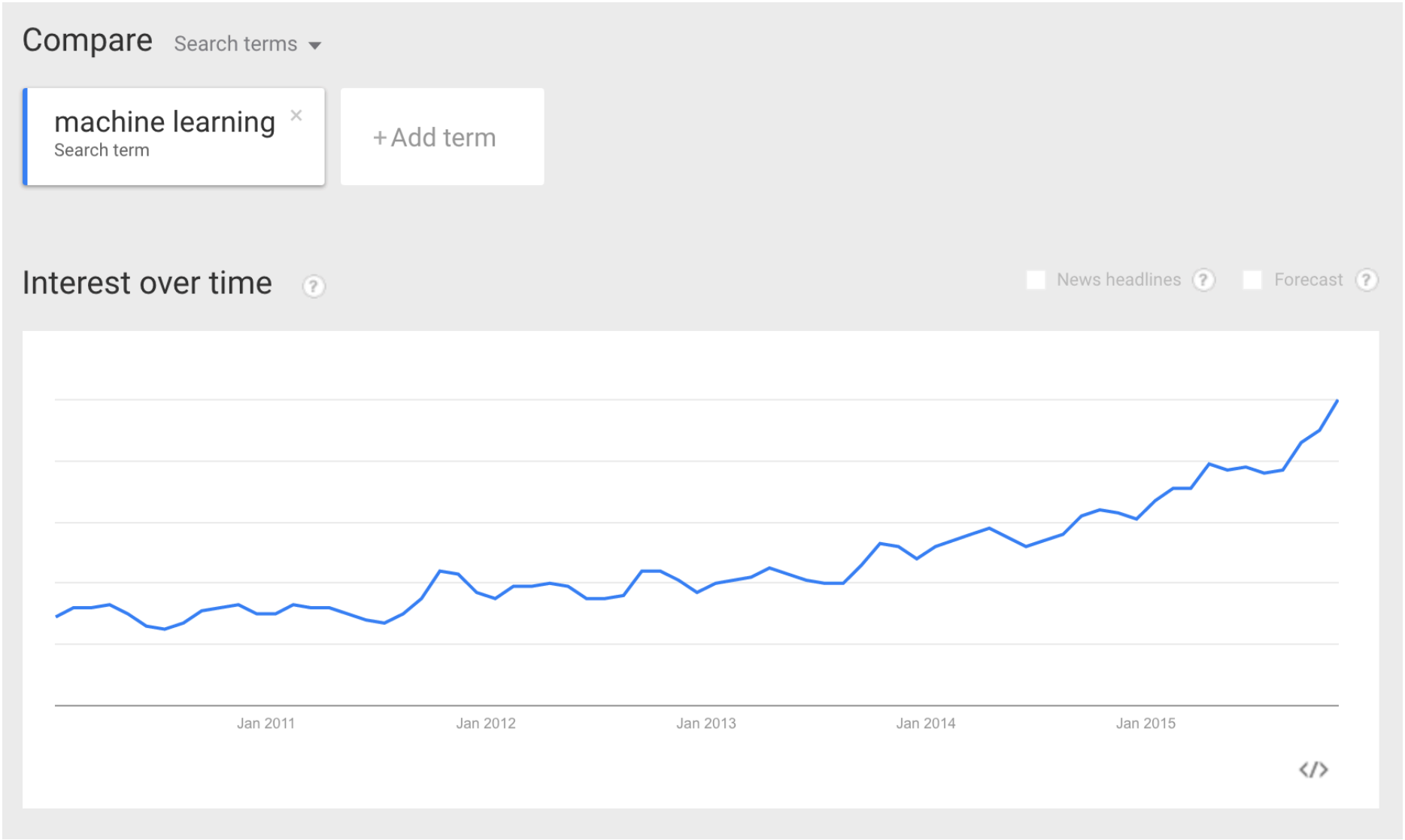
1/6/2015

My First Big Question

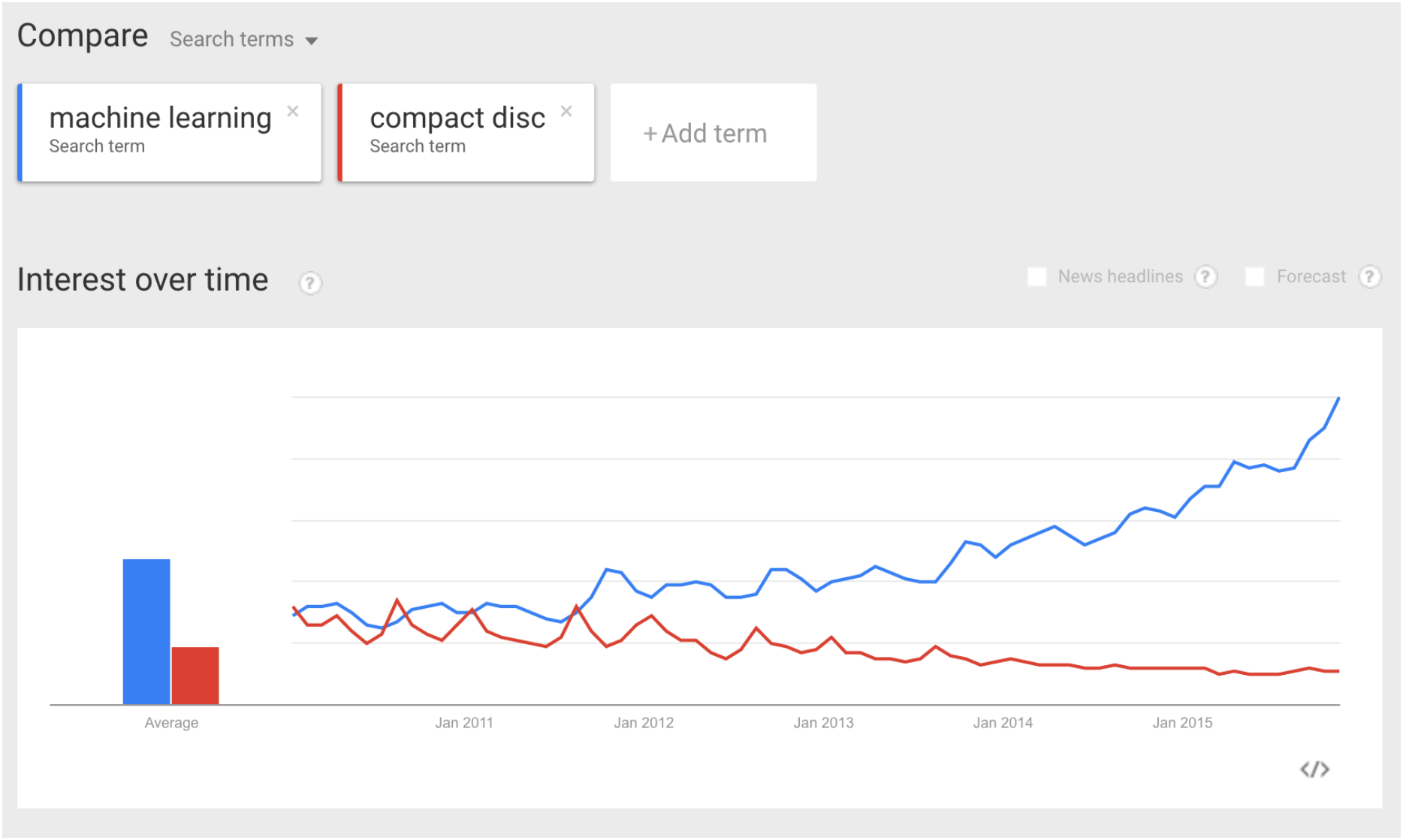
- What are so many of you people doing in this class room?
- No really…
- I had to fight to get this room to fit all you people.



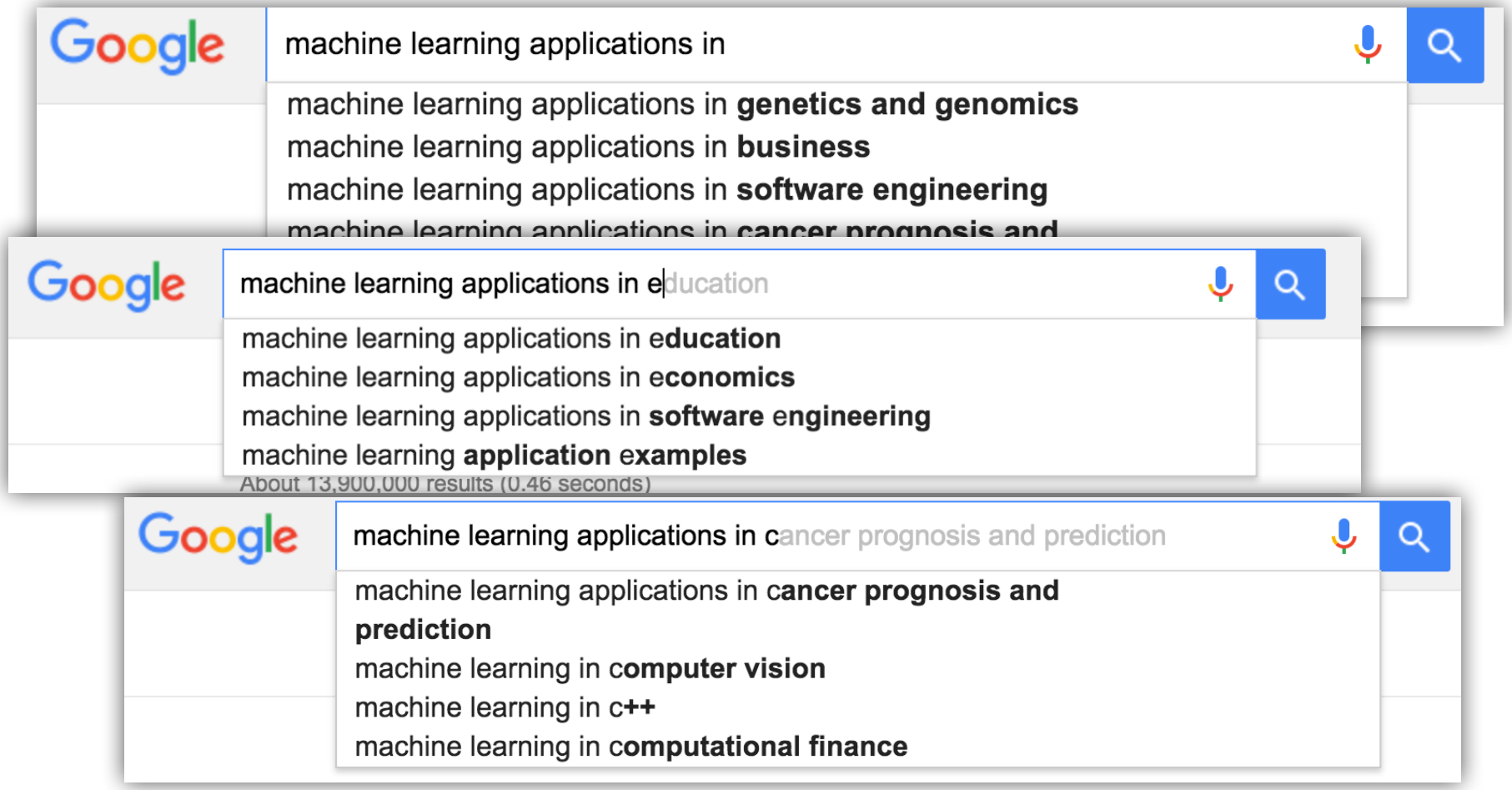
Machine Learning on Google Trends



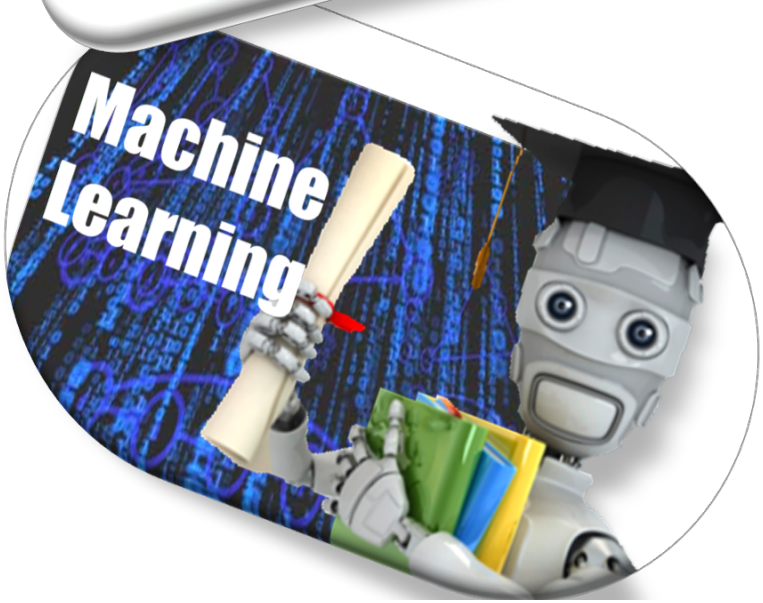
Machine Learning on Google Trends



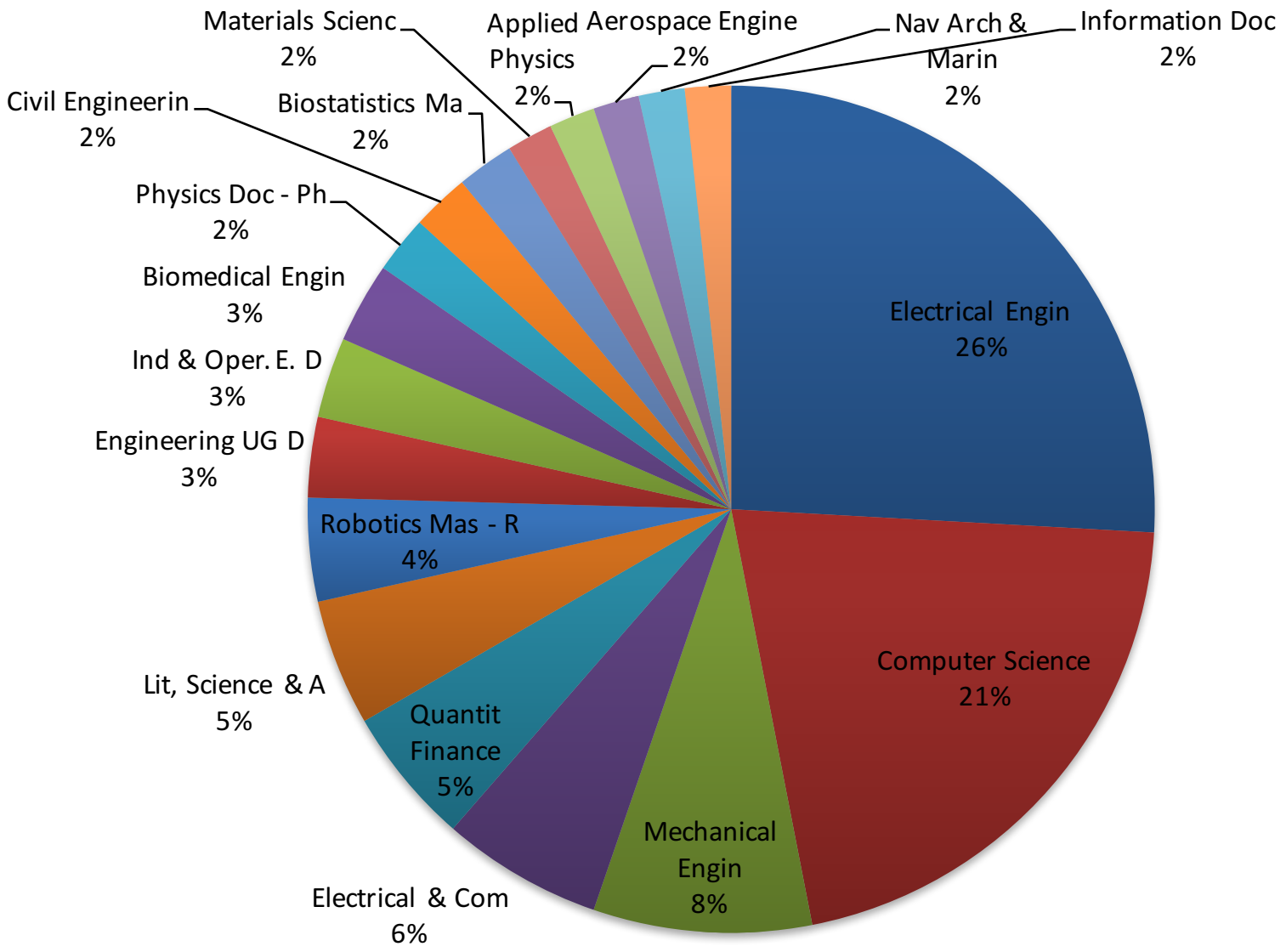
ML ➔ Lots of Applications



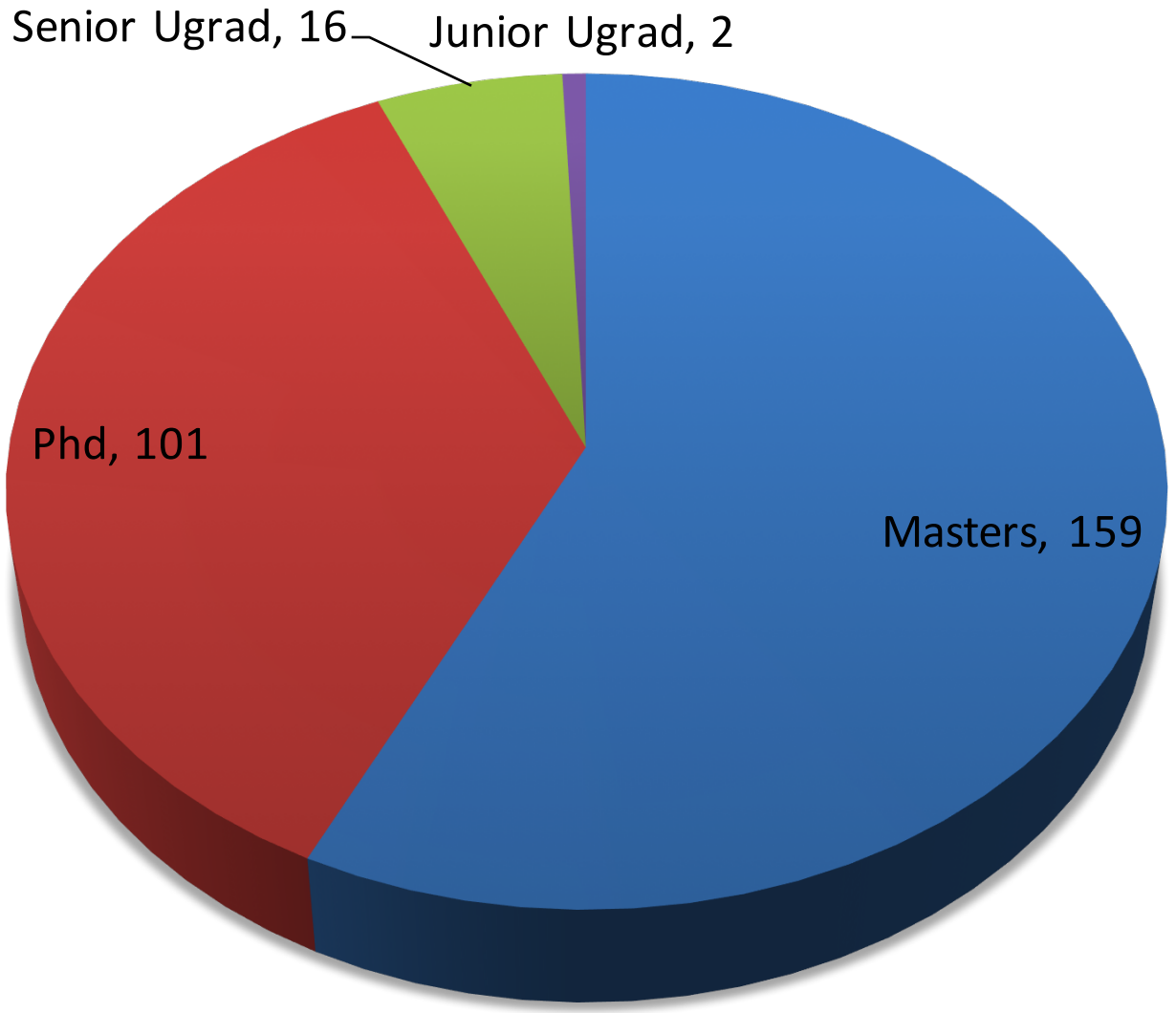
Machine Learning, according to Google Image Search



Where do you people come from?



What level are you people?



Outline

- Administrative

- What is machine learning?

Teaching staff

- Instructor: Jacob Abernethy
  - Email: jabernet@umich.edu
  - Office: CSE 3765
  - Office hour: Tuesdays 9:30am-12pm
- Graduate Student Instructor:
  - David Ke Hong – kehong@umich.edu
  - Daniel Lejeune – dlejeune@umich.edu
  - Changhan Wang - wangchh@umich.edu
  - (IA) Benjamin Bray – benrbray@umich.edu
  - Will hold review sessions on background materials (linear algebra, probability, Python, etc.)
- For all questions, please use Piazza:
  - http://piazza.com/umich/winter2016/eecs545

About this course

- Graduate-level introduction of machine learning
- Provide foundations of machine learning
  - Mathematical derivation, Implementation of the algorithms, Applications
- Topics
  - supervised learning
  - unsupervised learning
  - Kernel methods
  - Graphical models
  - Various advanced topics (TBD):learning theory, sparsity and feature selection, Bayesian techniques, ensemble methods, deep learning, multi-armed bandits

About this course

- Cover practical applications of machine learning
  - computer vision, data mining, speech recognition, text processing, bioinformatics, and robot perception and control
- Our goal is to help you to
  - Understand fundamentals of machine learning
  - Learn technical details of ML algorithms
  - Learn how to implement some important algorithms
  - Use machine learning algorithms for your research and applications.

Text books

- (main) Chris Bishop, “Pattern Recognition and Machine Learning”. Springer, 2007.
- (optional) Hastie, Tibshrani, Friedman, “Elements of Statistical Learning”. Springer, 2010.
  - (available online) http://www-stat.stanford.edu/~tibs/ElemStatLearn/
- (optional) Boyd and Vandenberghe, "Convex Optimization," Cambridge University Press, 2004.
  - (available online) http://www.stanford.edu/~boyd/cvxbook/
- (optional) Mackay, “Information Theory, Inference, and Learning Algorithms”. Cambridge University Press. 2003.
  - (available online) http://www.inference.phy.cam.ac.uk/itprnn/book.pdf

Prerequisites

- EECS 492: Introduction to AI
  - (official requirement, but you can still take this course without completing 492)
- Undergrad linear algebra (e.g., MATH 217, MATH 417)
- Multivariate calculus
- Undergrad probability and statistics (e.g., EECS 401)
- Programming skills (equivalent to EECS 280, EECS 281, experience in Python or Matlab)
  - Nontrivial level of programming is required.

- NOTE: If you **have not** taken **at least two** of linear algebra, multivariate calculus, and probability courses, it is **strongly recommended** that you finish them first before taking this course.

Grading policy

- Homework: 40% (6 HWs, lowest dropped)
- Midterm: 25%
- Final Exam: 25%
- Participation/Quizzes: 10%
- Some options for extra credit:
  - Scribing
  - HW Solutions
- OPTIONAL Final Project (Details to come.)

Homework

- There will be 6 (bi-weekly) problem sets.
- Goal: strengthen the understanding of the fundamental concepts, mathematical formulations, algorithms, and the applications.
- The problem sets will also include programming assignments to implement algorithms covered in the class.
- Homework #1 will be out next Monday (1/11) – due 1/25, 11pm.

Language of Choice: **Python**

- **Python** is a better language overall for data science, with modern libraries and excellent online tutorials for various tasks
- We will utilize **Python** throughout the course, and we encourage you to do the same
- GSIs will run **Python** help sessions and will provide fast answers on Piazza

- Homeworks will be *mostly* language-independent, so we will permit use of Matlab

Study group

- Form your study group early on!
  - Up to four people are allowed.
- For homework, you may discuss between the study group members, but you should write your own solution independently.
- In the homework submissions, you must put:
  - the names of other people you collaborated.
  - submission time.
- Please start on homework early. (Warning: cramming does not work!)

Other Information

- Review sessions
  - Will hold review sessions on background materials (linear algebra, probability, Python, etc.)
- Canvas site posted soon!
- No email policy! Use Piazza!
  - Piazza allows for private questions
  - Email your GSI if it's too personal for Piazza
  - Email me if it's too personal for your GSI

Any questions?

Outline

- Administrative

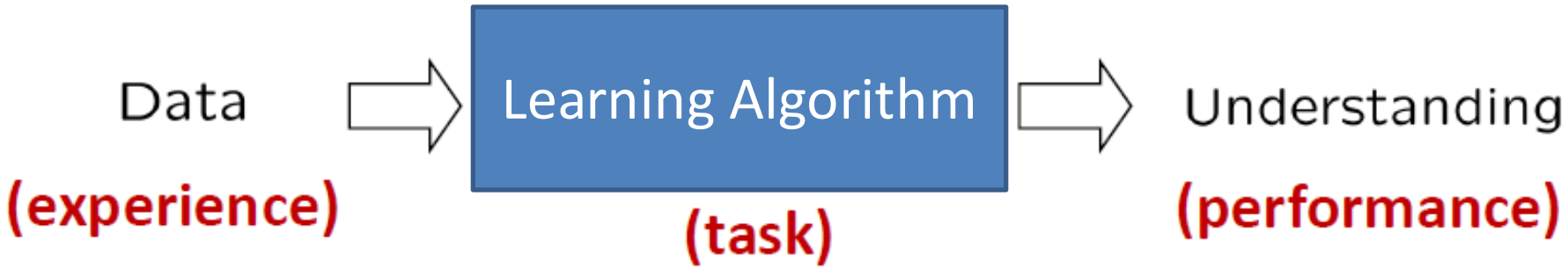
- What is machine learning?

Definition of Machine Learning

- Formal definition (Mitchell 1997): A computer program **A** is said to **learn** from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experience **E**.

Informal definition

- Algorithms that improve their prediction performance at some task with experience (or data).



Example: Spam email filtering

"A computer program is said to *learn* from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."

- Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam.

- Task:
  - Classifying emails as spam or not spam.

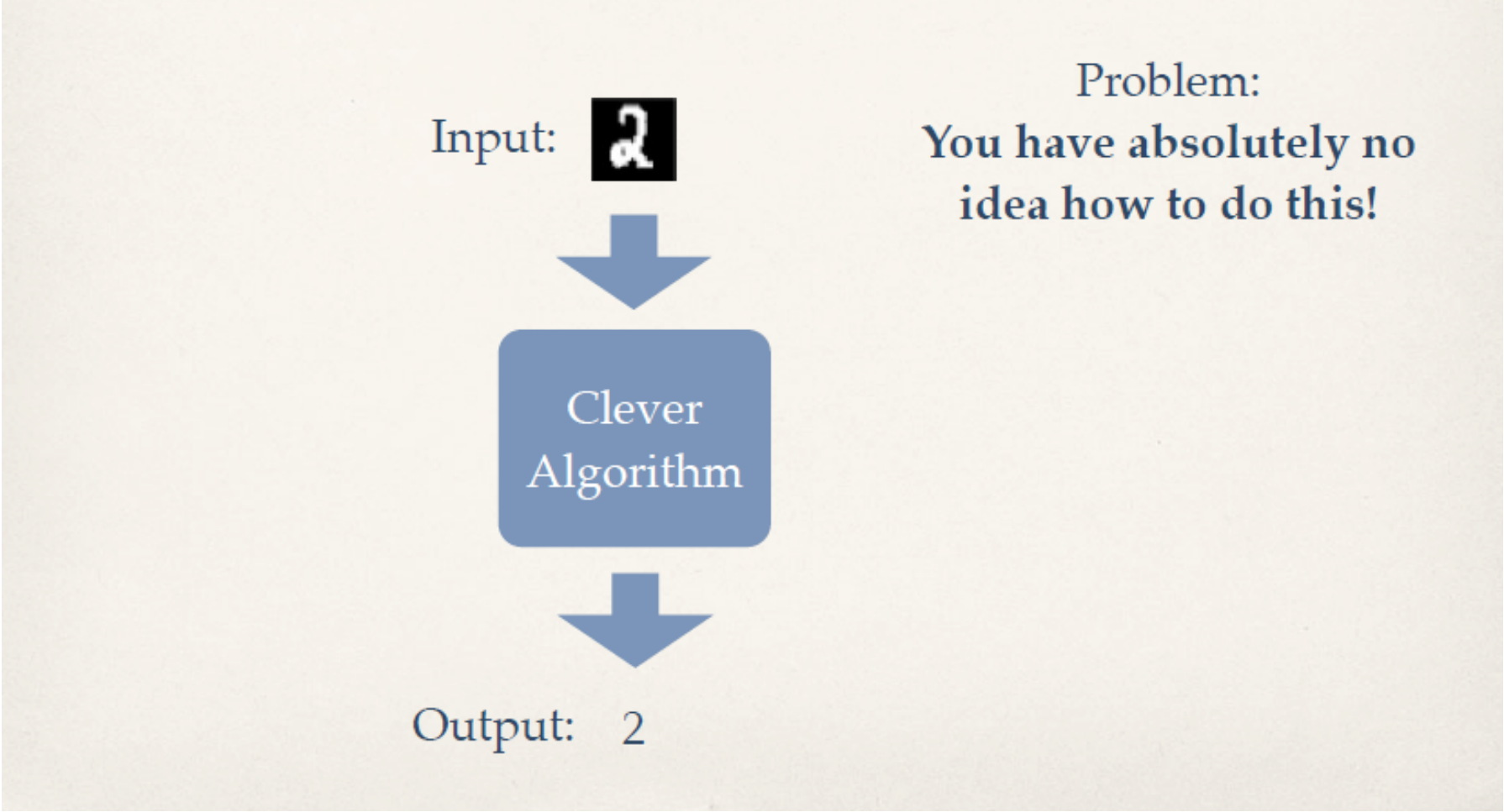
- Experience:
  - Watching you label emails as spam or not spam.

- Performance measure
  - The number (or fraction) of emails (disjoint from the training emails) correctly classified as spam/not spam.

Slide credit: Andrew Ng

Example

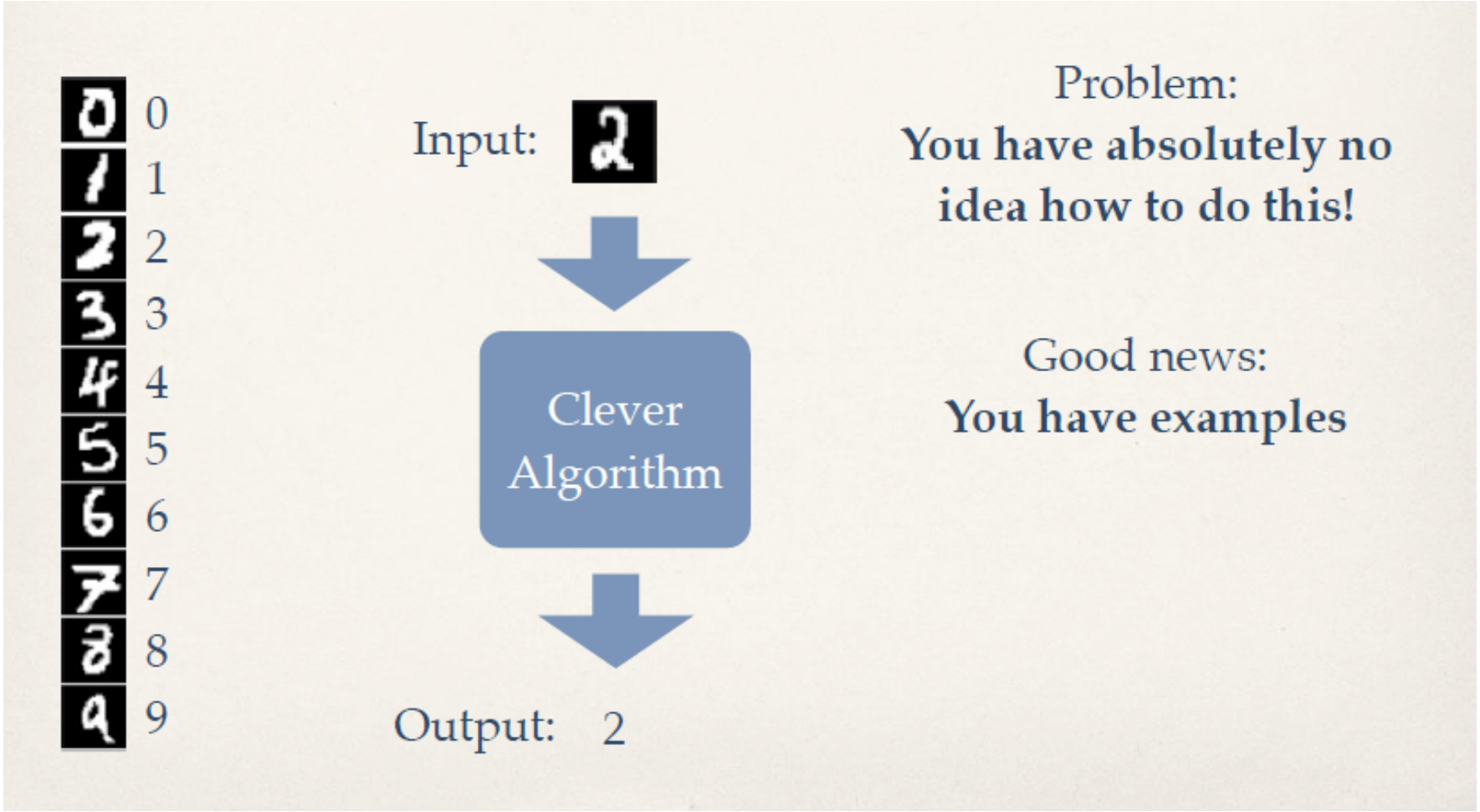
- Problem: Given an image of a handwritten digit, what digit is it?



Slide credit: Kilian Weinberger

Example

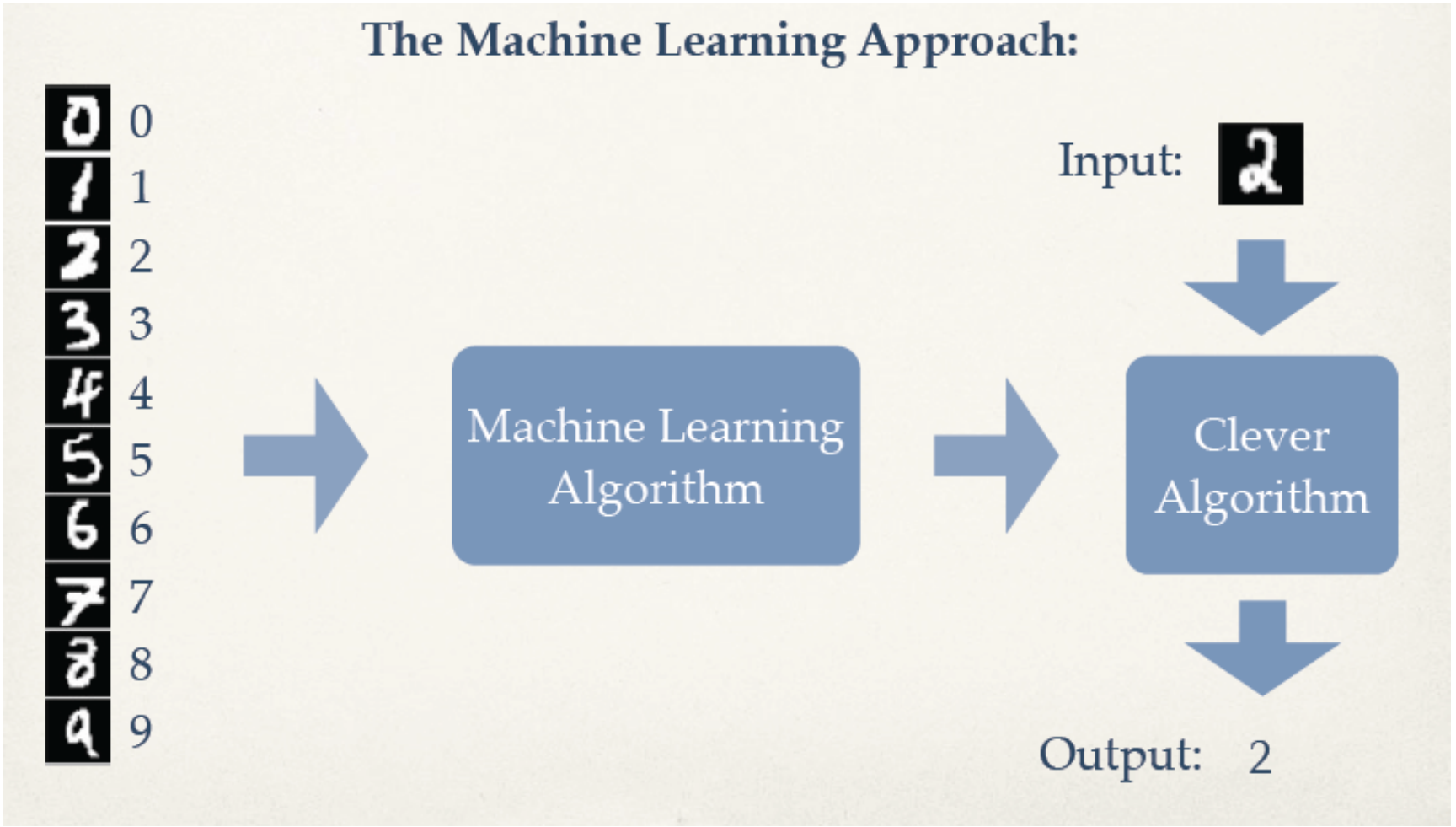
- Problem: Given an image of a handwritten digit, what digit is it?



Slide credit: Kilian Weinberger

Example

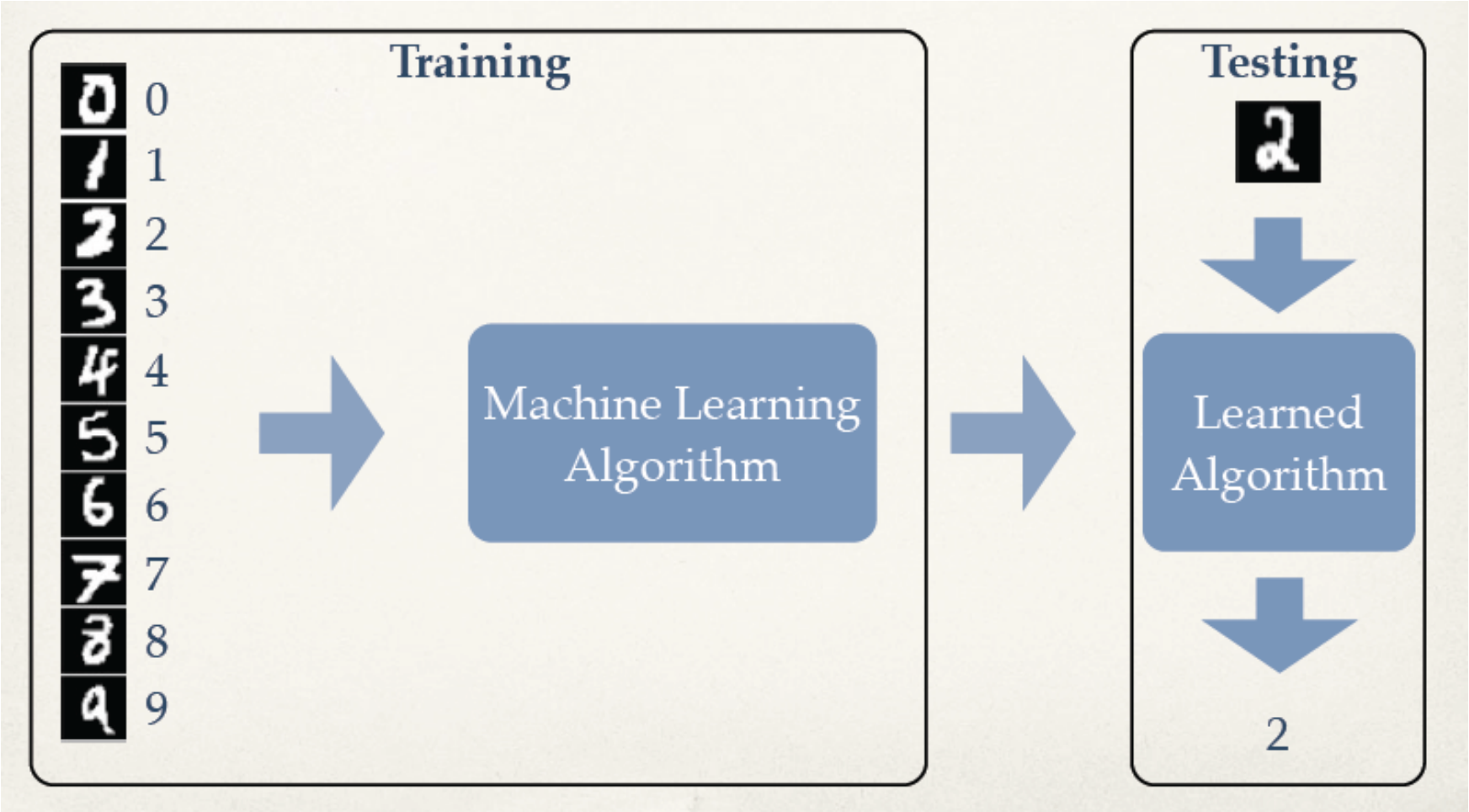
- Problem: Given an image of a handwritten digit, what digit is it?



Slide credit: Kilian Weinberger

Example

- Problem: Given an image of a handwritten digit, what digit is it?



Slide credit: Kilian Weinberger

Machine Learning Tasks

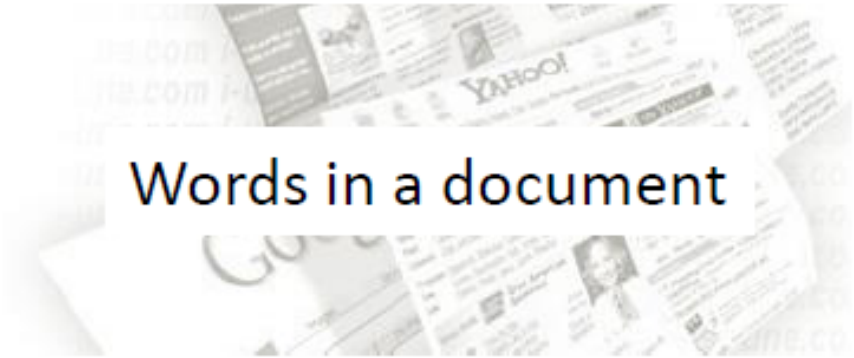
- Supervised Learning
  - Classification
  - Regression
- Unsupervised Learning
  - Clustering
  - Density estimation
  - Embedding / Dimensionality reduction
- Reinforcement Learning
  - Learning to act (e.g., robot control, decision making, etc.)

Supervised Learning

- Goal:
  - Given data X in feature space and the labels Y
  - Learn to predict Y from X
- Labels could be discrete or continuous
  - Discrete labels: **classification**
  - Continuous labels: **regression**

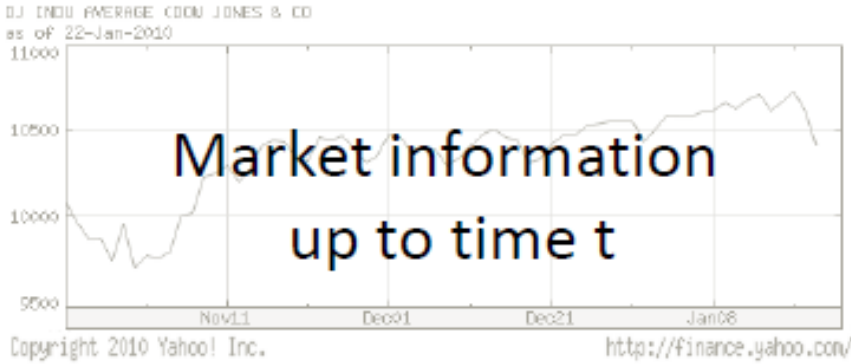
Supervised Learning

Feature Space $\mathcal{X}$



Label Space $\mathcal{Y}$

"Sports"
"News"
"Science"
...



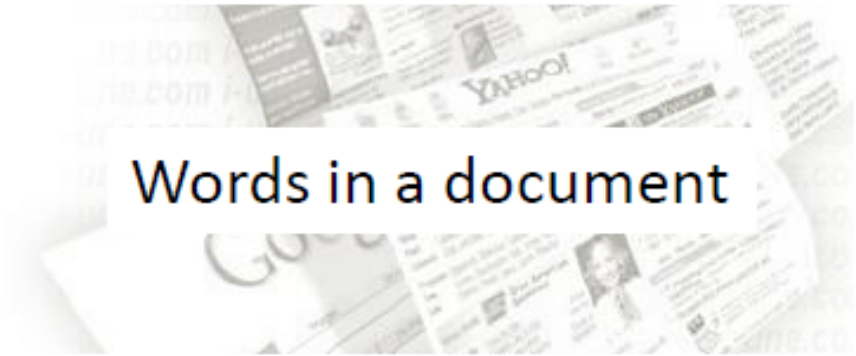
Share Price
"$ 24.50"

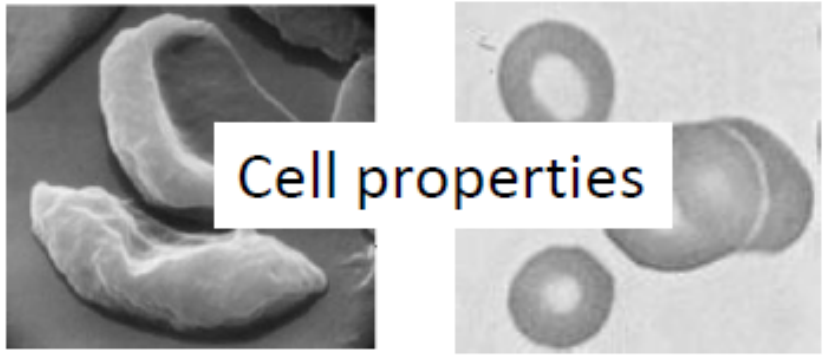
Task: Given $X \in \mathcal{X}$, predict $Y \in \mathcal{Y}$.

Slide credit: Aarti Singh

Supervised Learning - Classification

Feature Space $\mathcal{X}$





Label Space $\mathcal{Y}$

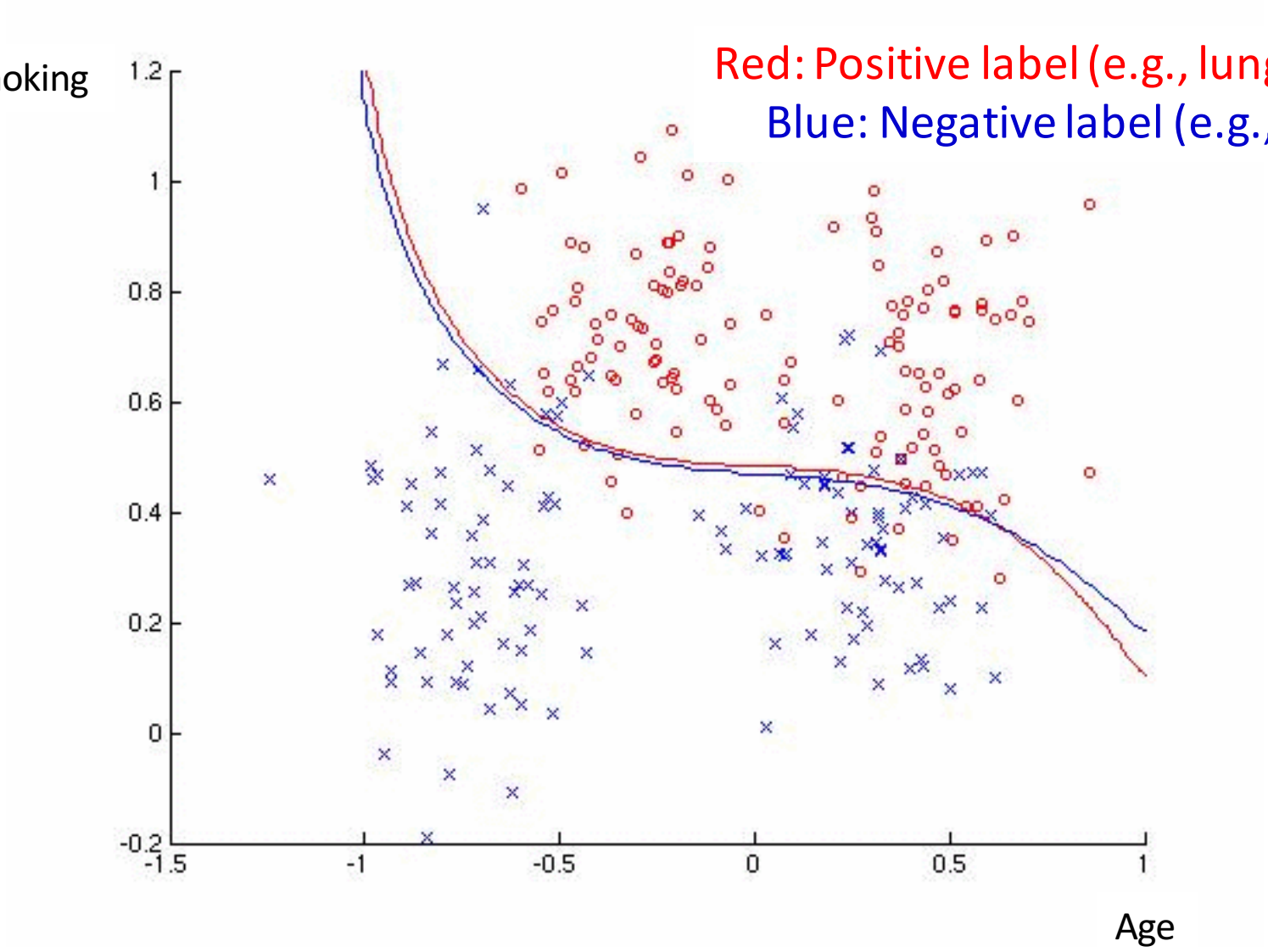
"Sports"
"News"
"Science"
...

"Anemic cell"
"Healthy cell"

Discrete Labels

Slide credit: Aarti Singh

Supervised Learning - Classification

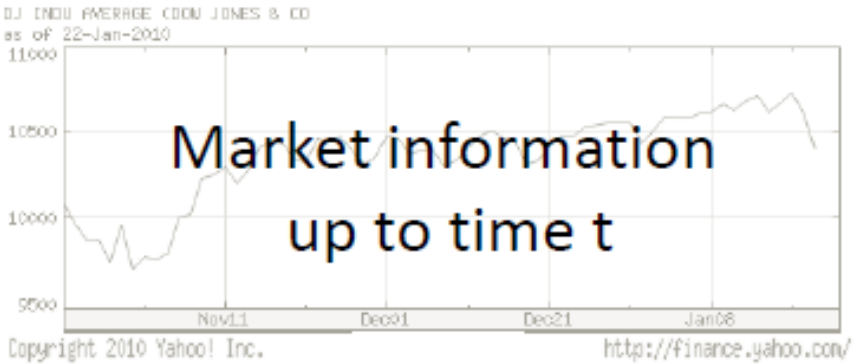


"Learning decision boundaries"

Supervised Learning - Regression

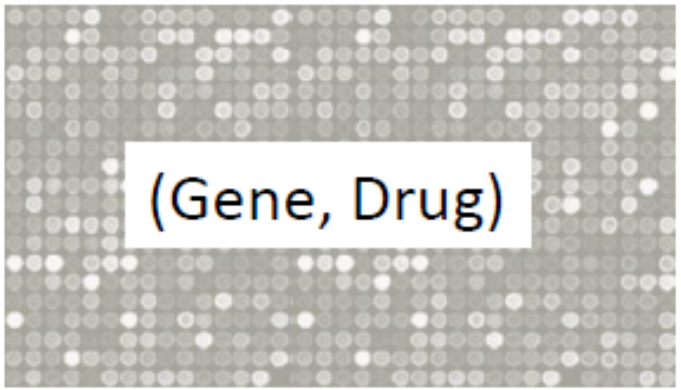
Feature Space $\mathcal{X}$

Label Space $\mathcal{Y}$



Market information up to time t

Share Price
"$ 24.50"

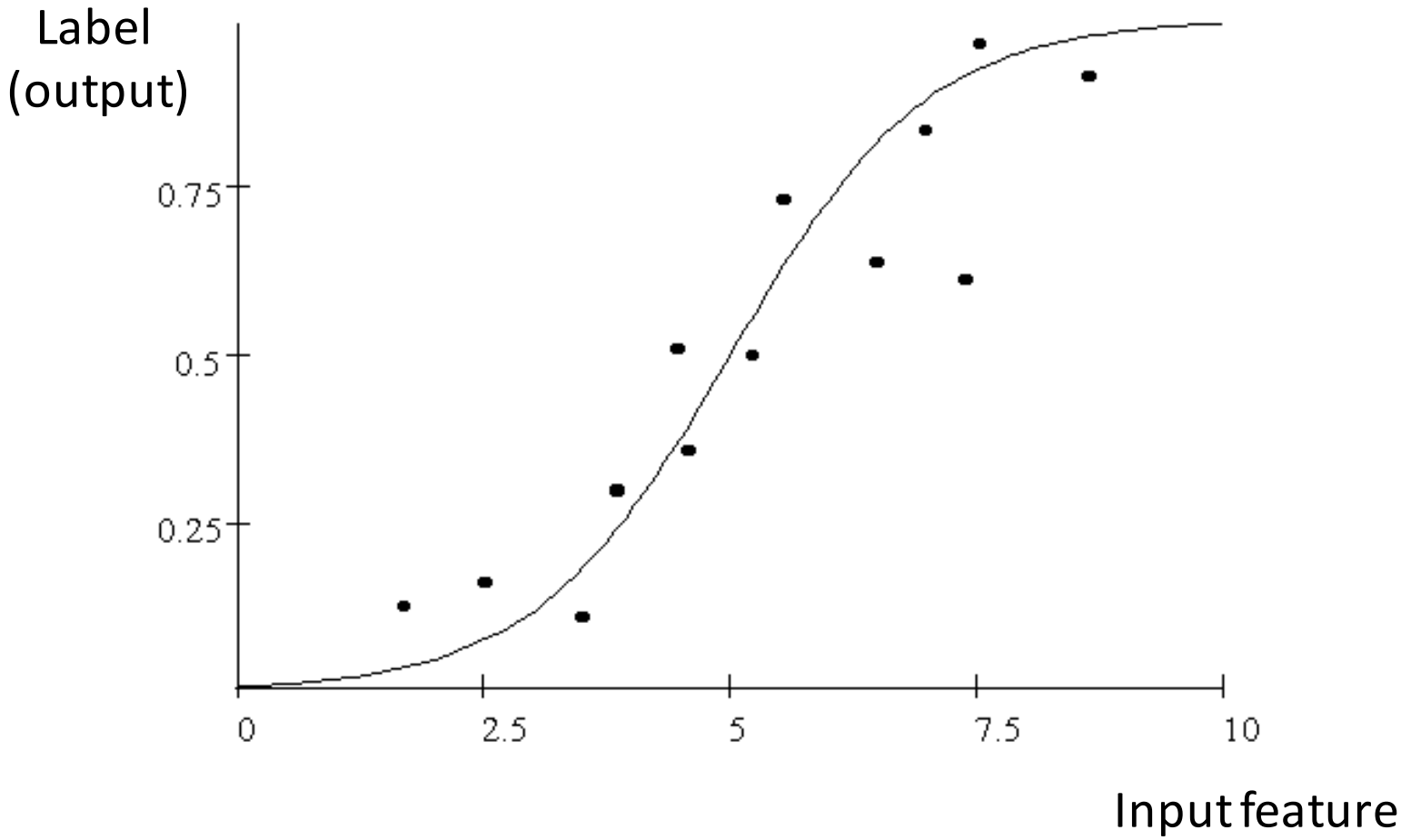


Expression level
"0.01"

Continuous Labels

Slide credit: Aarti Singh

Supervised Learning - Regression



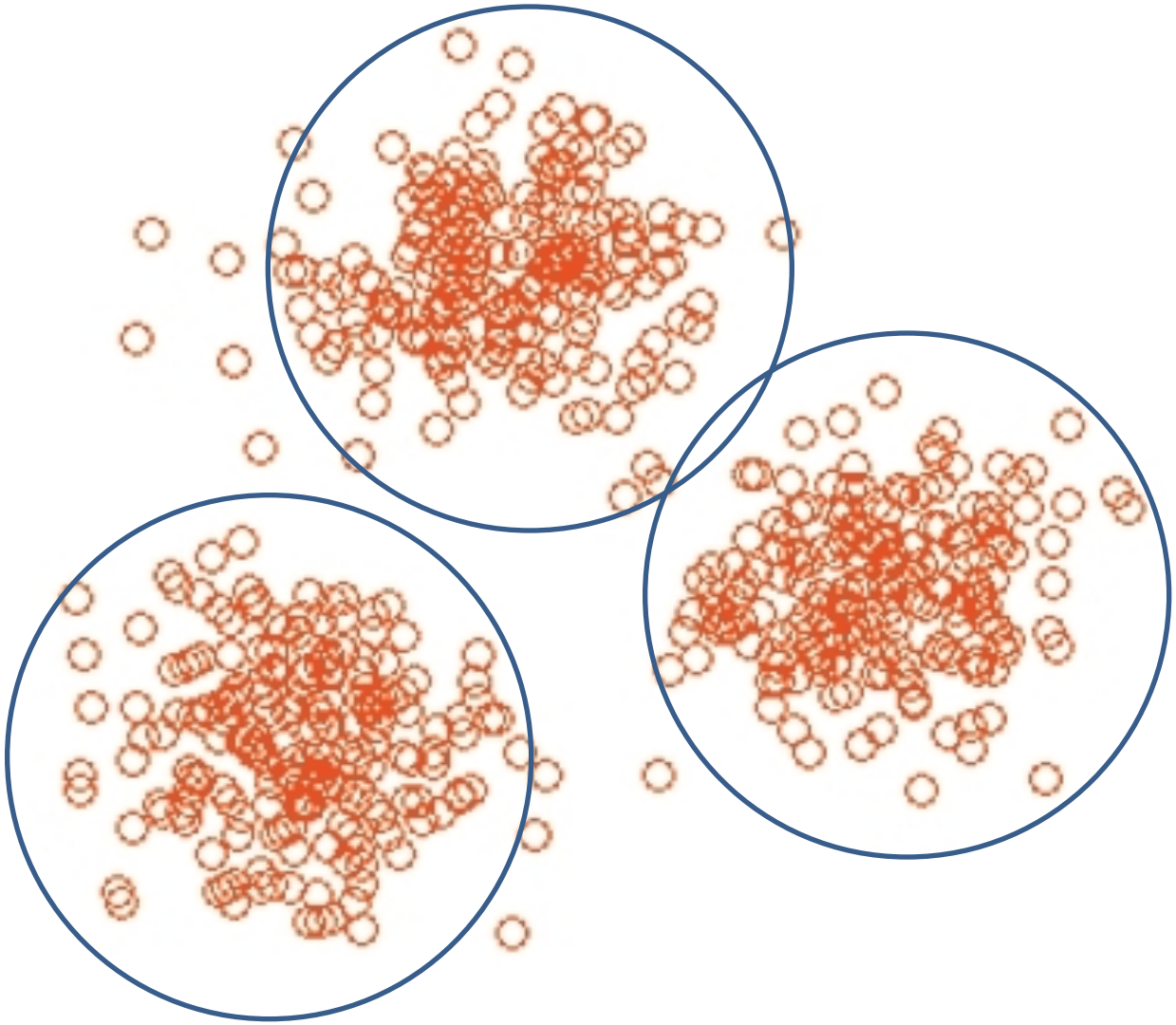
"Learning regression function f(X)"

Unsupervised Learning

- Goal:
  - Given data X without any labels
  - Learn the **structures** of the data
    - Clustering
    - Probability distribution (density)
    - Embedding & neighborhood relations
- "Learning without teacher"

Unsupervised Learning – Clustering

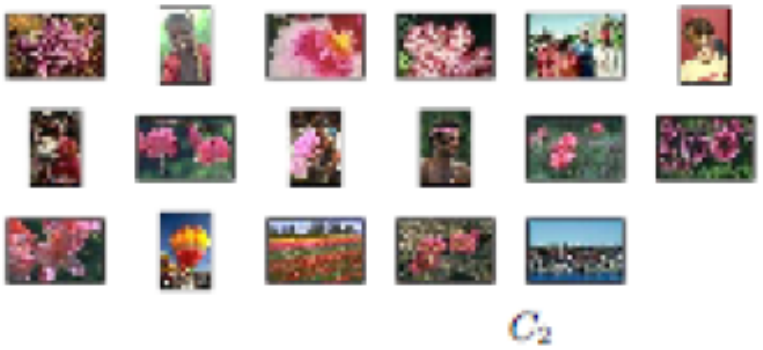
- "Grouping into similar examples"



Unsupervised Learning – Clustering

Group similar things e.g. images

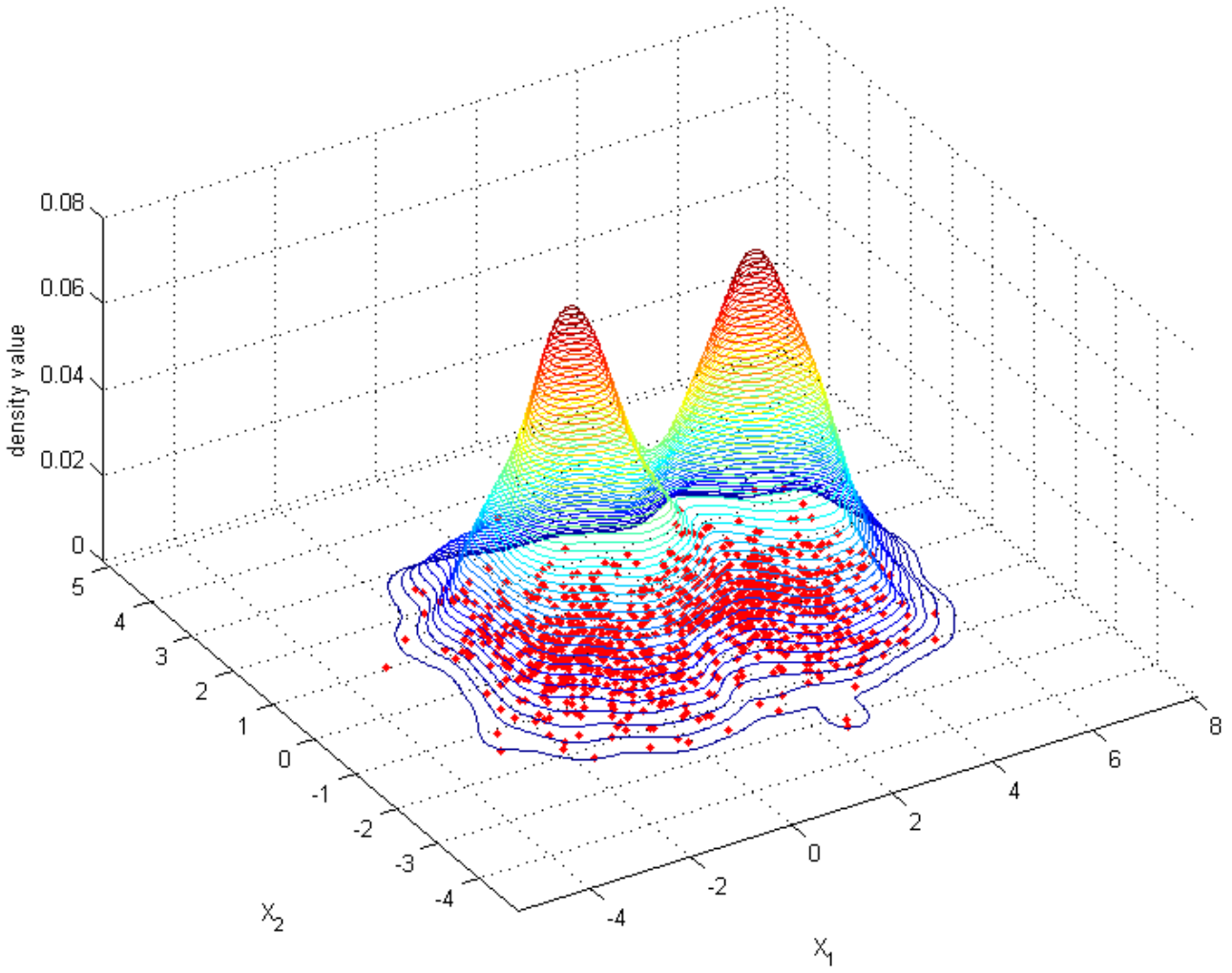
[Goldberger et al.]



Slide credit: Aarti Singh

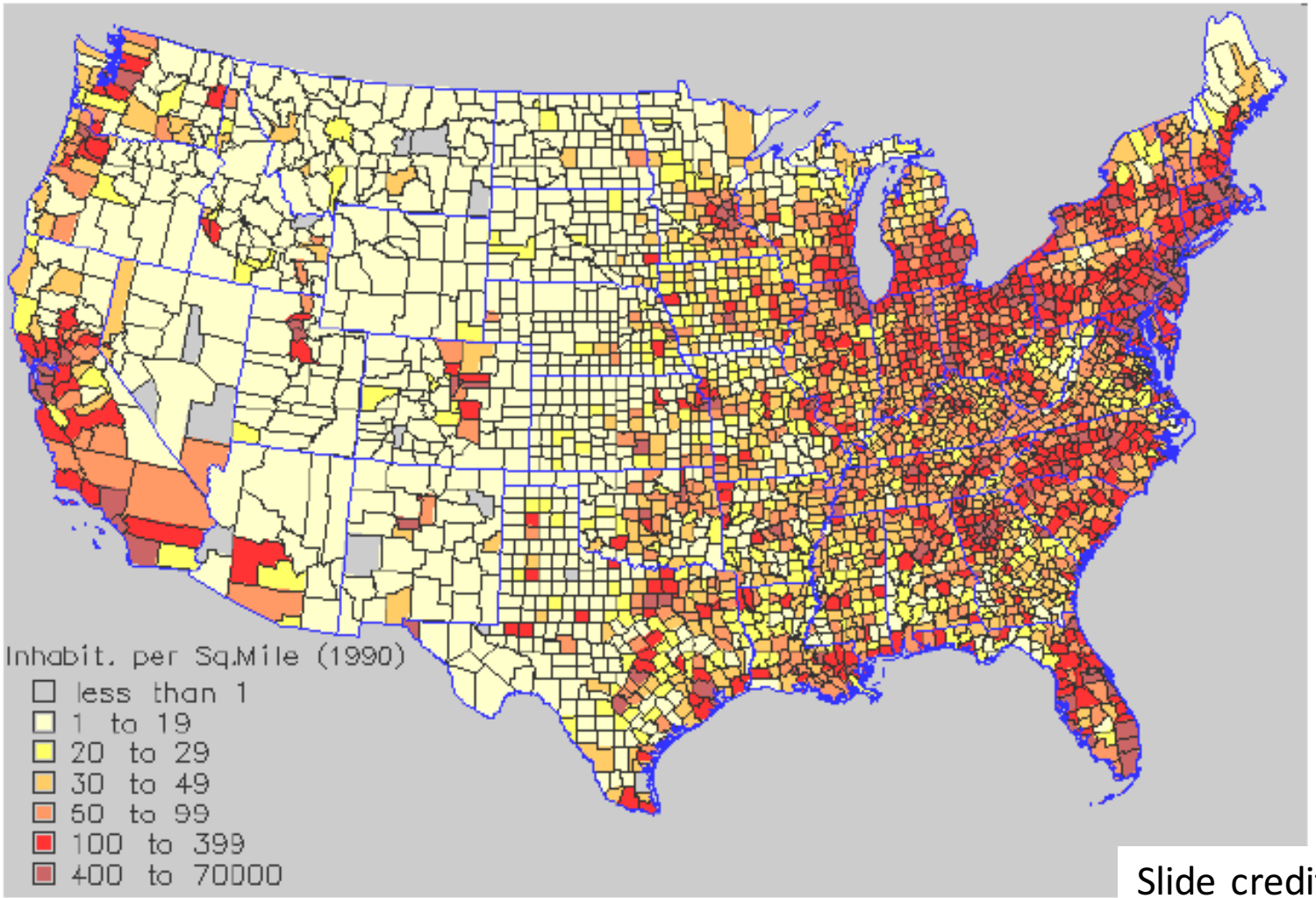
Unsupervised Learning – Density estimation

P(X1, X2)
"Probability"



Unsupervised Learning – Density estimation

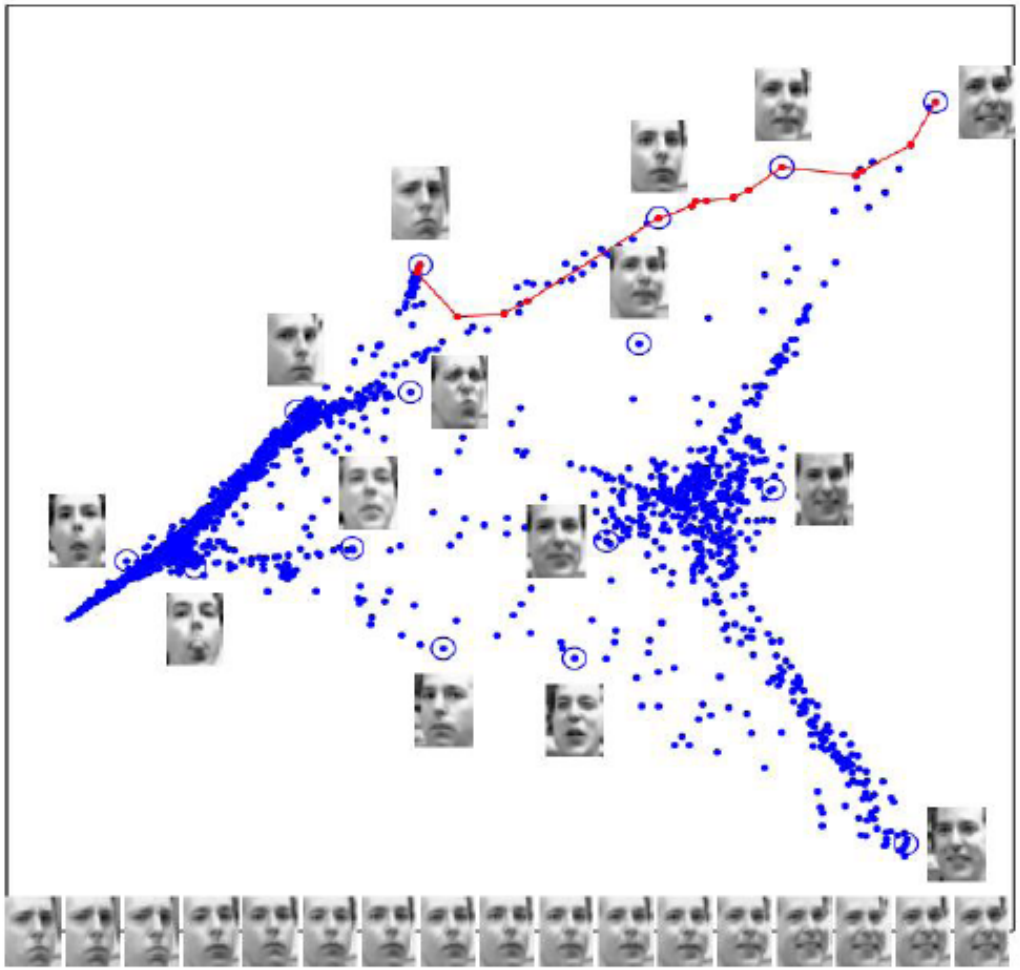
Population density



Slide credit: Aarti Singh

Unsupervised Learning-Embedding and Dimensionality reduction

- E.g., Reducing pixel images (several thousand pixels) into low dimensional coordinates



[Saul and Roweis, 03]

Reinforcement Learning

- Setting
  - Given sequence of states X and "rewards" (e.g., delayed labels)
  - Agent has to take actions A for each time step
- Goal:
  - How to "learn to act" or "make decisions" to maximize the sum of future rewards
- Example: Robot navigation task
  - Input: Dynamical environment + sensor input
  - Action: control signals
  - Rewards: time to reach goal without colliding with obstacles

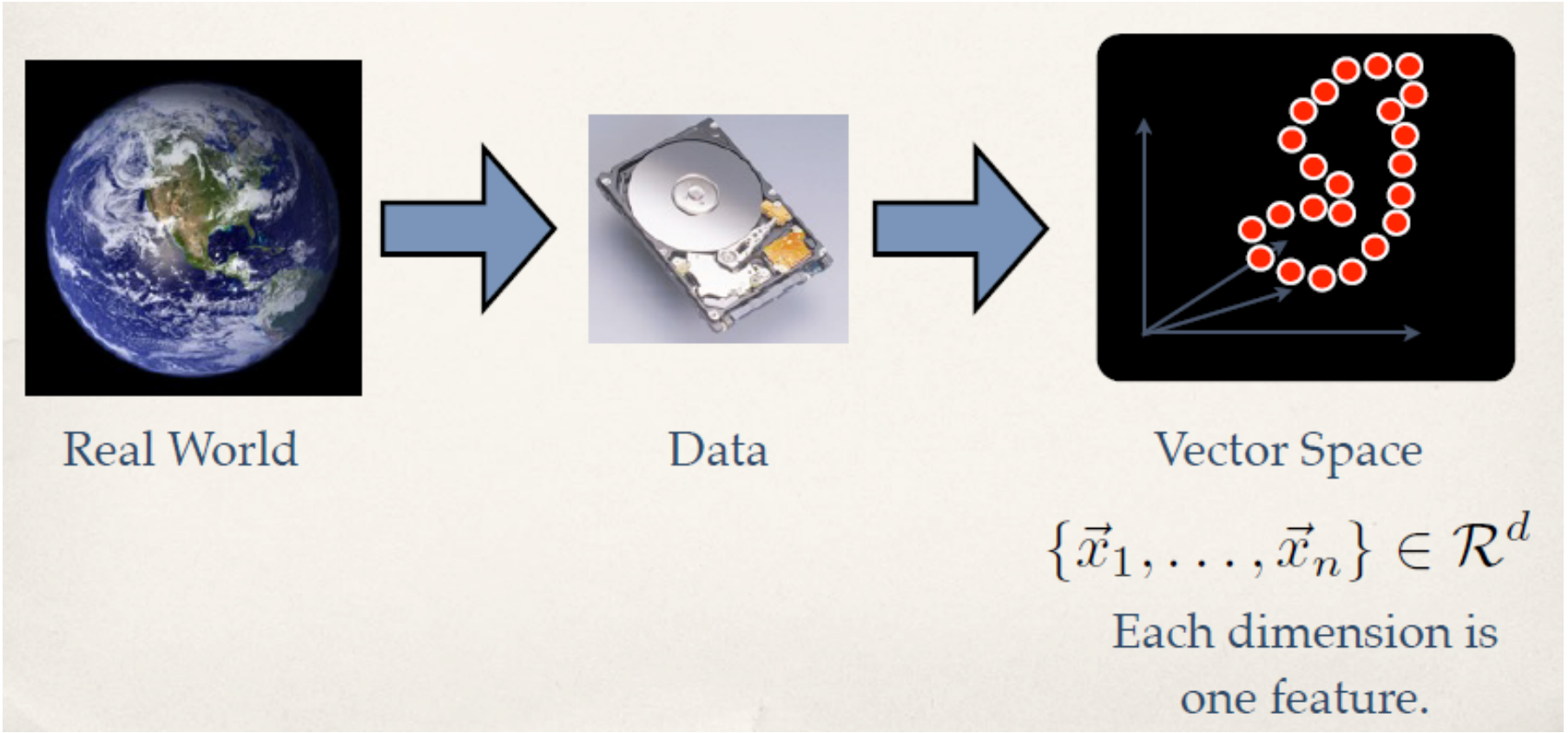
Reinforcement Learning – learning to control

- Example: Robot walking
  - States: sensor inputs, joint angles
  - Action: servo commands for joints
  - Rewards:
    - 1 for reaching the goal
    - -1 for falling down
    - 0 otherwise
- Goal: How can we provide control inputs to maximize the expected future rewards?

Feature representations

Feature Extraction

- Represent data in terms of vectors.
  - Features are **statistics** or **attributes** that describe the data.



Slide credit: Kilian Weinberger

Examples of features: Housing data

- **Given statistics about houses in a local area, predict median value of homes.**
  - #ROOM: average number of rooms per dwelling
  - AREA: average area of house in square foot
  - AGE: proportion of owner-occupied units built prior to 1940
  - CRIME: per capita crime rate by town
  - RESZONE: proportion of residential land zoned for lots over 25,000 sq.ft.
  - INDUS: proportion of non-retail business acres per town
  - NOX: nitric oxides concentration (parts per 10 million)
  - …..
- **Label: Median value of owner-occupied homes**

Examples of features: Recognizing handwritten-digits

- Input: 28x28 pixel digit images
- Output: class labels $\in \{0, \ldots, 9\}$
- The following basic features can be used:
  - Pixel values (784 dimensional vectors)
  - Aspect ratio of the tight bounding boxes
  - Existence of long vertical strokes
  - Existence of long horizontal strokes
  - ….

Learning pipeline

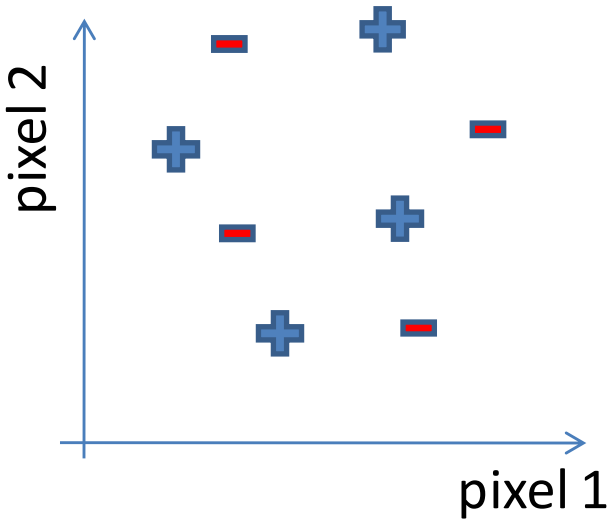
Input

Learning algorithm

Input space

Motorbikes

"Non"-Motorbikes



Learning pipeline

Input

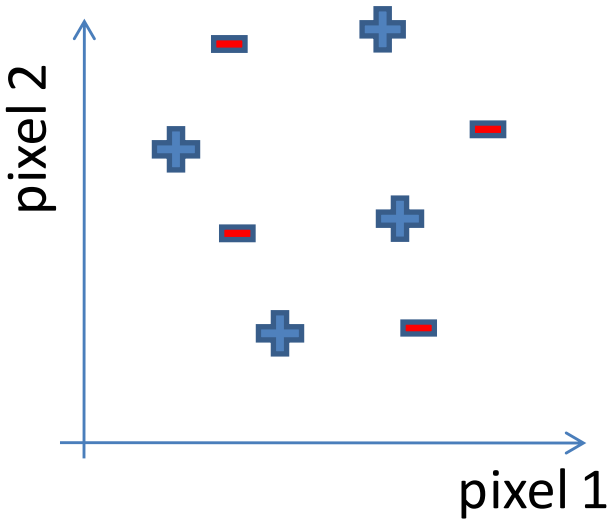
Feature representation

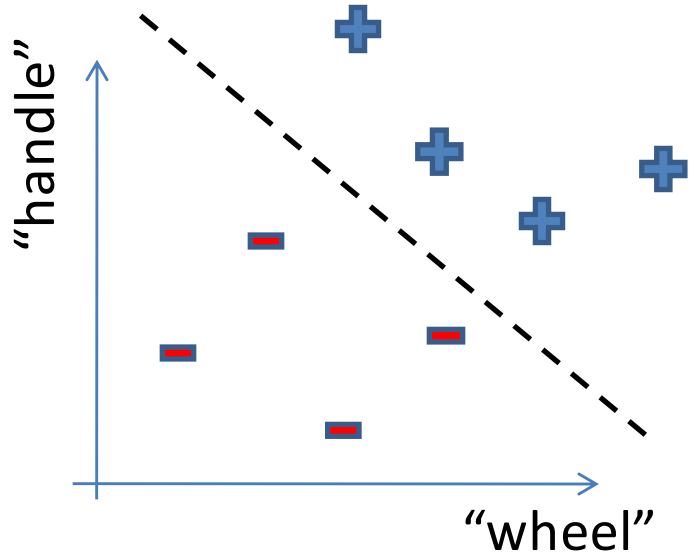
Learning algorithm

Input space

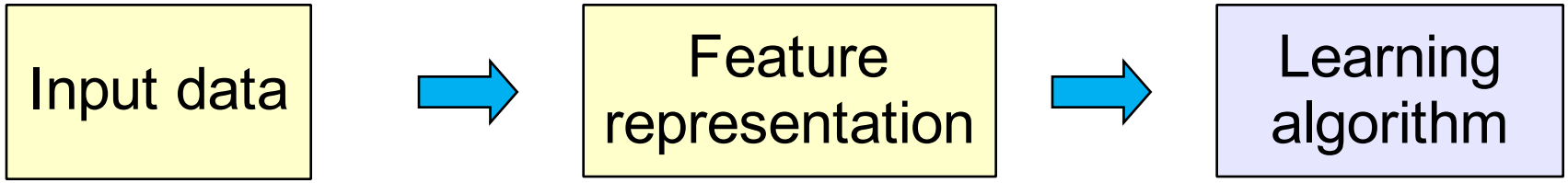
Motorbikes
"Non"-Motorbikes

Feature space



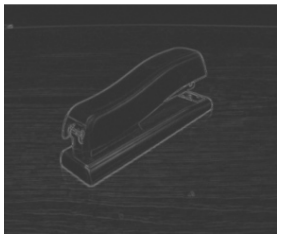


Computer Perception Pipeline



Object detection

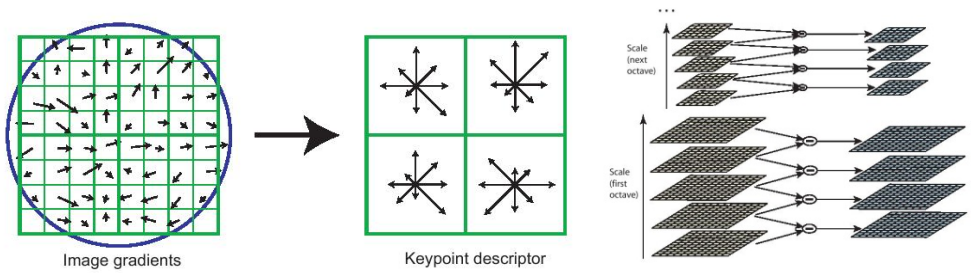
Image



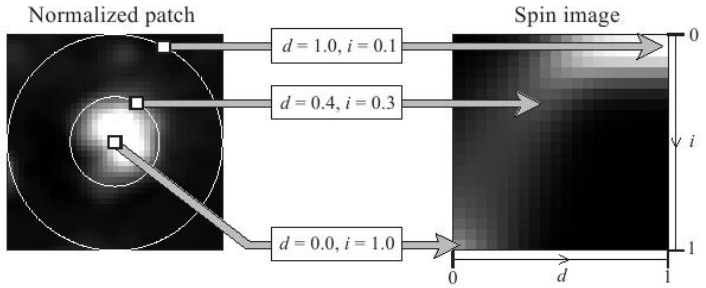
Low-level vision features

Recognition

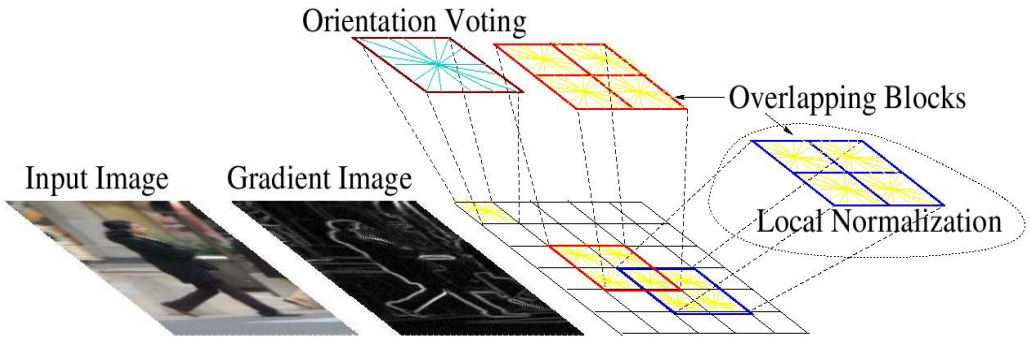
Computer vision features



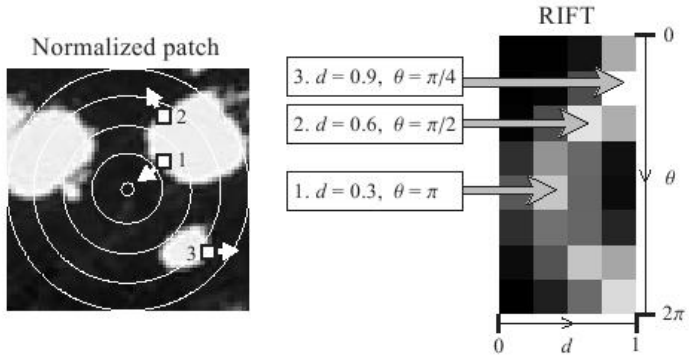
SIFT



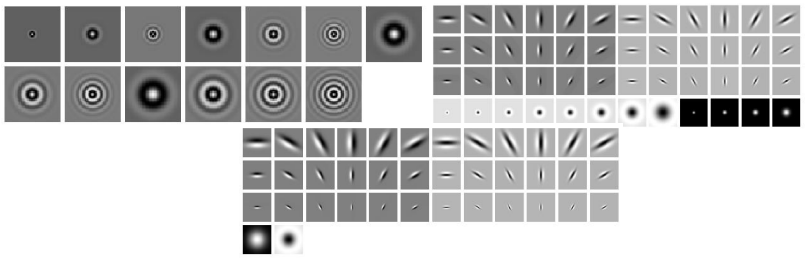
Spin image



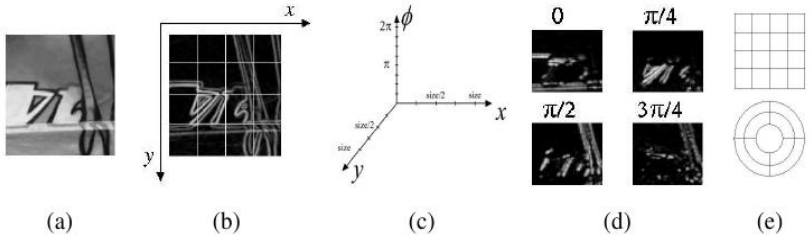
HoG



RIFT

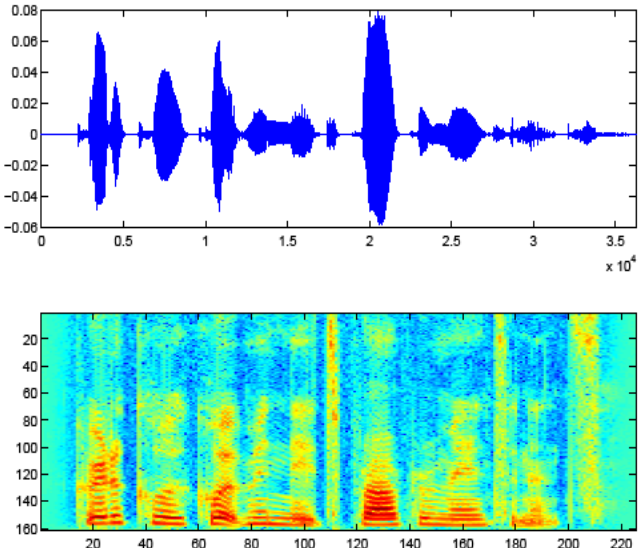


Textons

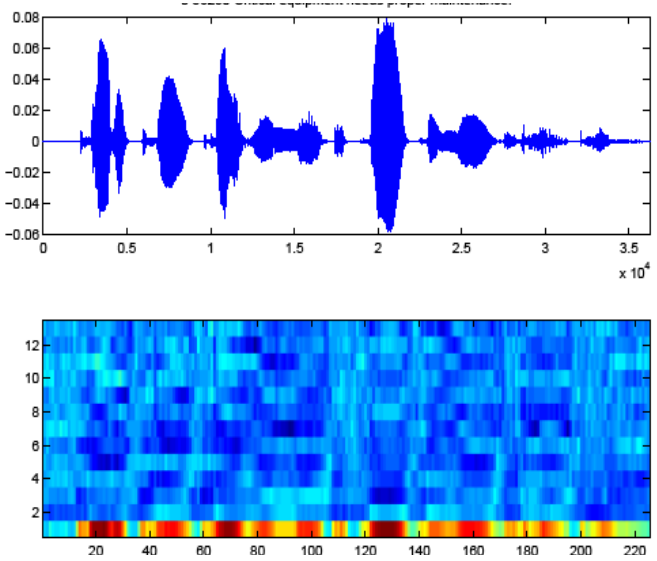


GLOH

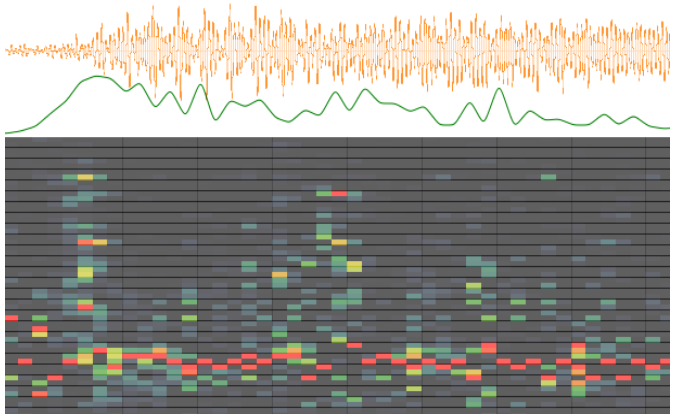
Audio features



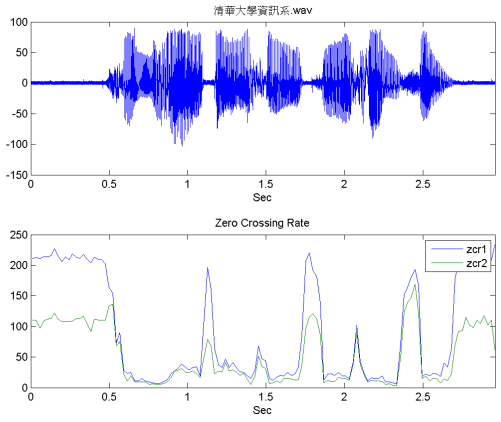
Spectrogram



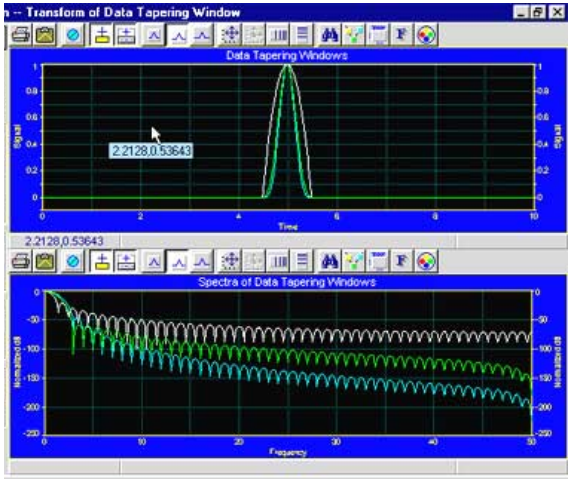
MFCC



Flux



ZCR



Rolloff

Learning Features

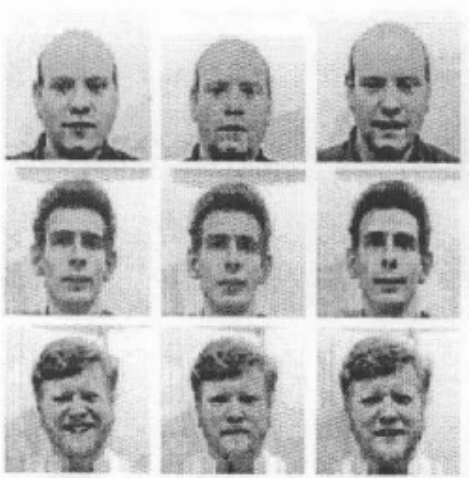
- Most state-of-the-art machine learning systems require careful hand-crafted features.

- However, "learning features" by machine learning has emerged as a very active research recently.
  - "Unsupervised Feature Learning"

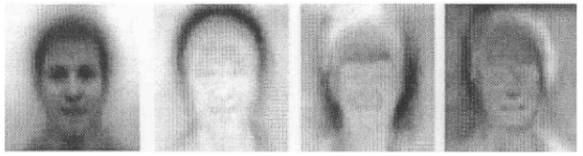
Constructing features via learning

- Example: Eigenfaces

Training face images



Learned PCA bases

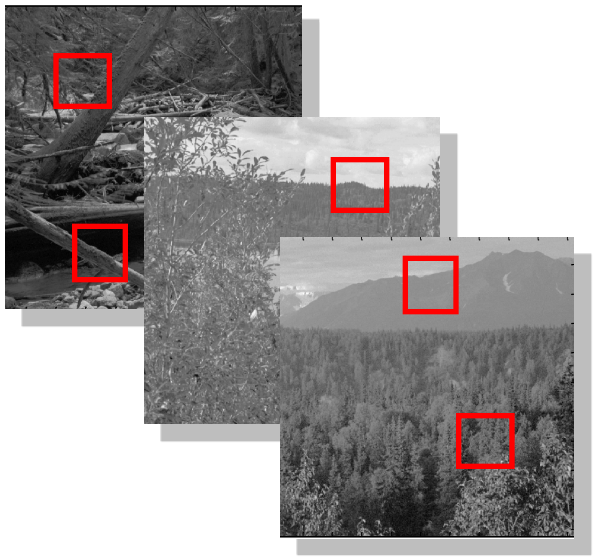


Test example

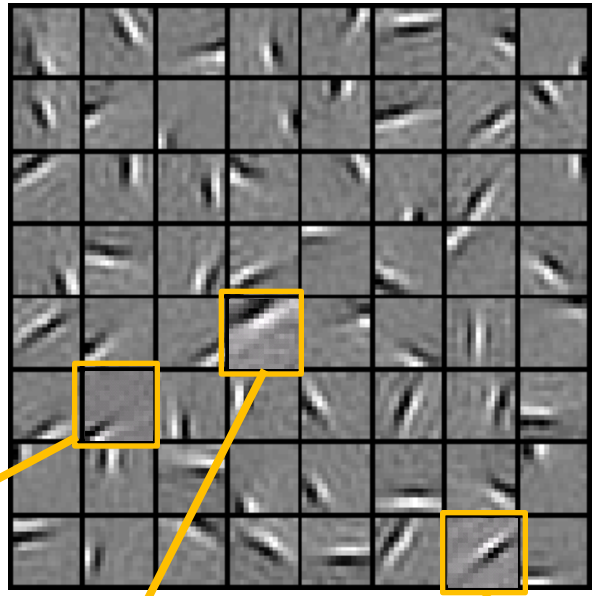
Images from www.cse.unr.edu/~bebis/CS485/Lectures/Eigenfaces.ppt

Learning features via sparse coding

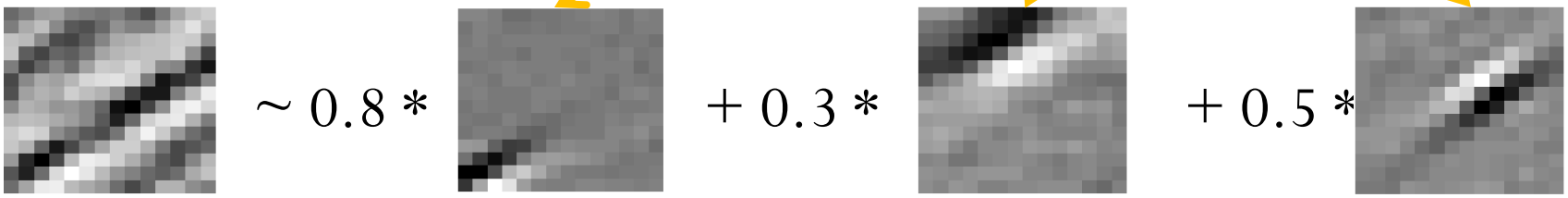
Natural Images



Learned bases: "Edges"



Test example



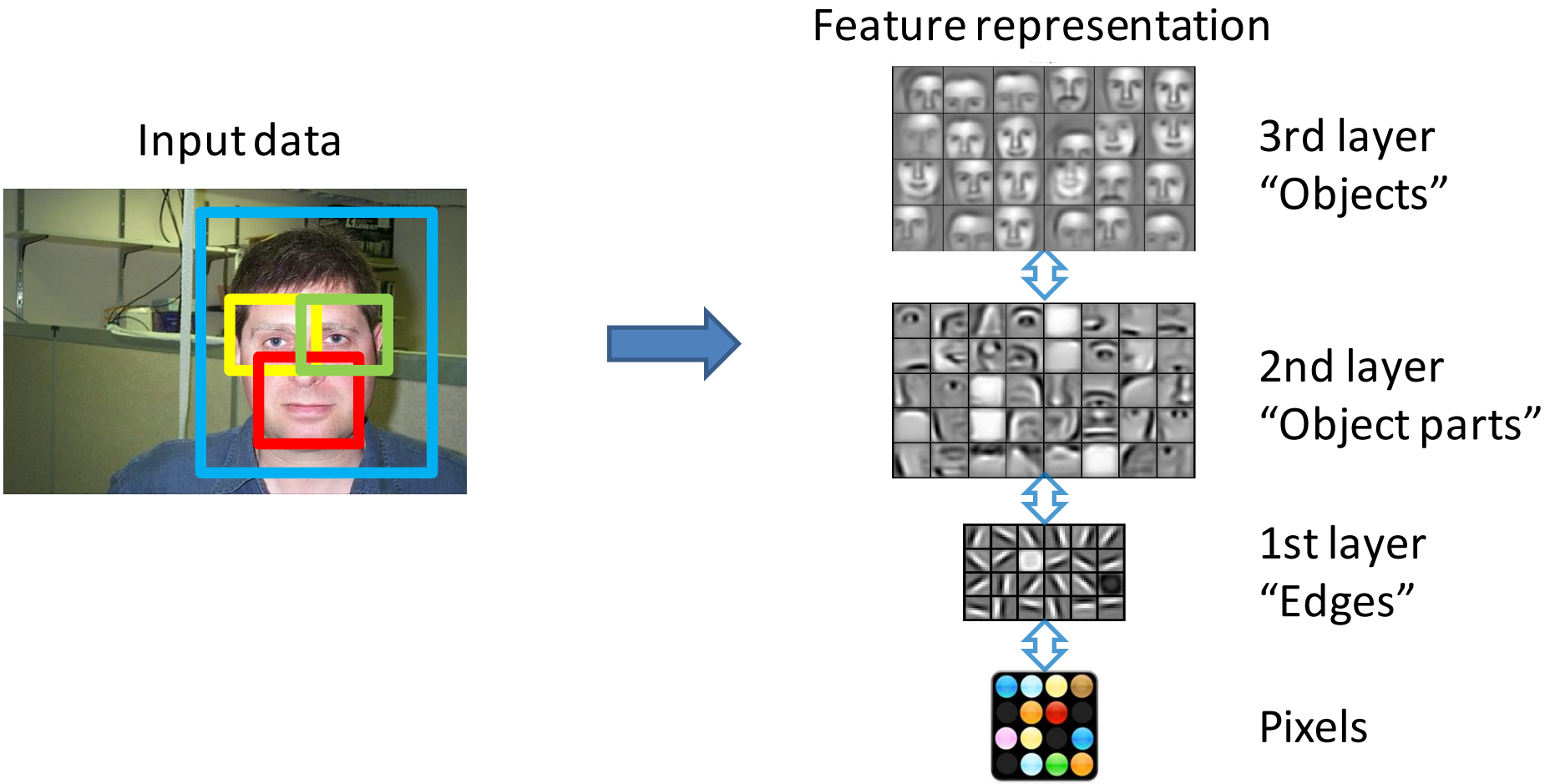
$$x \sim 0.8 * b_{36} + 0.3 * b_{42} + 0.5 * b_{65}$$

[0, 0, …, 0, **0.8**, 0, …, 0, **0.3**, 0, …, 0, **0.5**, …]
= coefficients (feature representation)

Compact & easily interpretable

Learning hierarchy of features

1. Learn high-level “structures” from data.

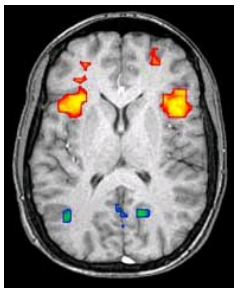


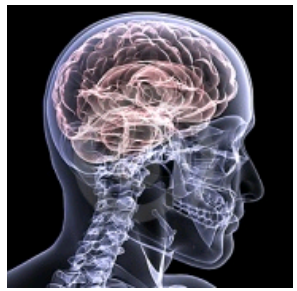
2. Use these features for ML tasks.

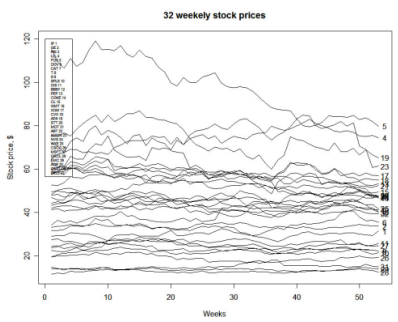
ML applications

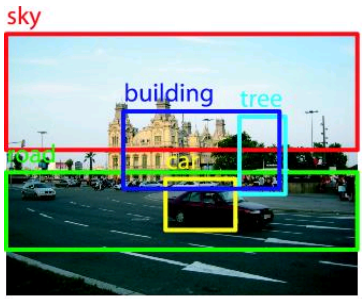
Examples of ML applications

- Text data mining
- Medical image recognition
- Time series prediction/classification
- Computer vision
- Speech recognition
- Robotics
- ….



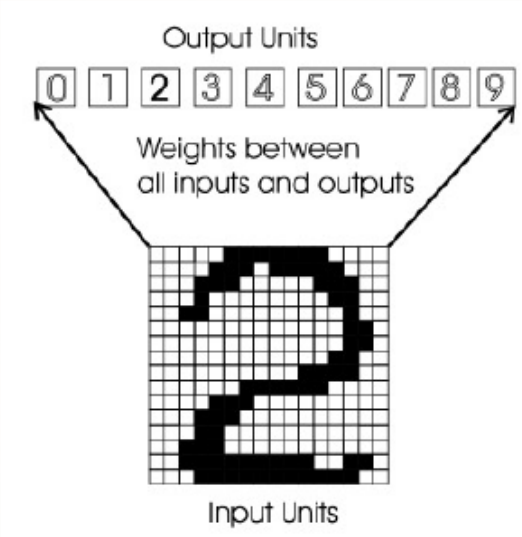




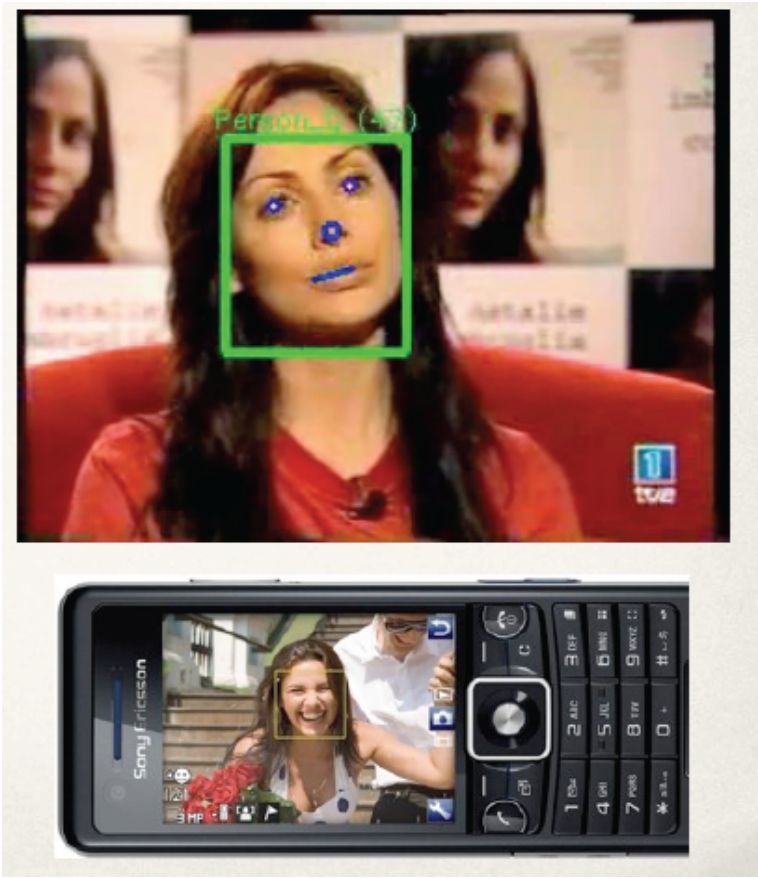


ML application: Computer Vision

- Handwritten digit recognition
  - LeCun et al., 1989



- Face detection
  - Viola & Jones face detector (2001)



ML applications: speech recognition

- Voice search (e.g., Google)

- Speech transcription
  - http://www.youtube.com/watch?v=W3DhnpLIKCQ

ML application: text processing/data mining

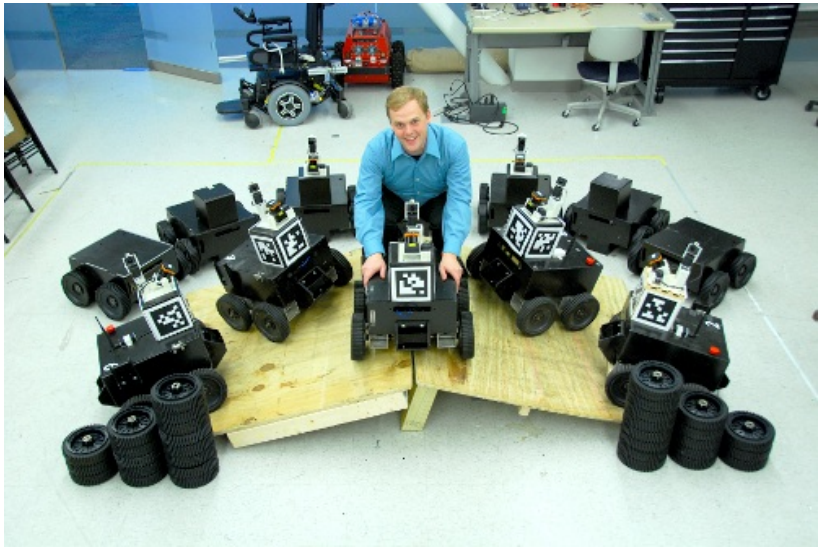
- Spam filtering
  - Given email, predict if it spam or not
- Document clustering
  - Given news articles, group them into different categories (e.g., google news)
- Web search
  - Given query, predict which document will be clicked on.
- Advertisement/product matching
  - Given user info, predict which ad will be clicked on.
  - Given user profile, predict which item user will like to purchase (e.g., Netflix, Amazon, etc.)

ML application: Robotics

- Helicopter control
  - Learn from human experts, but it is now better!
    http://www.youtube.com/watch?v=VCdxqn0fcnE&feature=player_embedded

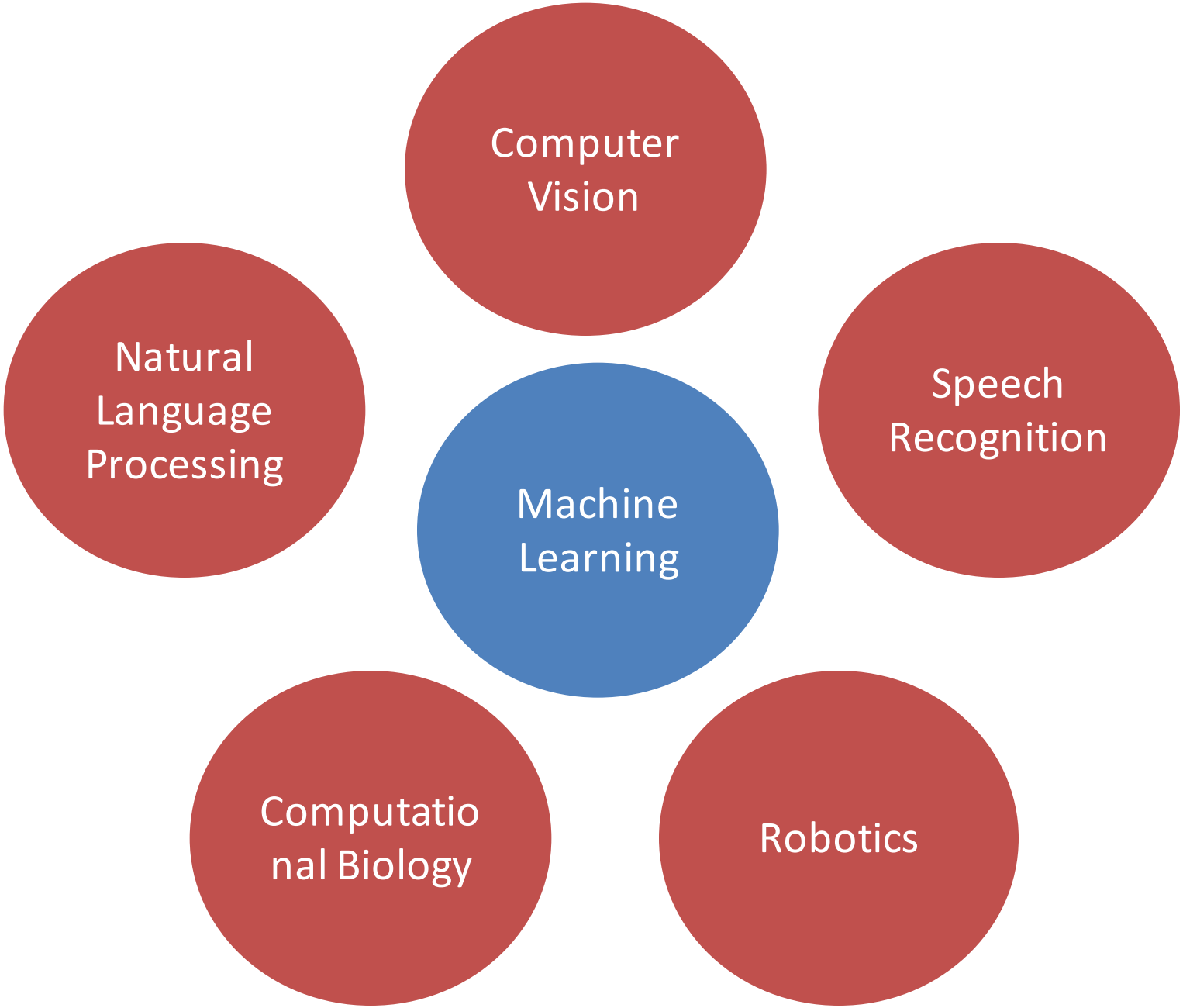
ML application: Robotics

- Robot perception and navigation
- STAIR (Stanford AI robot)
  - http://www.youtube.com/watch?v=mgHUNfqIhAc



- Team Michigan (http://april.eecs.umich.edu/magic/)
  - Prof. Edwin Olson's research team
  - Robot perception + nagivation + multigent coordination
  - Winner of MAGIC competition in 2010! ($750k prize)

Machine Learning and other fields



Next 3-4 classes

- Review of Linear Algebra and Optimization
- Review of Probability and Statistics
- Supervised Learning
  - Linear regression
  - Naïve Bayes
  - Etc.

Reminder

- Check syllabus on Canvas

- For all questions, please use Piazza: http://piazza.com/umich/winter2016/eecs545

Questions?