EECS 545: Machine Learning
Lecture 1. Introduction

Honglak Lee

1/5/2011

Outline

- Administrative

- What is machine leanring?

Teaching staffs

- Instructor: Honglak Lee
  - Email: honglak@eecs.umich.edu
  - Office: CSE 3773
  - Office hour: TBD (or email appointment)
- Graduate Student Instructor: James Boerkoel
  - Email: boerkoel@umich.edu
  - Office hours: TBD
  - Will hold review sessions on background materials (linear algebra, probability, Matlab, etc.)
- For all questions, please send email to eecs545qa@umich.edu!

Online syllabus and survey

- Check syllabus at
  - http://www.eecs.umich.edu/~honglak/teaching/eecs545
- Please fill out the online survey by 5pm today.
  - Required for enrollment
  - https://spreadsheets.google.com/viewform?hl=en&formkey=dHpmUXpsSXpvWEsxcWp2WGoyclNkMWc6MQ#gid=0

Text books

- Chris Bishop, “Pattern Recognition and Machine Learning”. Springer, 2006.
  - http://research.microsoft.com/en-us/um/people/cmbishop/prml/
- (optional) Sutton and Barto, “Reinforcement Learning: An Introduction”. MIT Press, 1998
  - (Free: MIT cognet) http://cognet.mit.edu/library/books/view?isbn=0262193981
  - http://searchtools.lib.umich.edu/V?func=native-link&resource=UMI02069
- (optional) Hastie, Tibshrani, Fiedman, “Elements of Statistical Learning”. Springer, 2010.
  - (Free) http://www-stat.stanford.edu/~tibs/ElemStatLearn/download.html
- (optional) Mackay, “Information Theory, Inference, and Learning Algorithms”. Cambridge University Press. 2003.
  - (Free) http://www.inference.phy.cam.ac.uk/itprnn/book.pdf

Prerequisites

- EECS 492: Introduction to AI
- Undergrad linear algebra (e.g., MATH 217, MATH 417)
- Undergrad probability and statistics (e.g., EECS 401)
- Programming skills (equivalent to EECS 280, EECS 281, and experience in MATLAB)
  - Nontrivial level of programming is required.
- Self-check:
  - Read Bishop Sections 1.1, 1.2, and 1.3 carefully.
  - Read Appendix C & E.

Grading policy

- Homework: 40%
- Midterm: 15%
- Project: 45%
  - progress report (10%)
  - final report (35%)
- Extra credits:
  - Up to 2% may be awarded for class participation.
  - 1% will be given for lecture scribing (more details later).

More about lecture scribing

- 1% will be given for lecture scribing.
- Student can scribe at most one lecture.
  - Sign up form will be available (google doc)
- <u>NOT</u> all lectures will be scribed. Instructor will designate lectures for scribing.
  - <u>Purpose: put detailed derivations and useful notes.</u>
  - Handwritten lecture note and other materials will be provided for scribing.
  - A latex template will be available. (Good opportunity to improve your latex skills)

Homework

- There will be five (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 Wednesday (1/12).

Study group

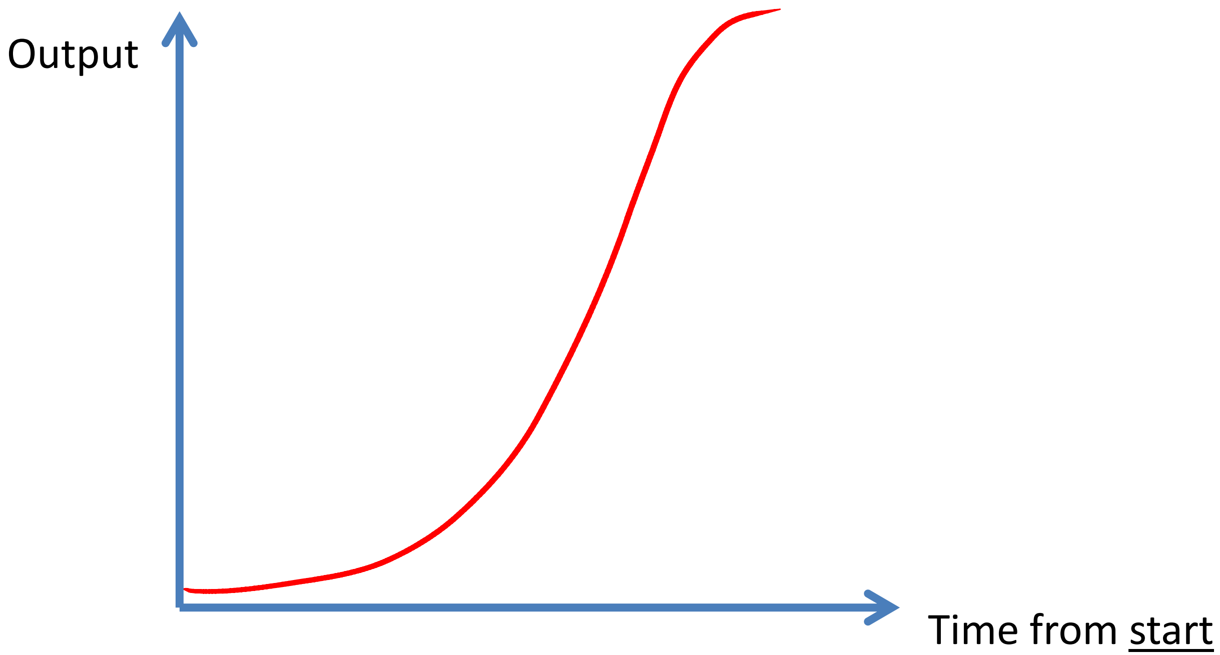
- Form your study group early on!
  - Up to <u>three</u> people are allowed.
- For homework, you may discuss between the study group members, but you should write your own solution <u>independently</u>.
- You must put the names of the other members in your homework submissions.
- Please start on homework early. (Warning: cramming does not work!)

Course Project

- Scope
  - develop new theory and algorithms in machine learning,
  - apply existing algorithms to new problems,
  - apply to their own research problems.
- Milestones
  - project proposals
  - progress reports
  - poster presentations and the final report. (~4/25)
- Requirement
  - Write a 8-page paper
  - Submit the final code
  - Give a poster presentation
- Evaluation is based on the quality of the project.

Course Project

- Up to three people can form a project group.
- Talk to instructor if you want to get suggestions about project topics.
- Start early!!! (form your group and start working)



Any questions?

Outline

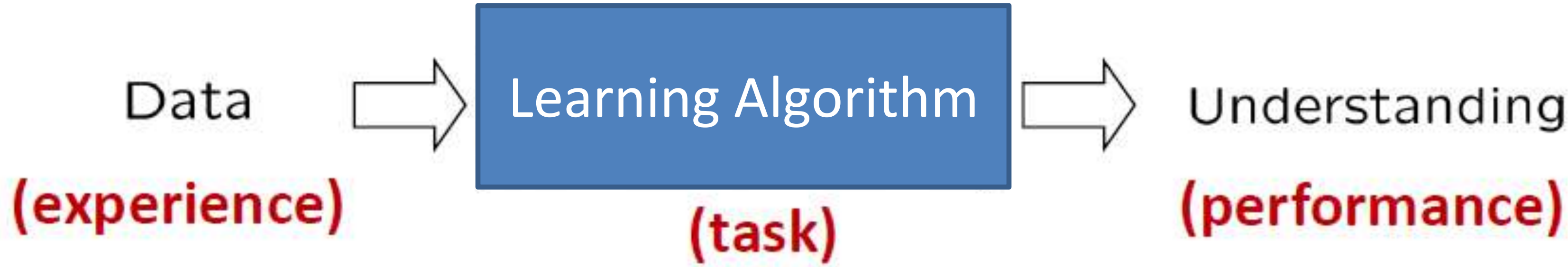
- Administrative
- What is machine learning?

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 performance at some task with experience.



Example:

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

Input: 

Clever Algorithm

Output: 2

Problem:

You have absolutely no idea how to do this!

Slide credit: Kilian Weinberger

Example:

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

Problem:

You have absolutely no idea how to do this!

Good news:

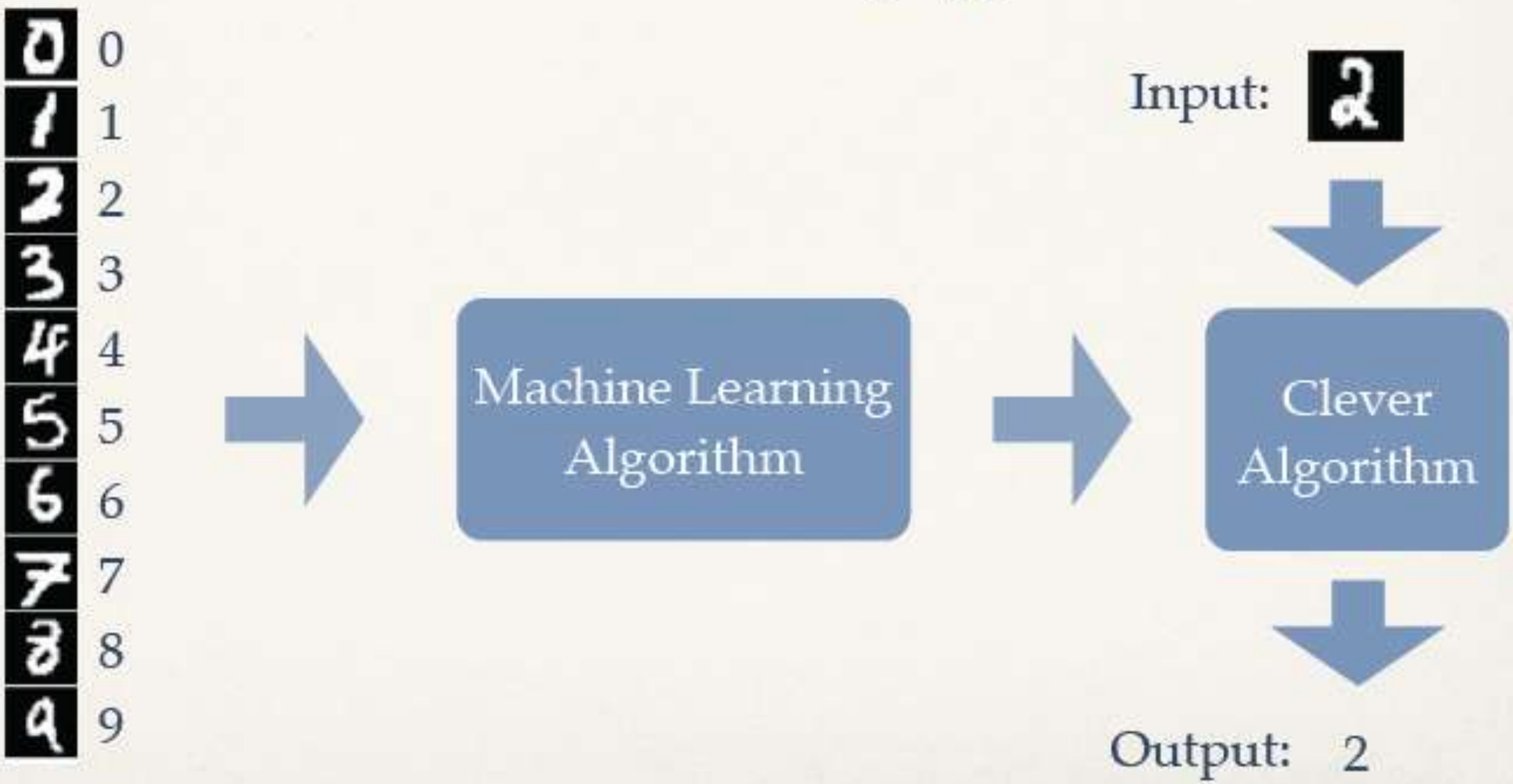
You have examples

Slide credit: Kilian Weinberger

Example:

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

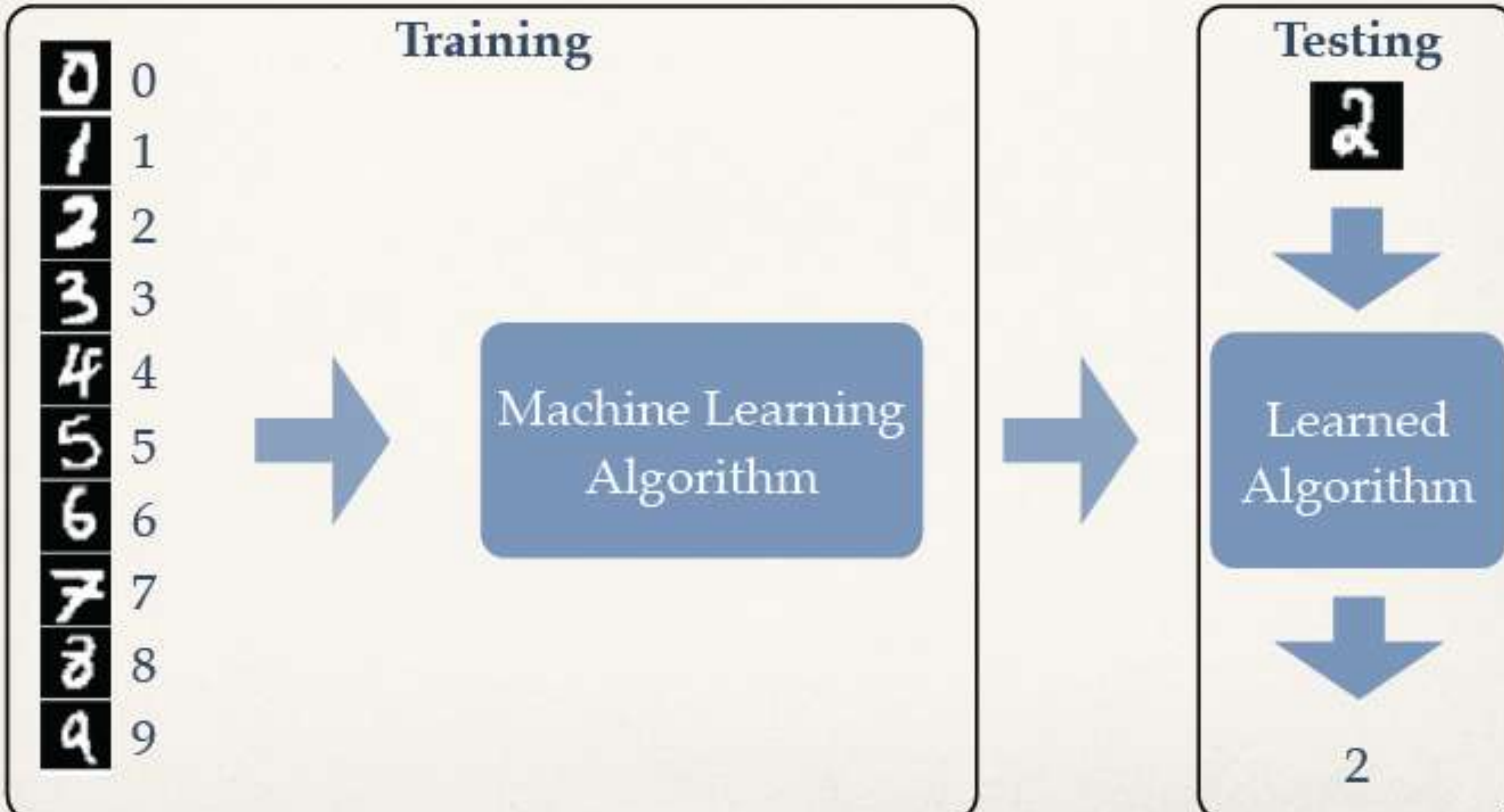
The Machine Learning Approach:



Slide credit: Kilian Weinberger

Example:

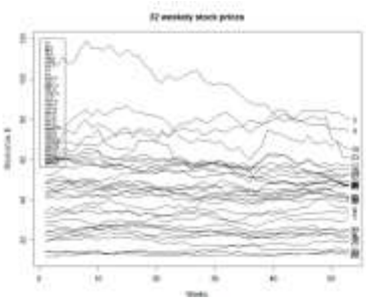
Problem: Given an image of a handwritten digit, what digit is it?

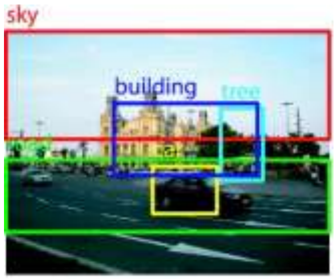


Slide credit: Kilian Weinberger

Examples of ML applications

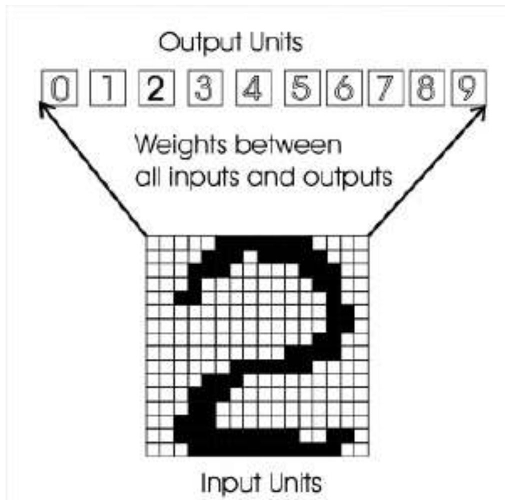
- Text data mining
- Understanding fMRI data
- Stock price prediction
- Computer vision
- Speech recognition
- Robotics
- ….



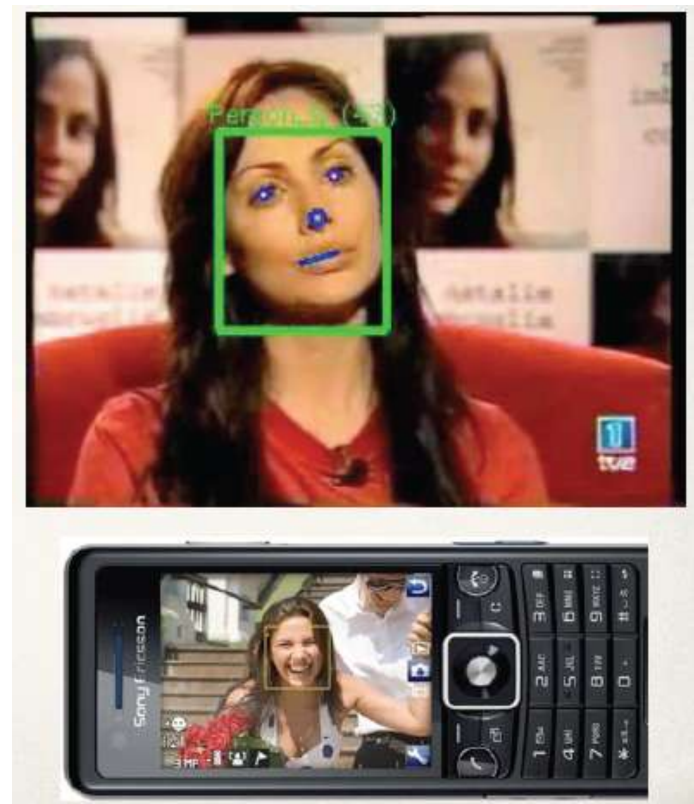


ML application: Computer Vision

- Handwritten digit recognition
  - LeCun et al., 1989



- Face recognition
  - 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

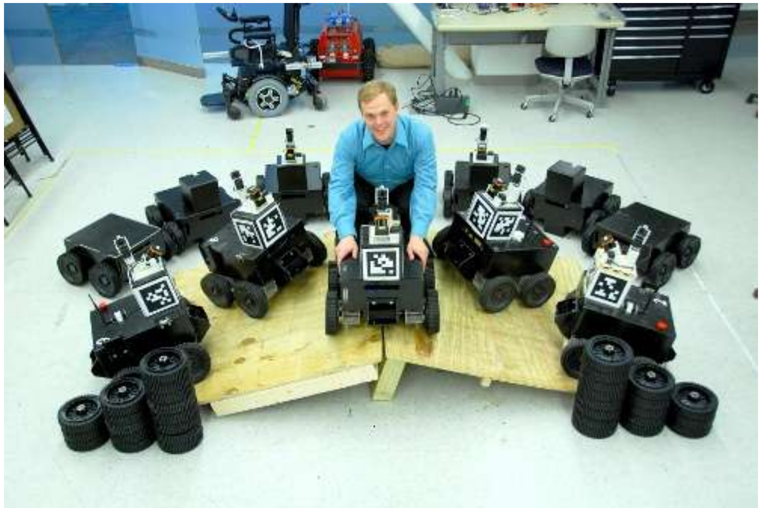
- Spam filtering
  - Given email, predict if it spam or not
- Document clustering
  - Given news articles, group them into different categories
- Web Search
  - Given query, predict which document will be clicked on.
- Advertisement matching
  - Given user info, predict which ad will be clicked on.

ML application: Robotics

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

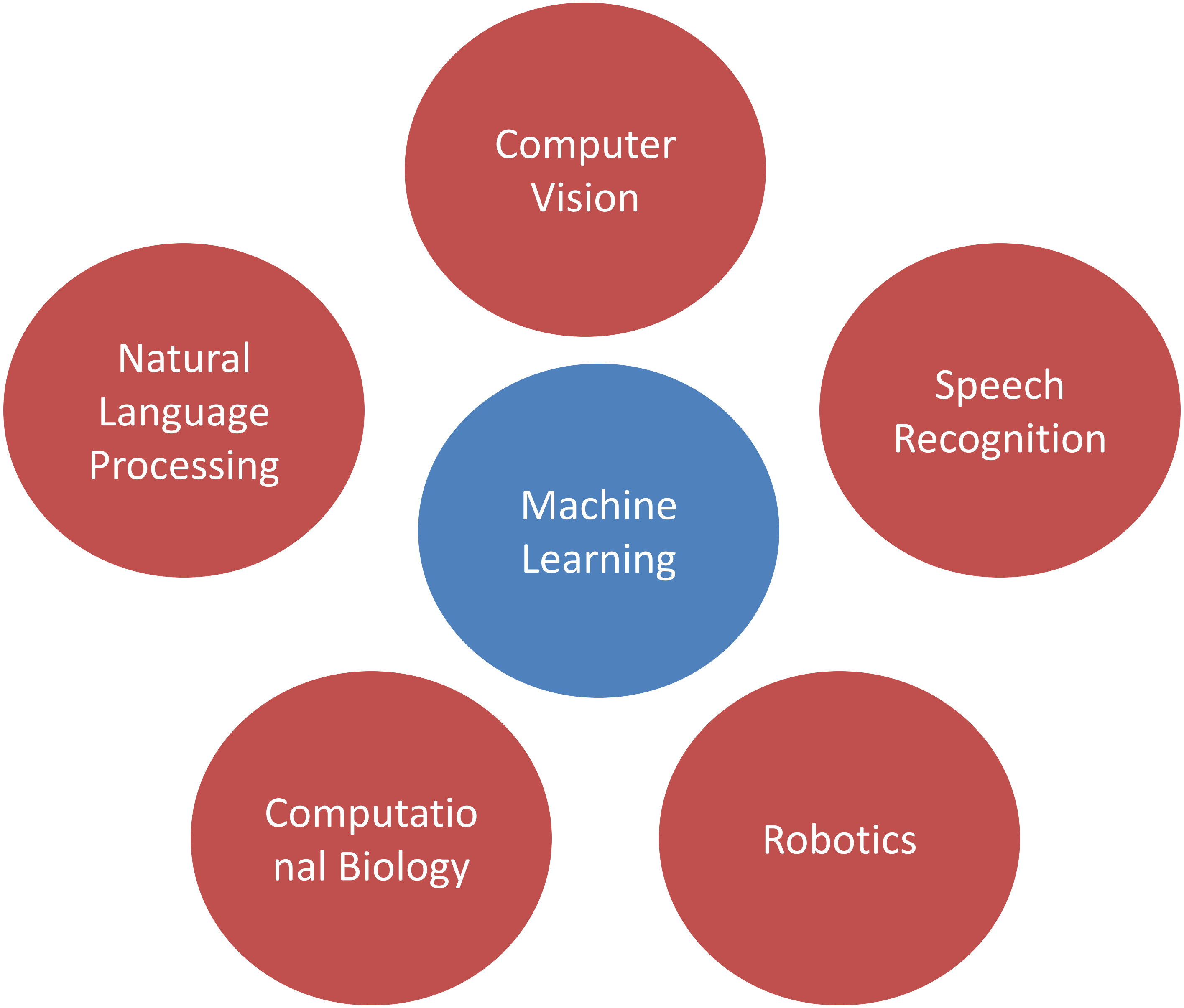
ML application: Robotics

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



- Team Michigan (http://april.eecs.umich.edu/magic/)
  - Robot perception + nagivation + multigent coordination
  - MAGIC competition winner, 2010!! ($750k prize)
  - Prof. Edwin Olson’s group

Machine Learning and other fields



This course is …

- Graduate-level introduction of machine learning
- Provide foundations of machine learning
  - Mathematical derivation, Implementation of the algorithms, Applications
- Topics
  - supervised learning
  - unsupervised learning
  - learning theory
  - reinforcement learning
- Additional advanced topics
  - sparsity and feature selection, Bayesian techniques, and deep learning.

This course is …

- 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.
- This will be a fun class!!

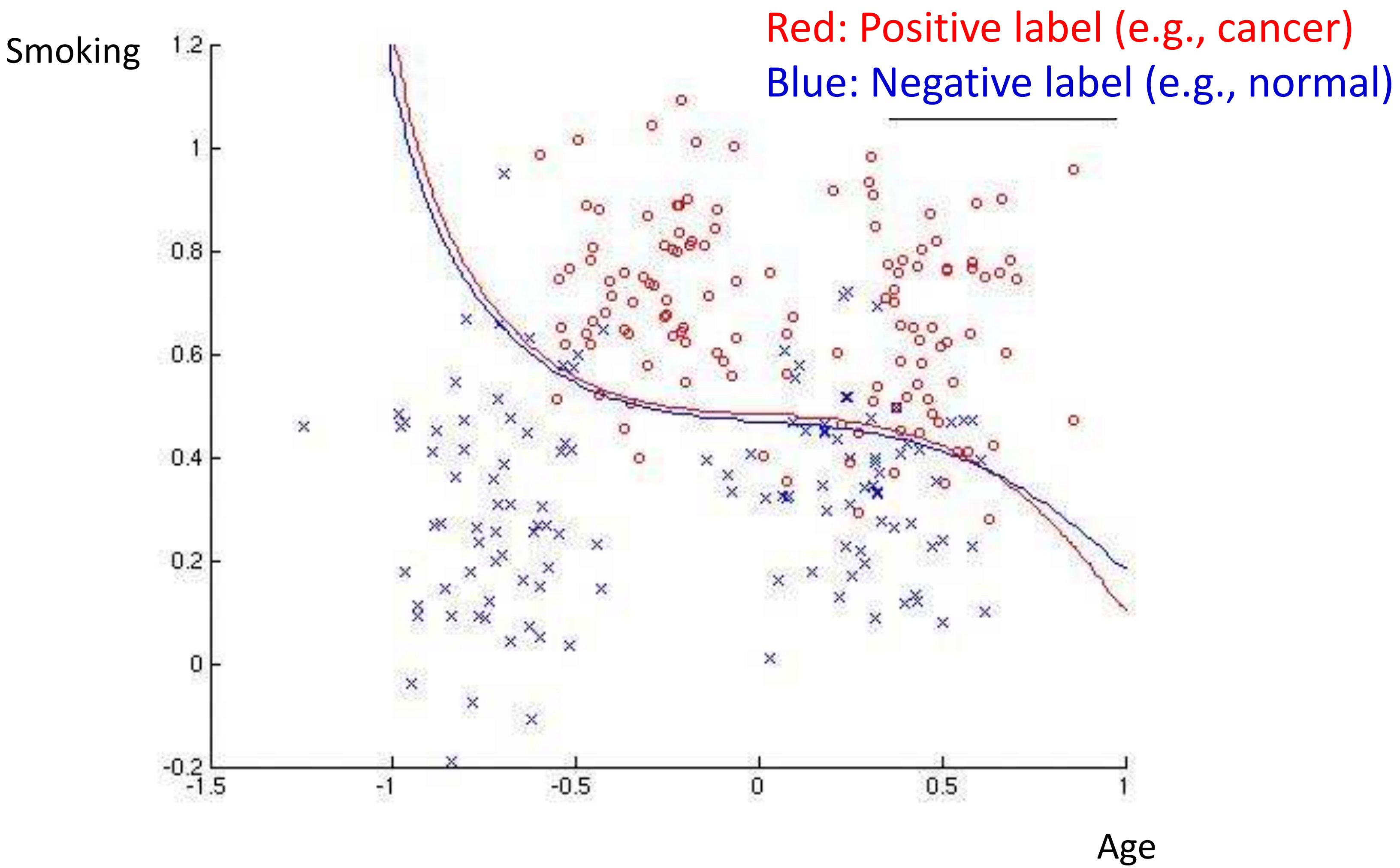
Machine Learning Tasks

- Supervised Learning
  - Classification
  - Regression
- Unsupervised Learning
  - Density estimation
  - Clustering
  - 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 - Classification



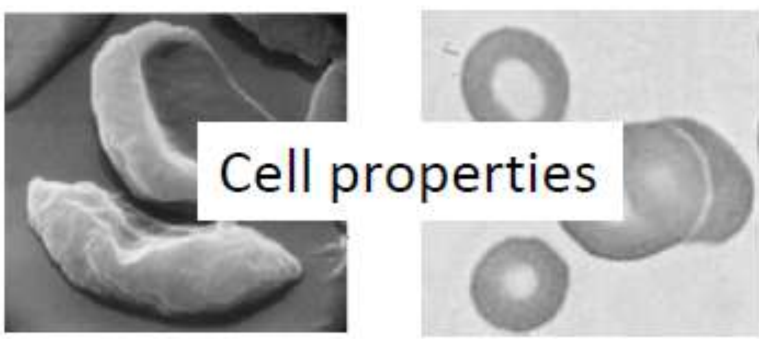
"Learning decision boundaries"

Supervised Learning - Classification

Feature Space $\mathcal{X}$

Label Space $\mathcal{Y}$

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

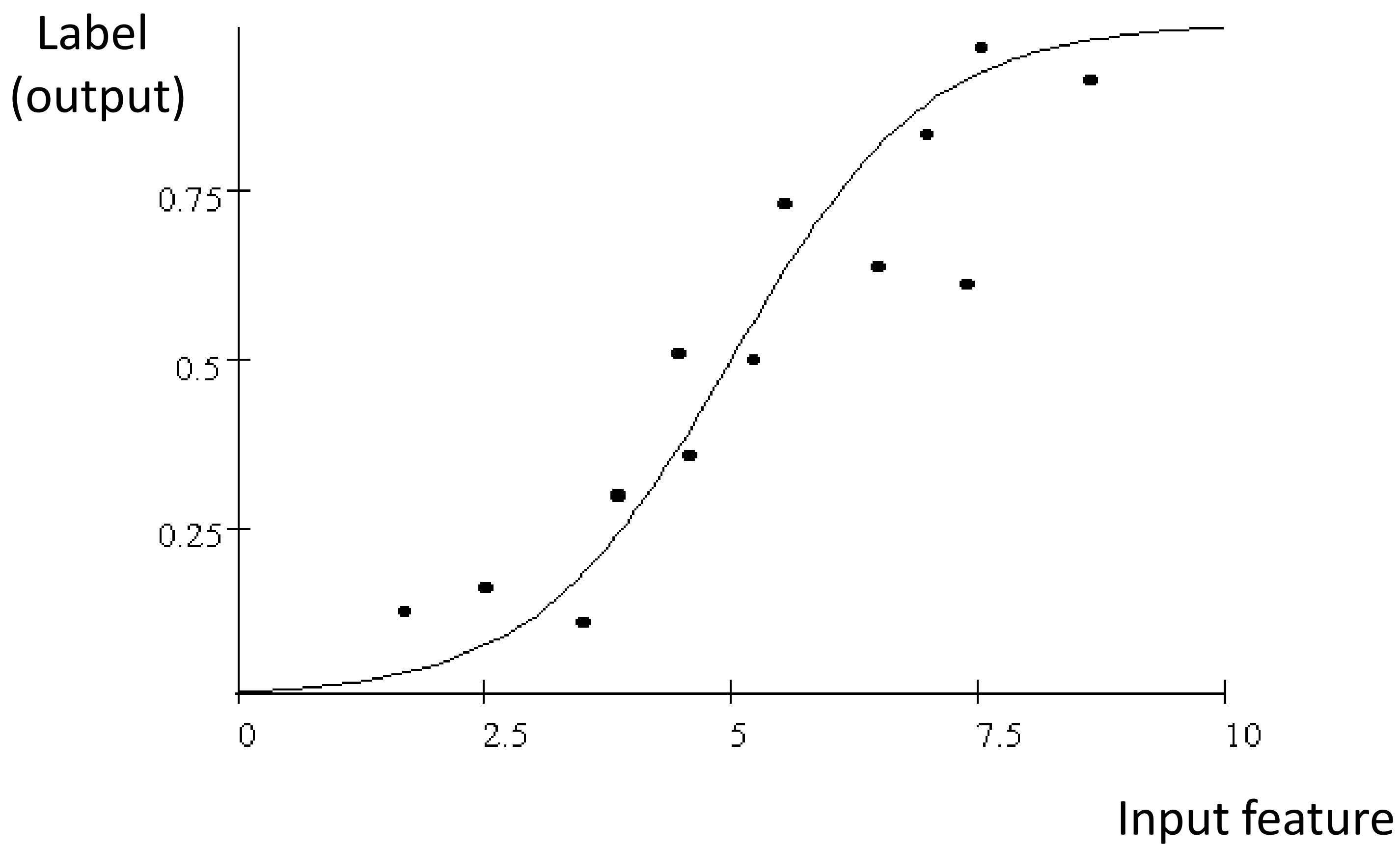


"Anemic cell"
"Healthy cell"

Discrete Labels

Slide credit: Aarti Singh

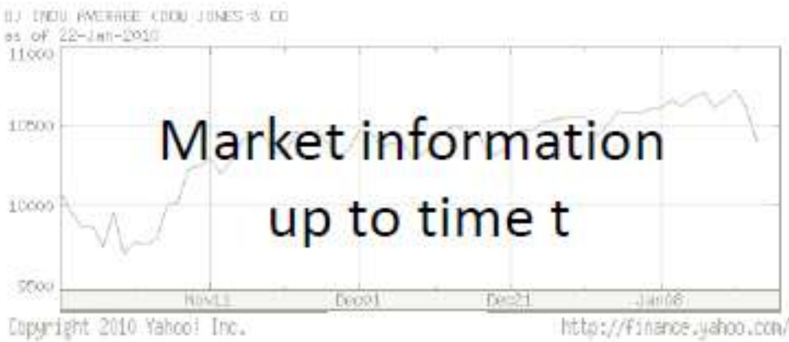
Supervised Learning - Regression



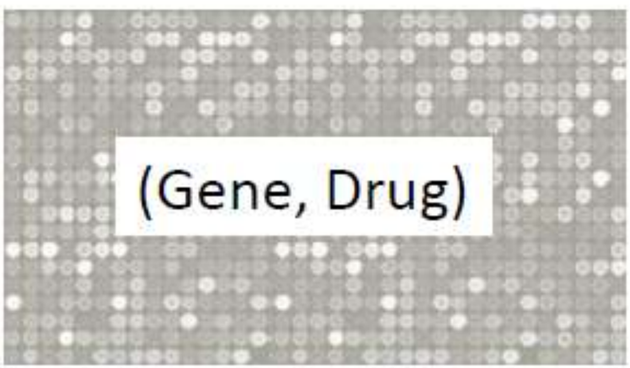
"Learning regression function f(X)"

Supervised Learning - Regression

Feature Space $\mathcal{X}$ | **Label** Space $\mathcal{Y}$



→ Share Price
"$ 24.50"



→ Expression level
"0.01"

Continuous Labels

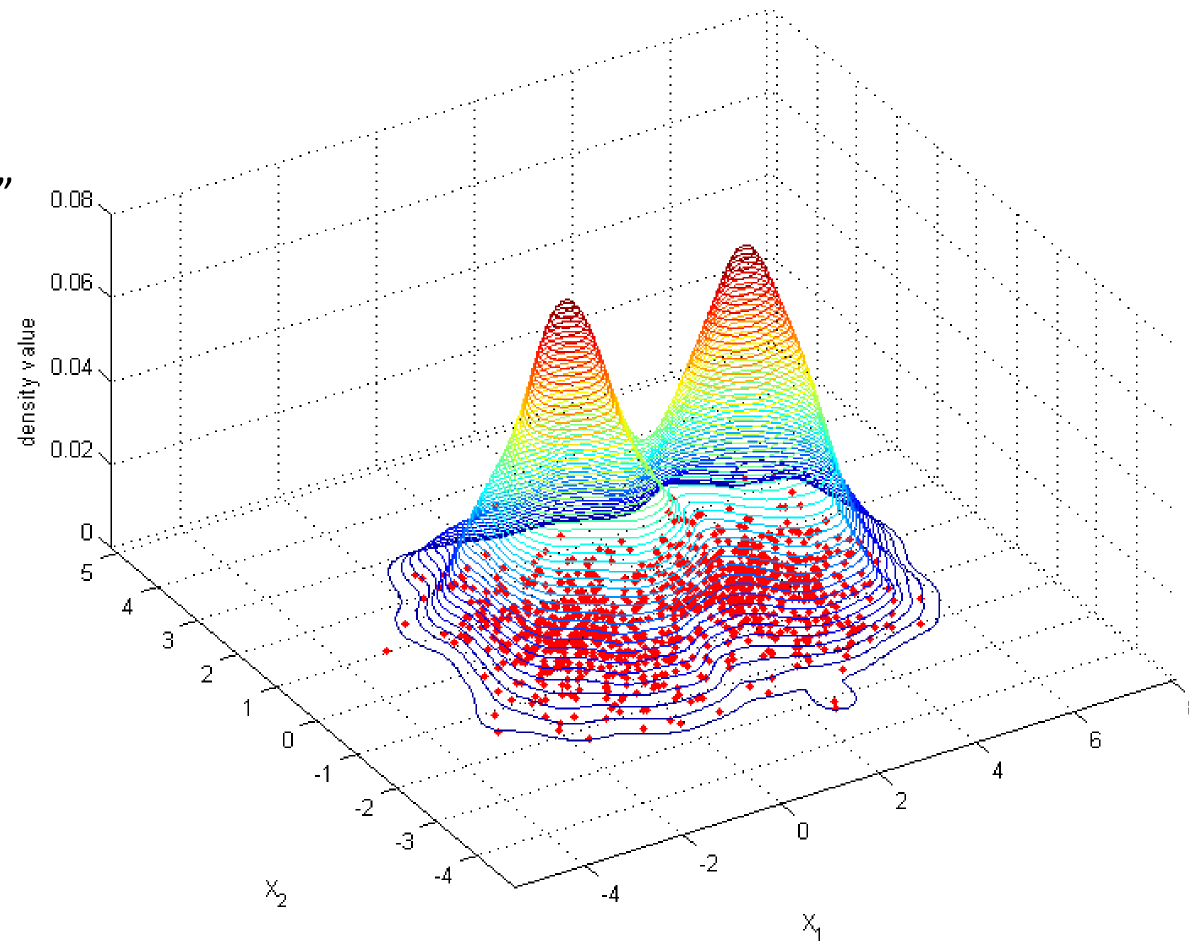
Slide credit: Aarti Singh

Unsupervised Learning

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

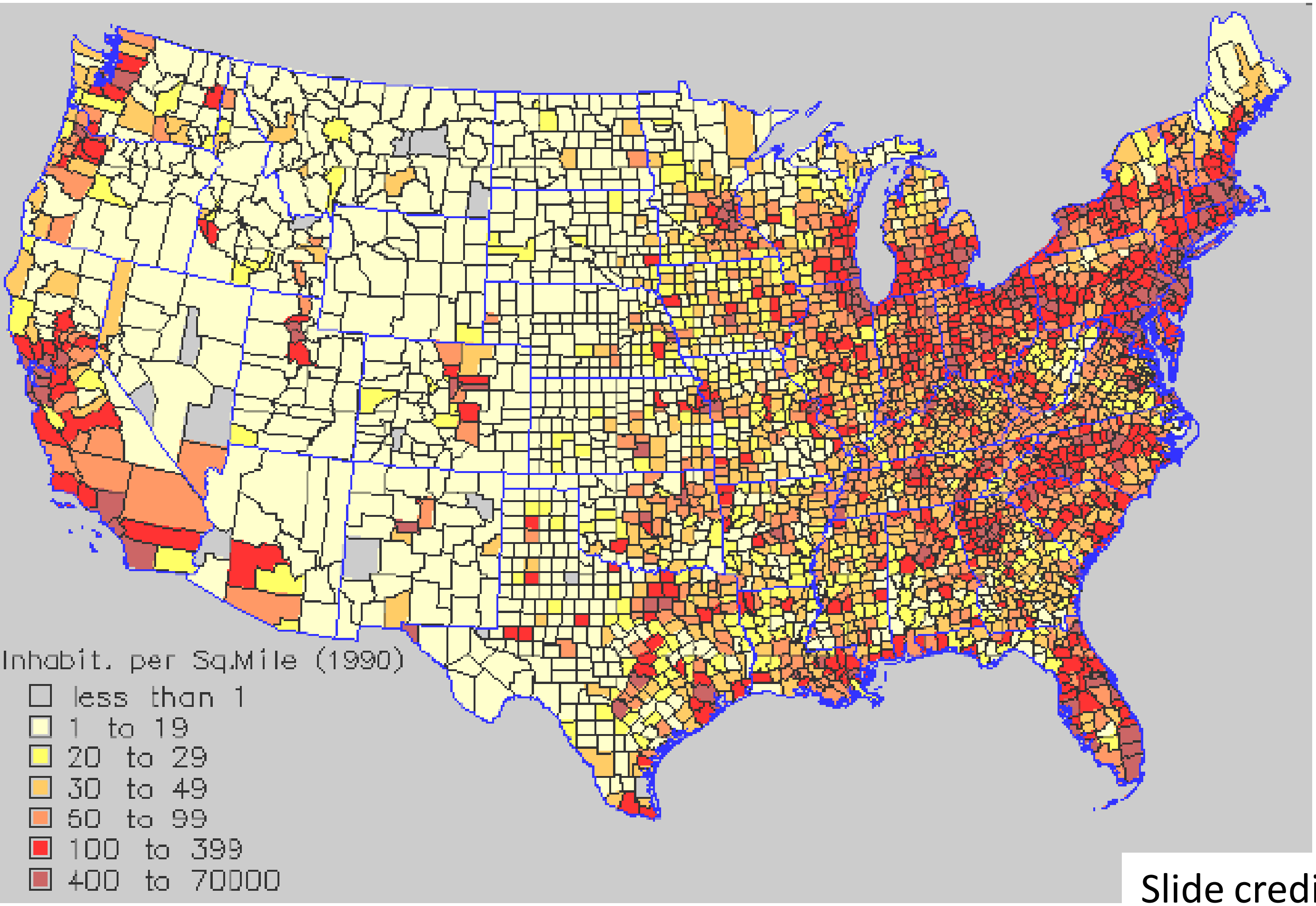
Unsupervised Learning – Density estimation

P(X1, X2)
"Probability"



Unsupervised Learning – Density estimation

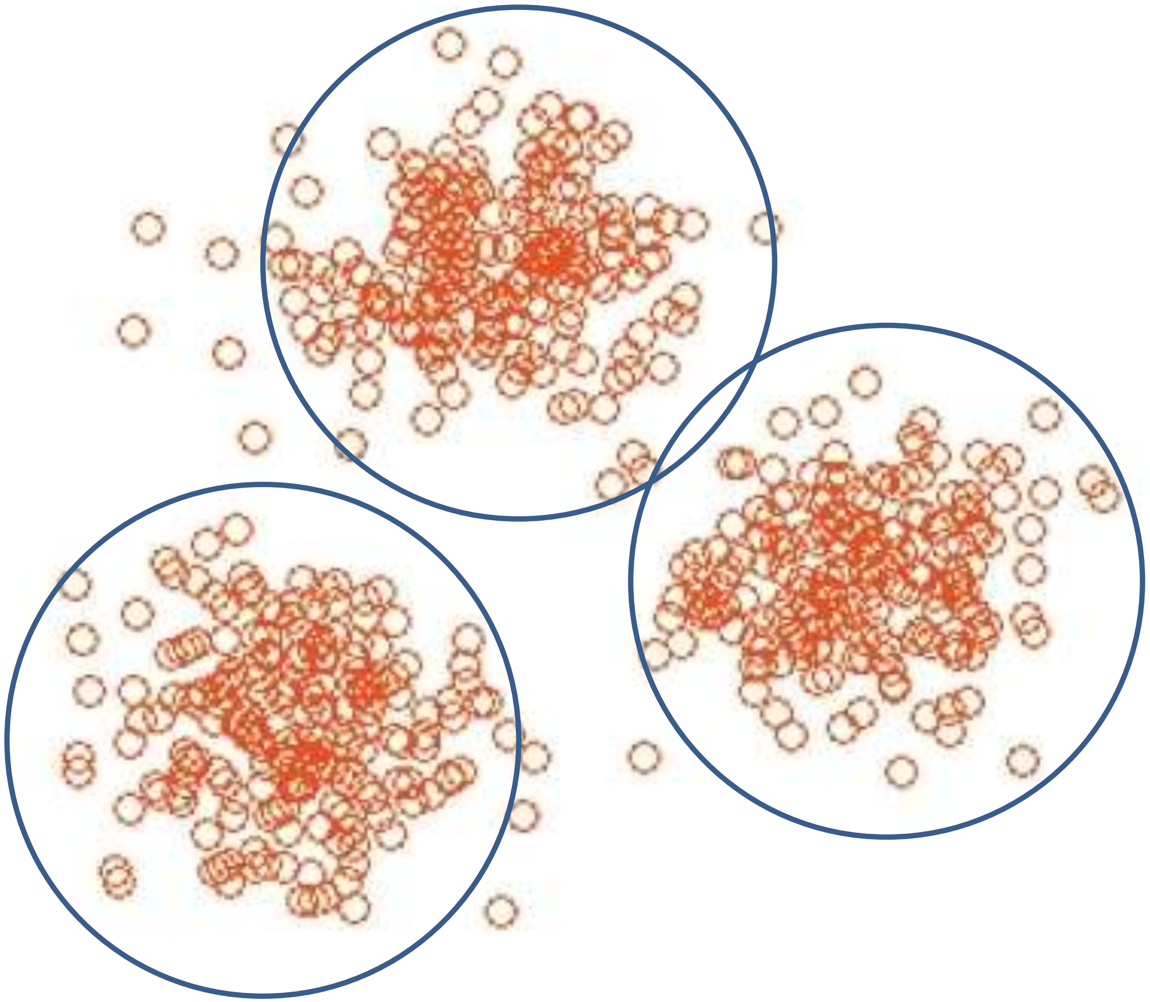
Population density



Slide credit: Aarti Singh

Unsupervised Learning – Clustering

- "Grouping into similar examples"



Unsupervised Learning – Clustering

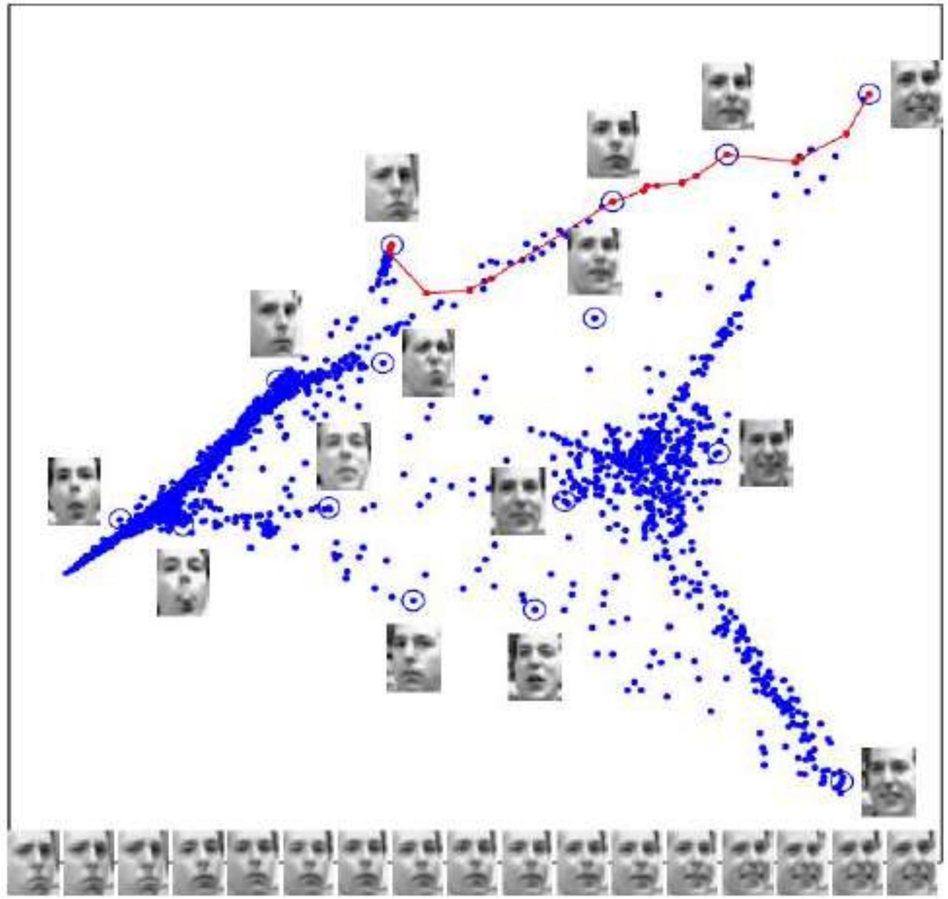
Group similar things e.g. images

[Goldberger et al.]

Slide credit: Aarti Singh

Unsupervised Learning-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: Dynamical environment, 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?

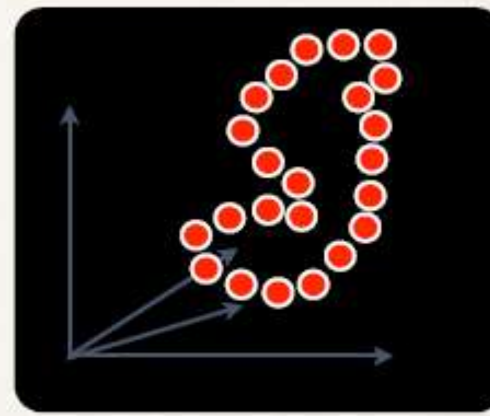
Feature representations

Feature Extraction

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

Real World

Data



Vector Space

$$\{\vec{x}_1, \ldots, \vec{x}_n\} \in \mathcal{R}^d$$

Each dimension is
one feature.

Slide credit: Kilian Weinberger

Learning pipeline

pixel 1

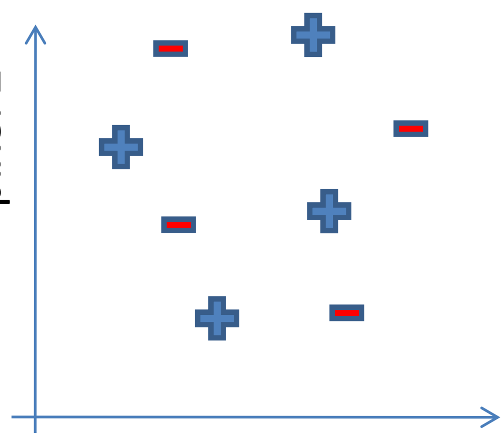
pixel 2

Input

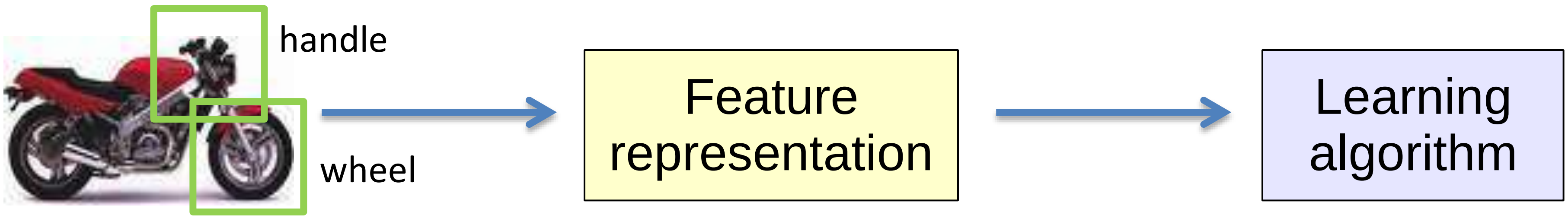
Learning algorithm

Input space

- + Motorbikes
- – “Non”-Motorbikes



Learning pipeline

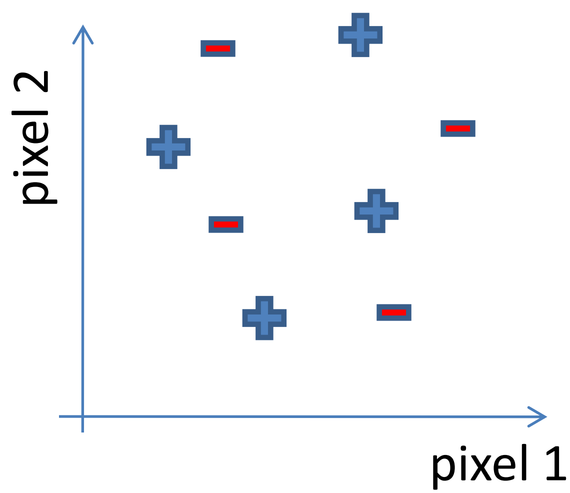


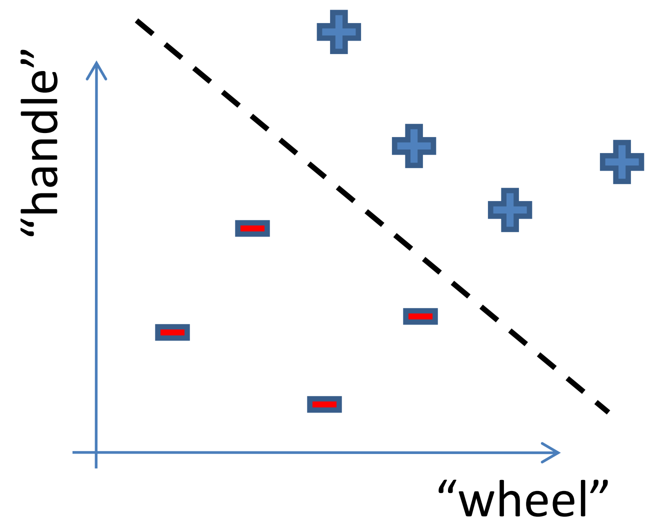
Input

Input space

- Motorbikes
- "Non"-Motorbikes

Feature space





Examples of features: Housing data

- **Given statistics about houses in a local area, predict median value of homes.**
  - 1. CRIM: per capita crime rate by town
  - 2. ZN: proportion of residential land zoned for lots over 25,000 sq.ft.
  - 3. INDUS: proportion of non-retail business acres per town
  - 4. CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
  - 5. NOX: nitric oxides concentration (parts per 10 million)
  - 6. RM: average number of rooms per dwelling
  - 7. AGE: proportion of owner-occupied units built prior to 1940
  - …..
- **Label: MEDV: Median value of owner-occupied homes in $1000's**

How is computer perception done?

State-of-the-art:
"hand-tuning"

Input data → Feature representation → Learning algorithm

Object detection

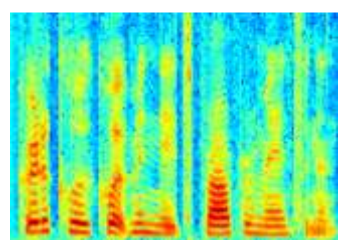
Image

Low-level
vision features

Recognition

Audio classification

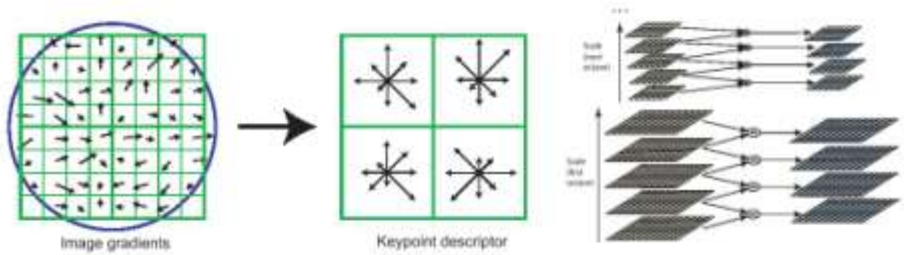
Audio



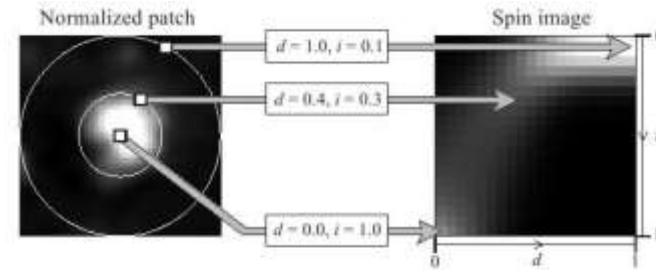
Low-level
audio features

Speaker
identification

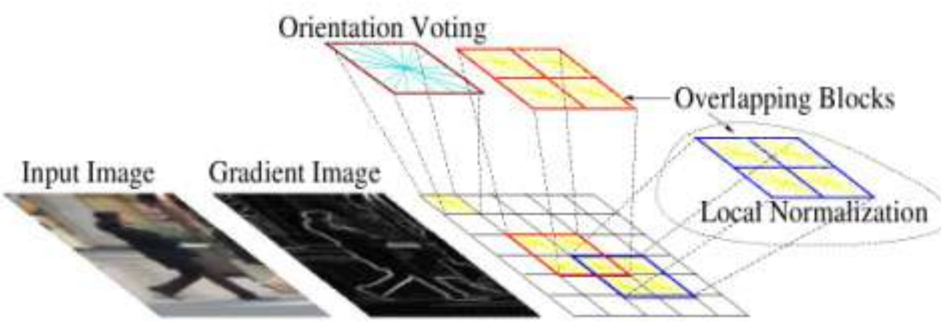
Computer vision features



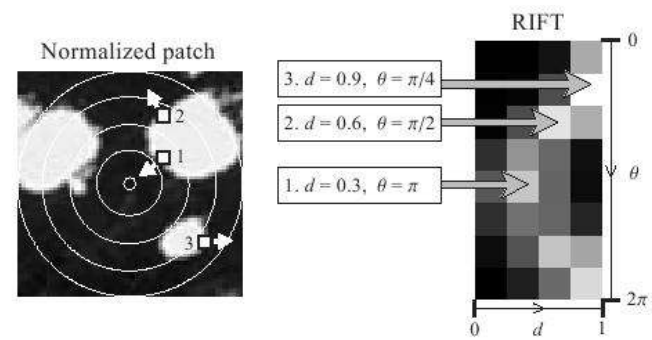
SIFT



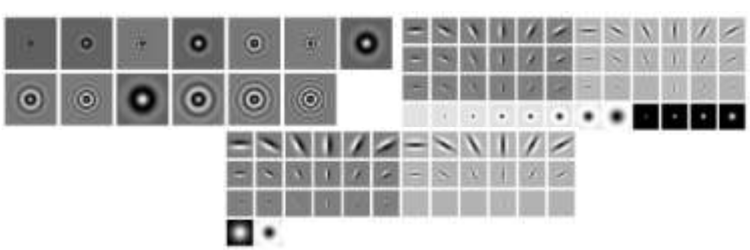
Spin image



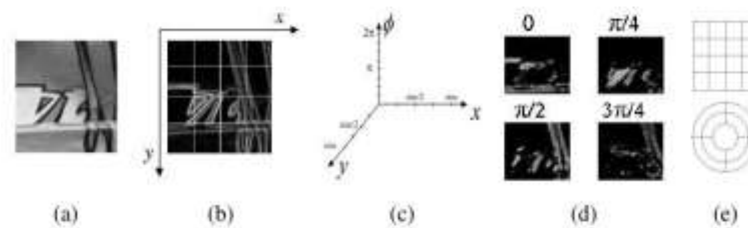
HoG



RIFT

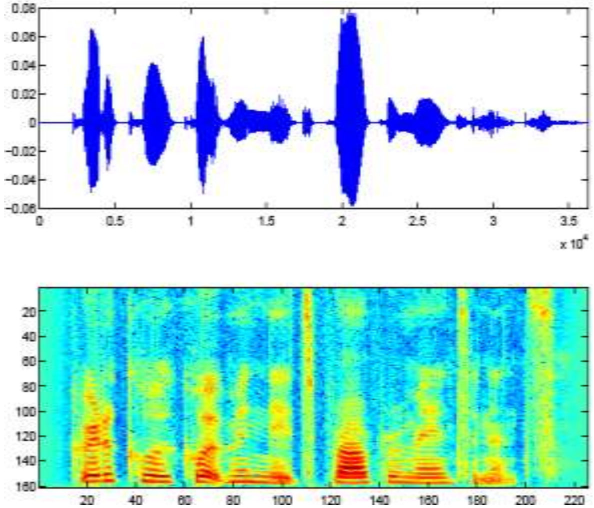


Textons

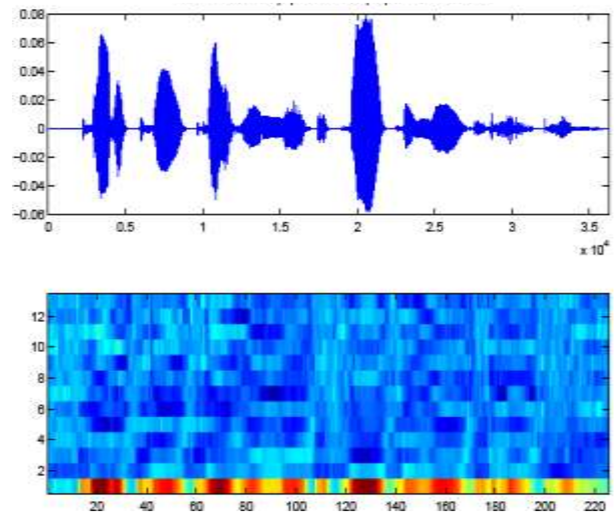


GLOH

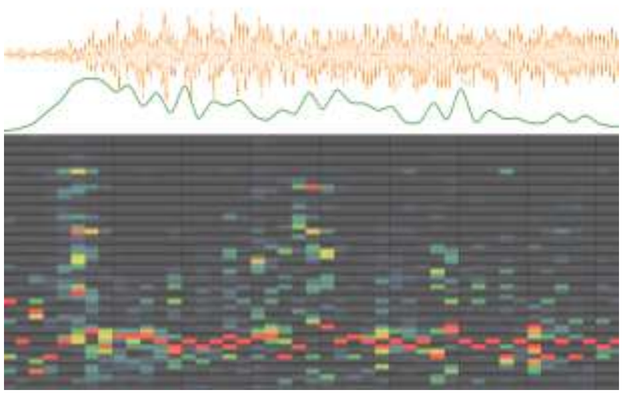
Audio features



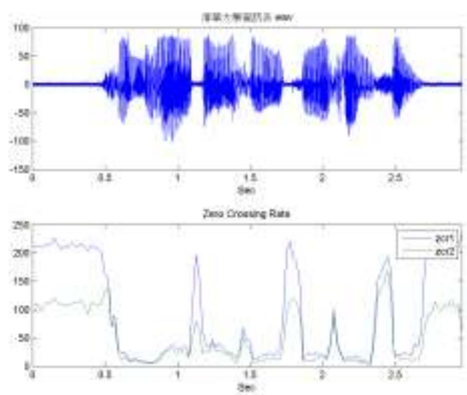
Spectrogram



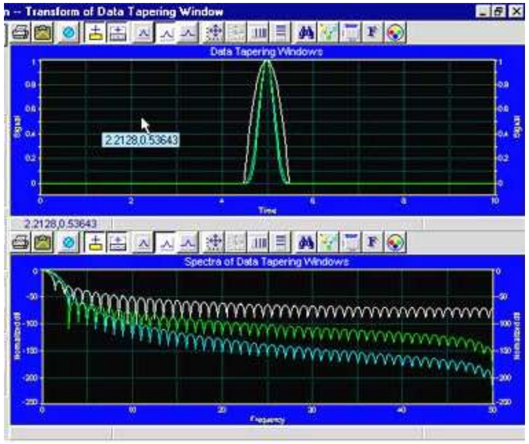
MFCC



Flux



ZCR



Rolloff

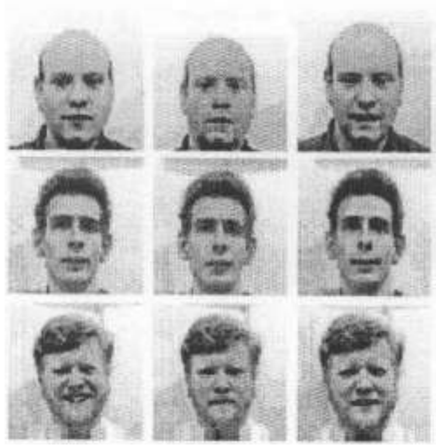
Advanced topic: Learning features!

- Problem of hand-engineered features
  - 1. Needs expert knowledge
  - 2. Requires time-consuming hand tuning
  - 3. Does not generalize to other domains
- Key question: Can we automatically learn good feature representations from input data?

Learning features via subspaces

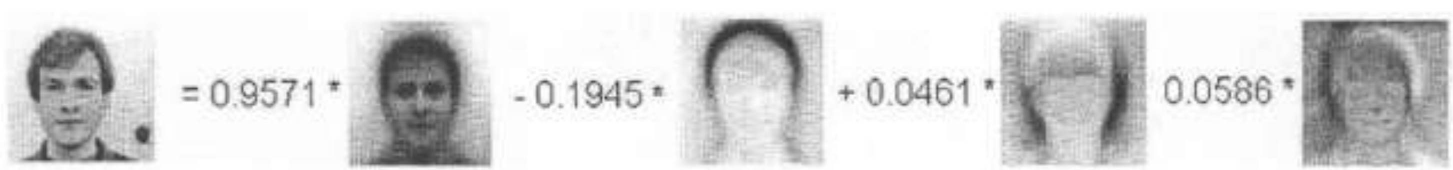
- Example: Eigenfaces

Training face images



Learned PCA bases

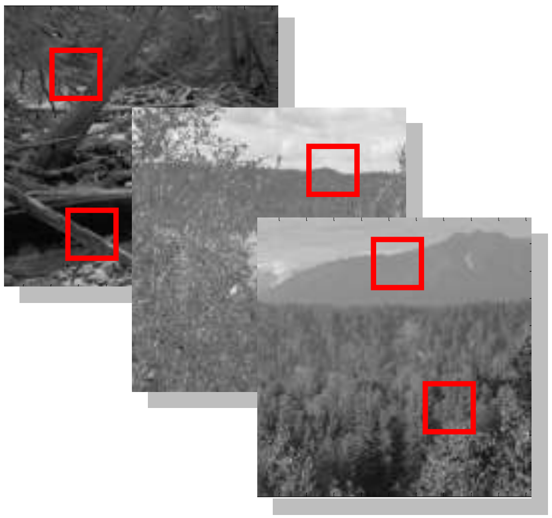
Test example



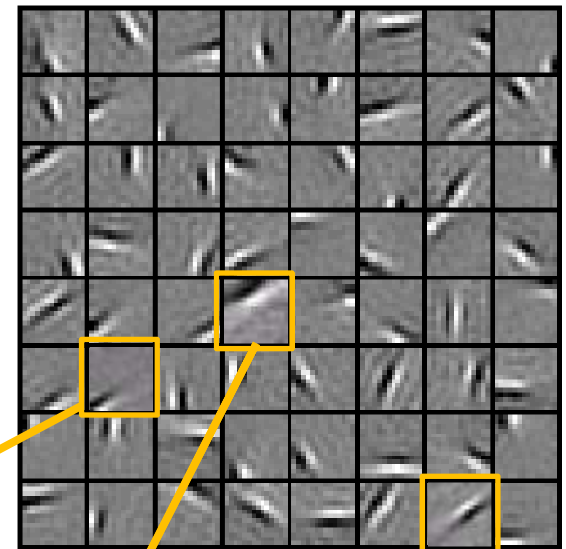
Images from www.cse.unr.edu/~bebis/CS485/Lectures/Eigenfaces.ppt

Learning features via sparse coding

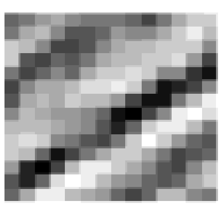
Natural Images



Learned bases: “Edges”

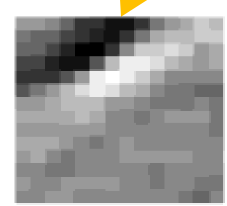


Test example

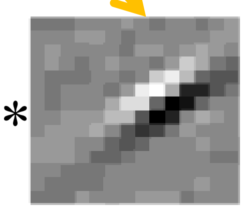


$\sim 0.8 *$

$+ 0.3 *$



$+ 0.5 *$



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

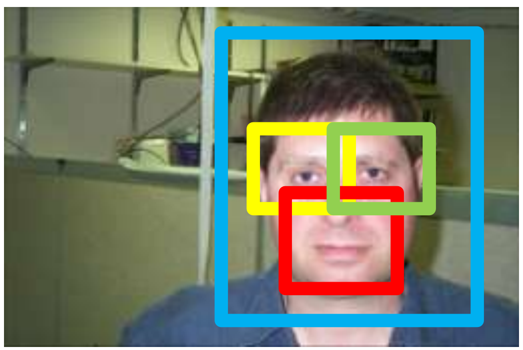
[0, 0, …, 0, **0.8**, 0, …, 0, **0.3**, 0, …, 0, **0.5**, …]
= coefficients (feature representation)

Compact & easily interpretable

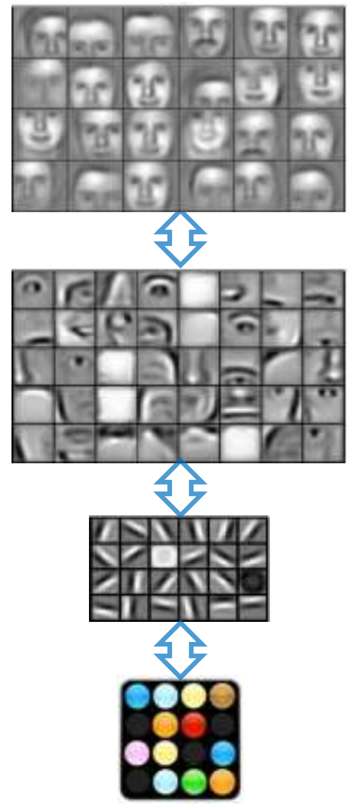
Learning hierarchy of features

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

Input data



Feature representation



3rd layer
“Objects”

2nd layer
“Object parts”

1st layer
“Edges”

Pixels

2. good performance in prediction.

Next class

- Supervised Learning
  - Linear regression

Reminder

- Check syllabus at
  - http://www.eecs.umich.edu/~honglak/teaching/eecs545
- Please fill out the online survey by 5pm today.
  - Required for enrollment!
  - https://spreadsheets.google.com/viewform?hl=en&formkey=dHpmUXpsSXpvWEsxcWp2WGoyclNkMWc6MQ#gid=0
- For all questions, please send email to eecs545qa@umich.edu (not to individual staff)

Questions?