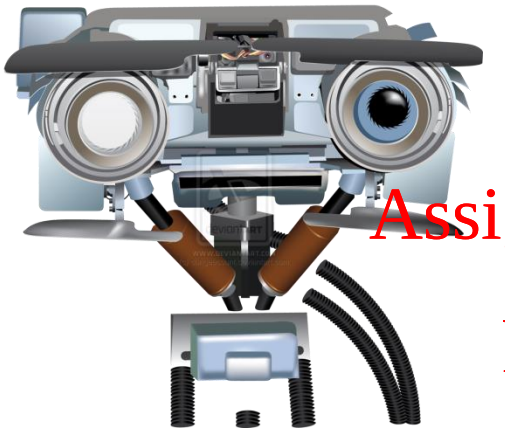


Machine Learning Introduction

Bryan Pardiner

Slides adapted from Kate Saenko

about me

8 year gap

A.S., MCC

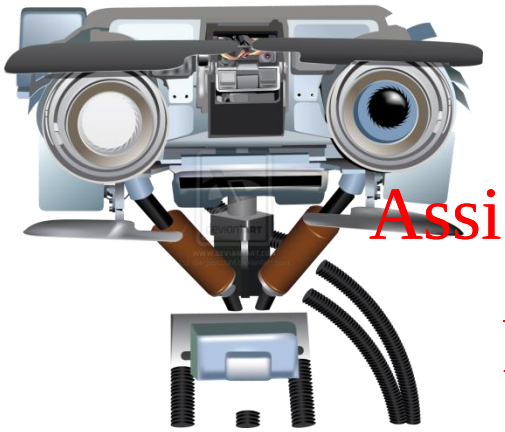
B.S. & PhD, UIUC

At BU 2018-
Tenure Track 2020-

- Research: Artificial Intelligence
  - Deep Learning for Vision
    - Vision and language understanding
    - Representation learning, Explainable AI, Efficient Neural Networks

Today

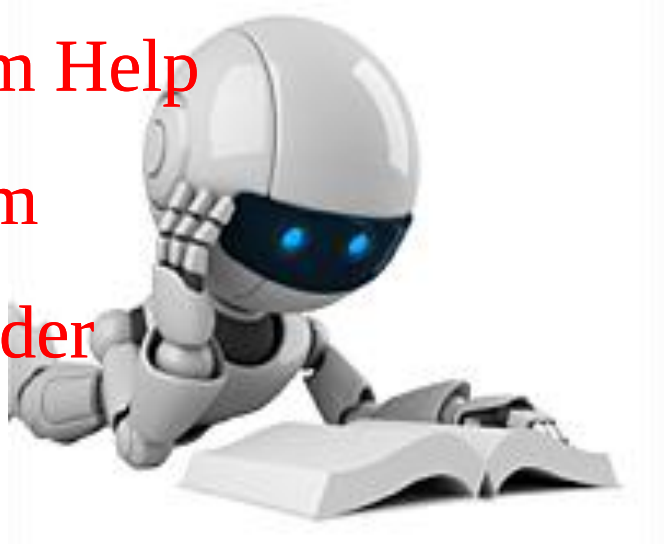
- What is machine learning?
- Supervised learning intro
- Course logistics



Why Do We Need Machine Learning?

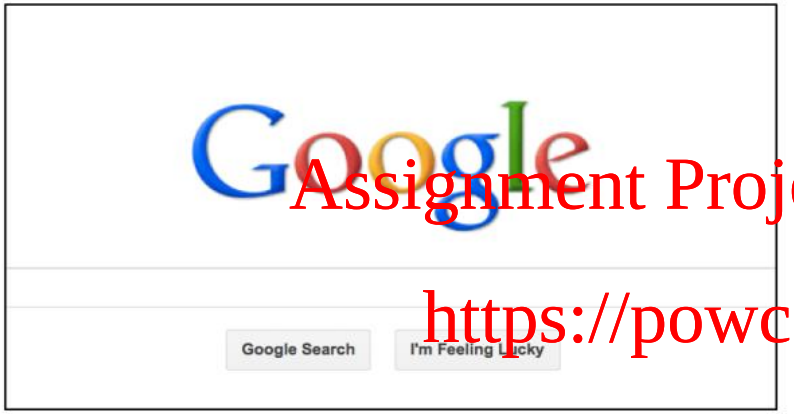
Machine Learning:
Why do we need it?

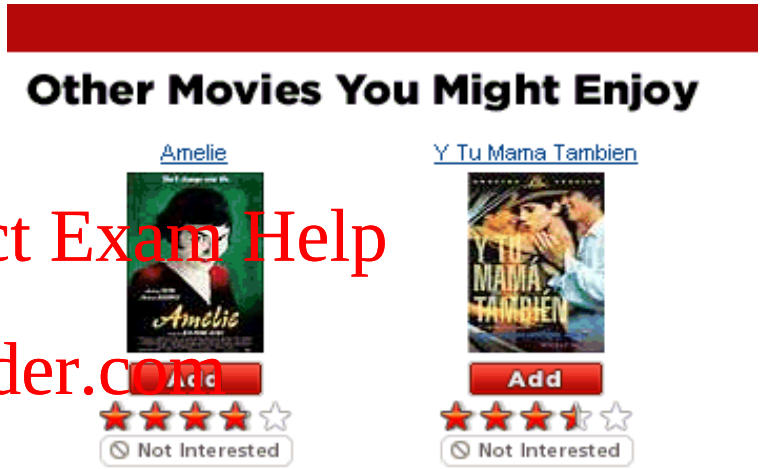
- Help automate boring, hard tasks
- Hard to program computer directly to do the task
- Instead, program a computer to **learn** from examples
- Often use “big data” examples



Machine Learning:

used in lots of ways in our everyday life!





ML wins Jeopardy!

Machine Learning in Real Life:
Smart Cars

- Stanford/Google one of the first to develop self-driving cars
- Cars “see” using many sensors: radar, laser, cameras

Machine Learning in Real Life:

Medical and Scientific Data

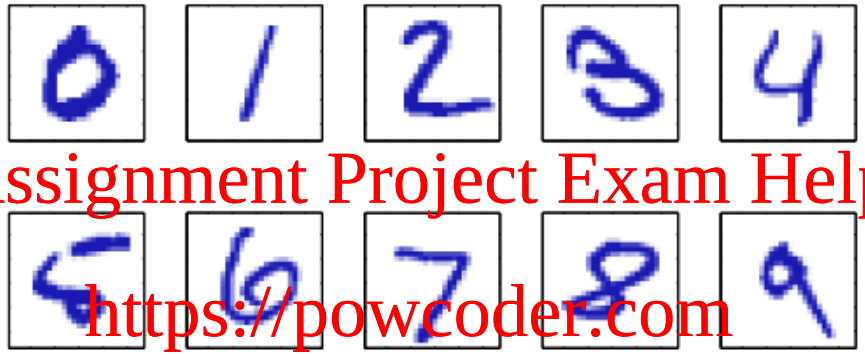
Machine Learning in Real Life:
Robotics

Vierreck, Ten Pas, Saenko, Platt. Learning a Visuomotor Controller for Real World Robotic Grasping… CORL17

Machine Learning in Real Life:

Image Classification

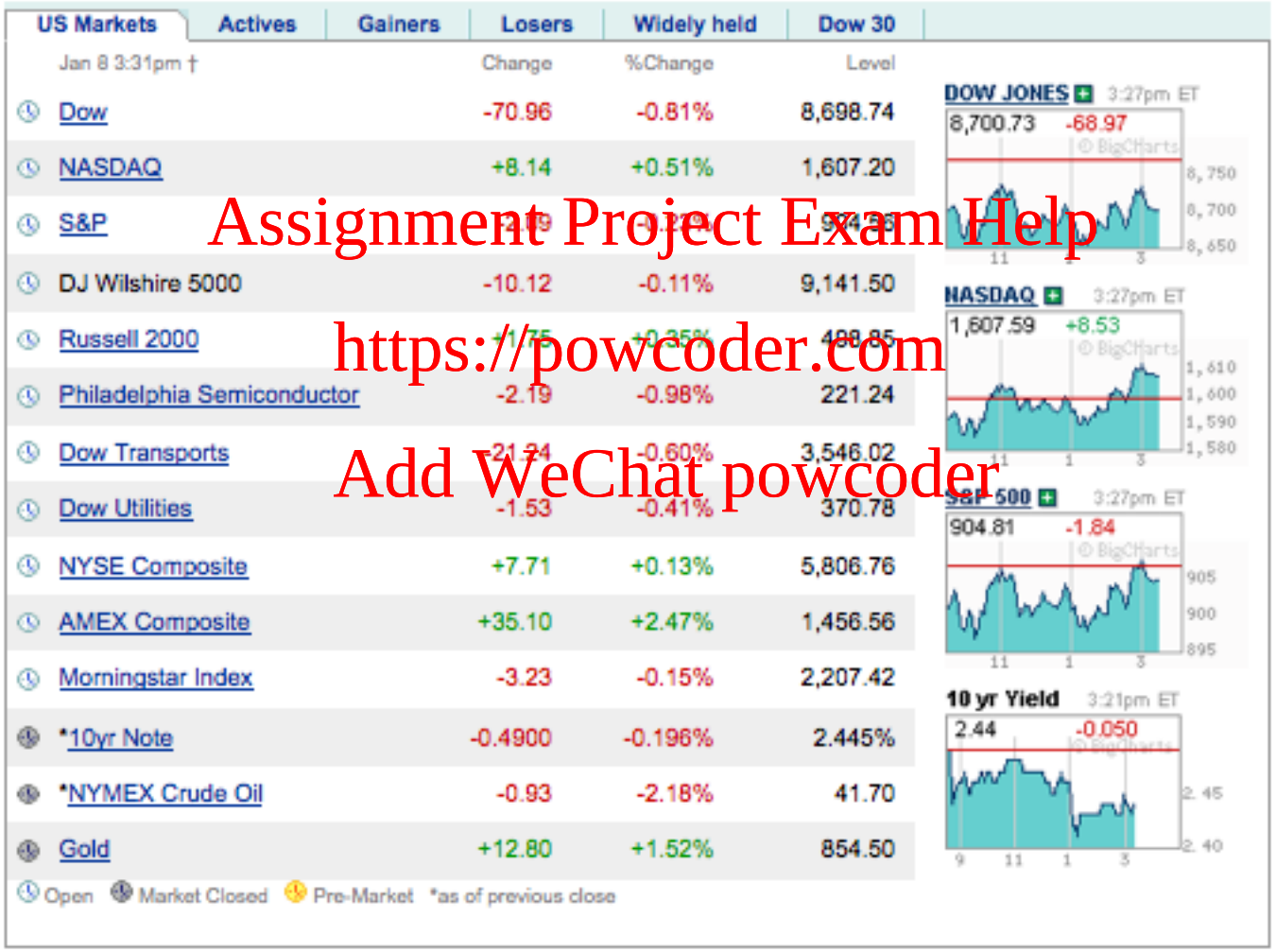
handwritten digits



face tagging on social media

Machine Learning in Real Life:

Computational Finance



US Markets | Actives | Gainers | Losers | Widely held | Dow 30

Jan 8 3:31pm †

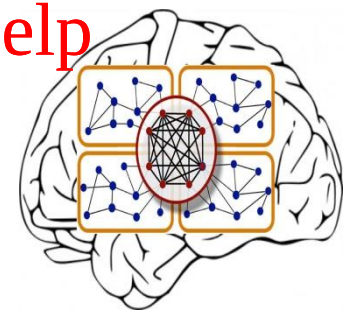
	Change	%Change	Level
Dow	-70.96	-0.81%	8,698.74
NASDAQ	+8.14	+0.51%	1,607.20
S&P	-1.84	-0.20%	904.81
DJ Wilshire 5000	-10.12	-0.11%	9,141.50
Russell 2000	-1.75	-0.35%	488.85
Philadelphia Semiconductor	-2.19	-0.98%	221.24
Dow Transports	-21.24	-0.60%	3,546.02
Dow Utilities	-1.53	-0.41%	370.78
NYSE Composite	+7.71	+0.13%	5,806.76
AMEX Composite	+35.10	+2.47%	1,456.56
Morningstar Index	-3.23	-0.15%	2,207.42
*10yr Note	-0.4900	-0.196%	2.445%
*NYMEX Crude Oil	-0.93	-2.18%	41.70
Gold	+12.80	+1.52%	854.50

Open | Market Closed | Pre-Market | *as of previous close

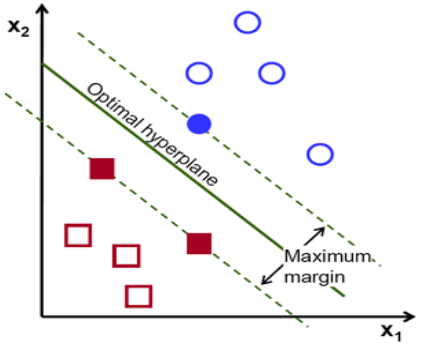
Figure from J.Corso

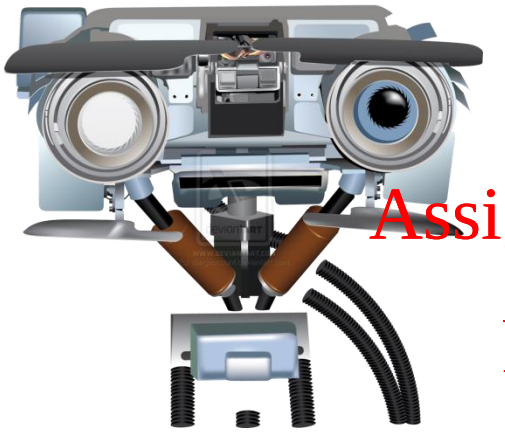
Machine Learning from Big Data

Artificial Neural Network



Support Vector Machine





Introduction: What is Machine Learning?

Machine Learning

- Branch of Artificial Intelligence
- *"creating machine algorithms that can learn from data"*
- Closely related to
  - Pattern recognition
  - Data Mining
  - Big Data
  - Deep learning

Types of learning

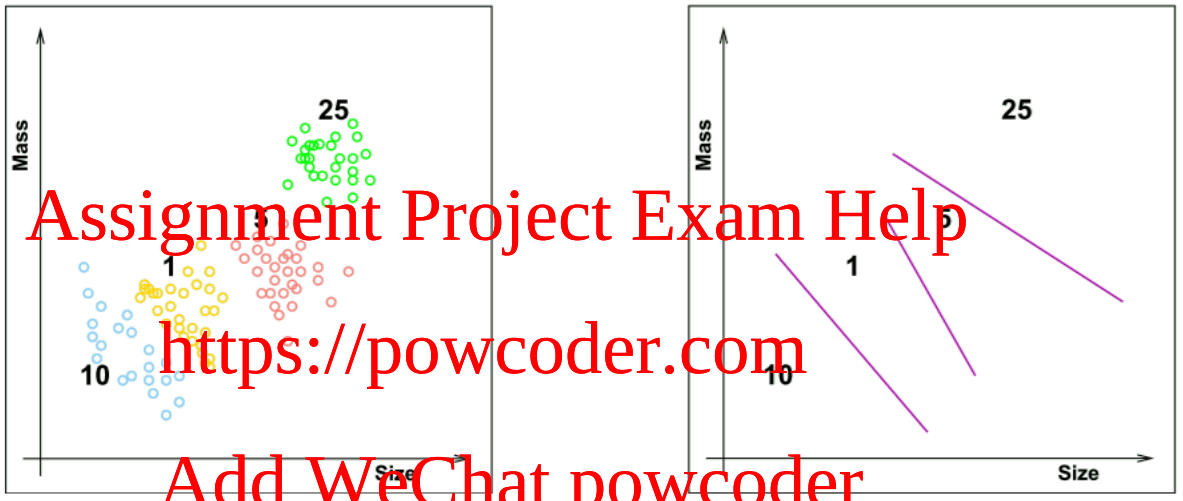
- Supervised
- Unsupervised
- Reinforcement

Supervised Learning

- Given a training set consisting of inputs and outputs, learn to map novel inputs to outputs
- The novel inputs are called a test set
- Outputs can be
  - Categorical (classification)
  - Continuous (regression)

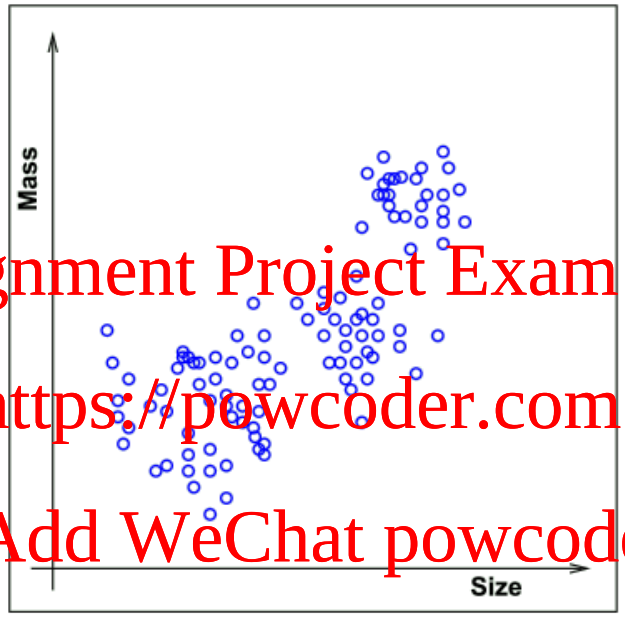
Example of Supervised Learning

recognize coins



- Given training set consisting of coin denomination (penny, nickel, dime, quarter), mass and size
- Learn to predict denomination
- What is input? Output?

Unsupervised Learning

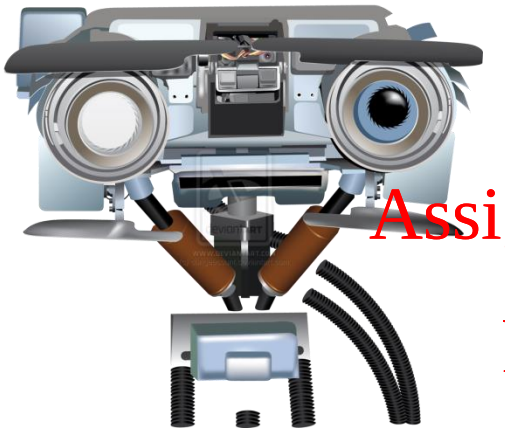


- Given training set consisting of ~~coin denomination (penny, nickel, dime, quarter)~~ mass and size
- Learn… something?

Reinforcement Learning

learn to pick up coins

- Given only input, but can take action
- Predict output (action), get a reward for it

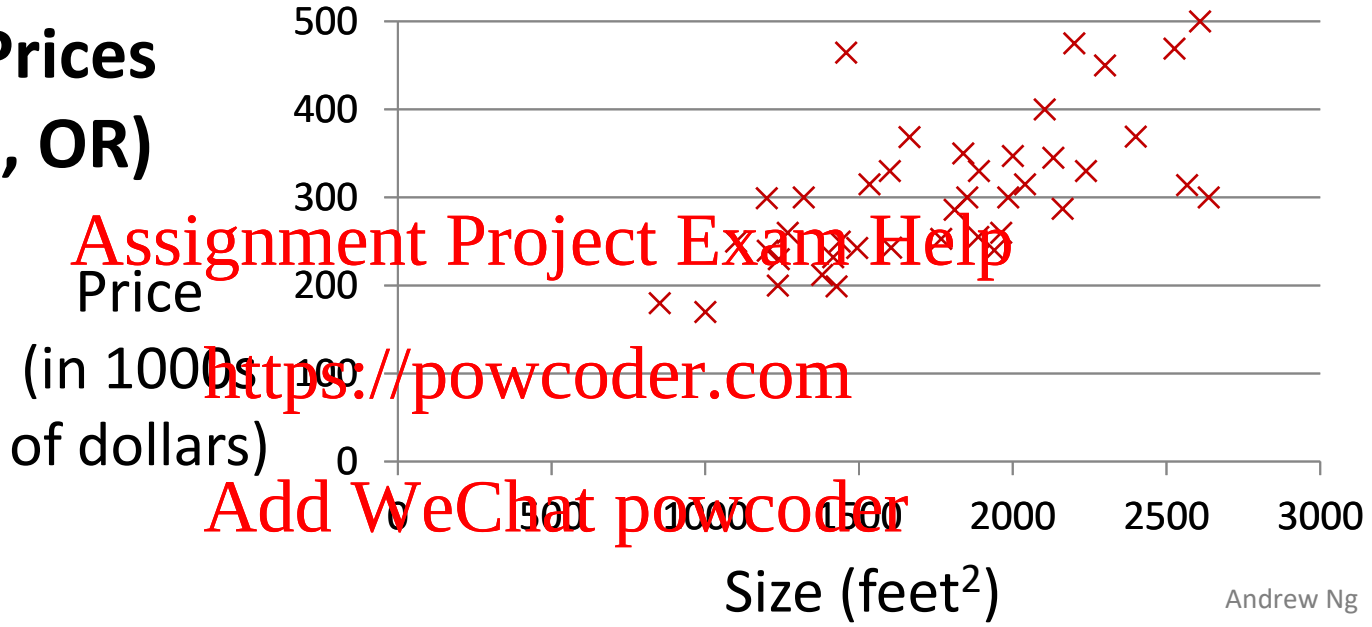


Supervised Learning

Cost functions

Example: house price prediction

Housing Prices (Portland, OR)

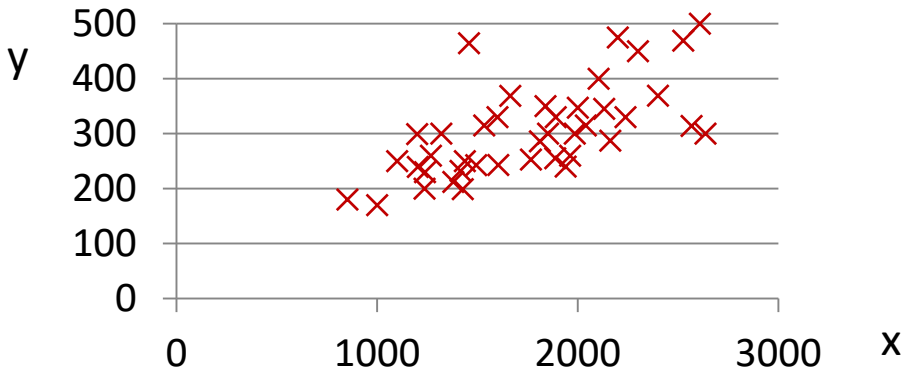


Andrew Ng

Supervised Learning

What should the learner be??

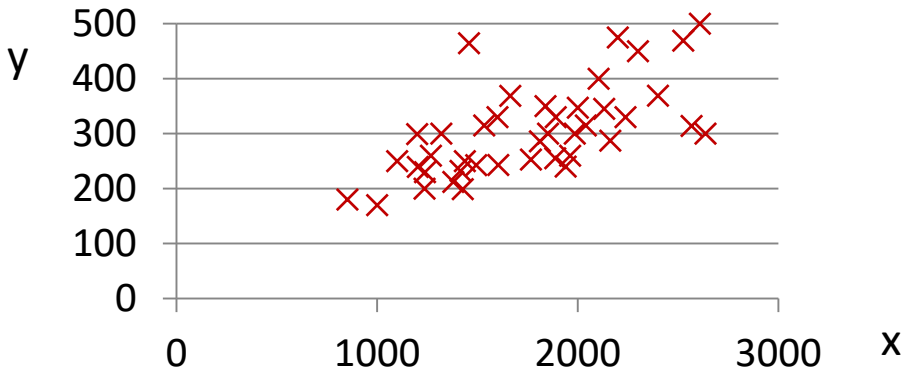
Want: input x → ? → output y



Hypothesis *h*

h : a function parametrized by ϑ

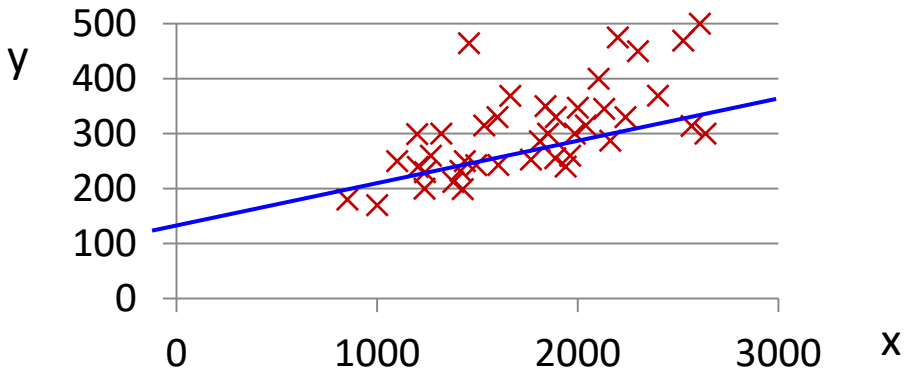
Want: input x → h_θ → output y



How to learn ϑ ?

Given: Training Set $\{x^i, y^i\}$ **But what if $y \neq y^i$??**

Want: input x → h_θ → output y



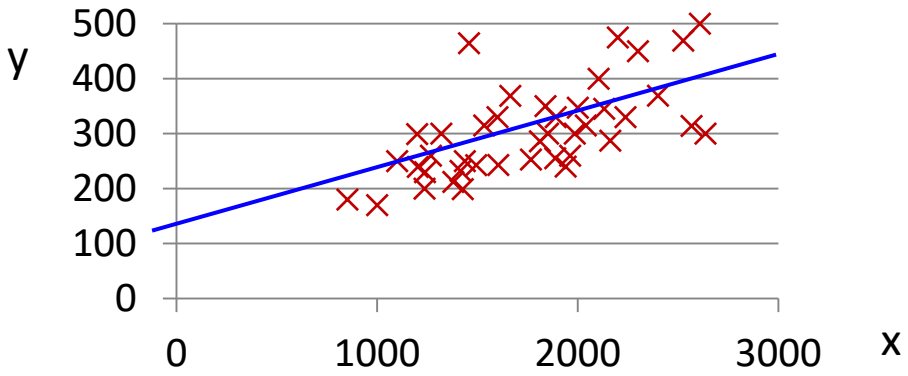
Cost function

Given: Training Set $\{x^i, y^i\}$

Cost function $Cost(y, y^i)$

learning == minimizing cost

Want: input x^i → h_θ → output y



Supervised Learning

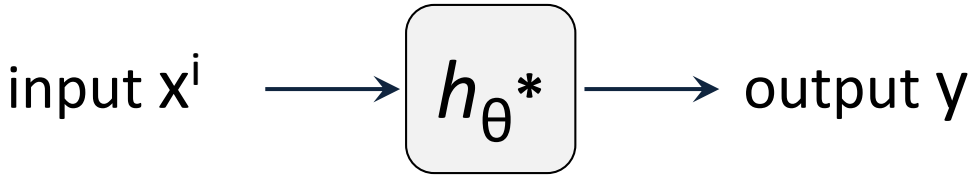
Given: Training Set $\{x^i, y^i\}$

Cost function $\text{Cost}(y, y^i)$

learning == minimizing cost

Learn θ^*: $\min_\theta \text{Cost}(h_\theta(x^i), y^i)$

Want:



Training set

Training set:

Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

Notation:

m = Number of training examples

$x^{(i)}$ = "input" variable / features

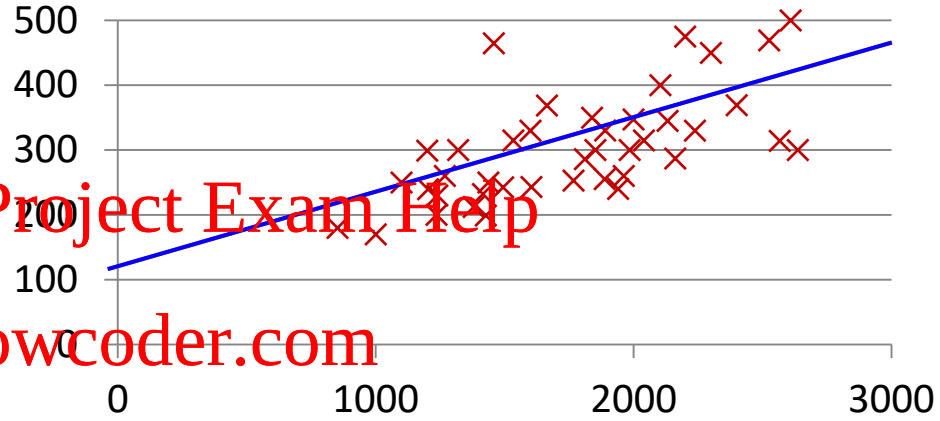
$y^{(i)}$ = "output" variable / "target" variable

What should *h* be?

Linear hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

θ_i's: Parameters



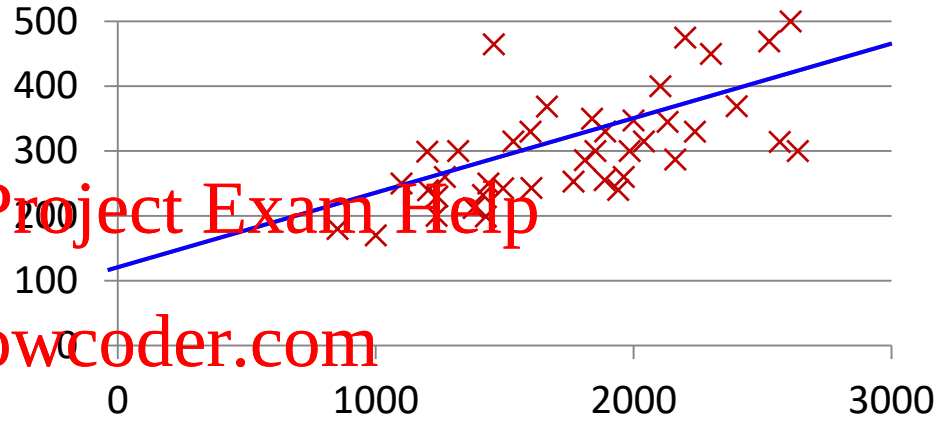
$$\min_{\theta} \text{Cost}(h_\theta, \{x^i, y^i\})$$

What's a good cost function for this problem?

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

θ_i's: Parameters



How about "Sum of squared differences"

Cost Function:

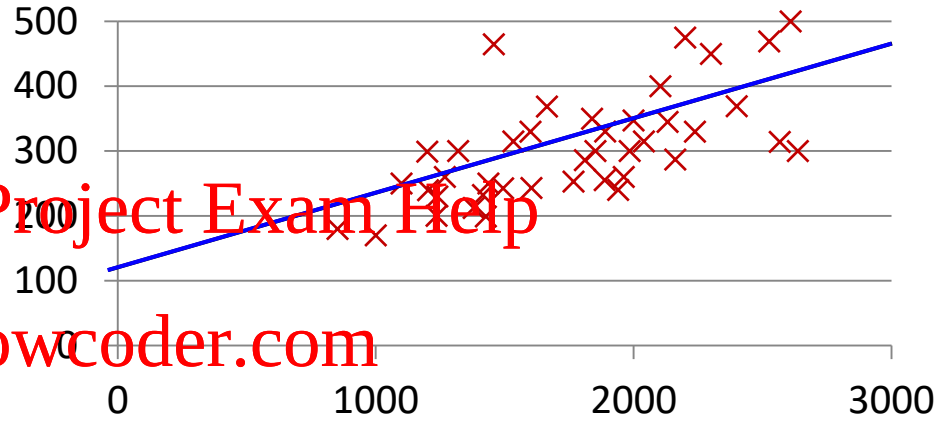
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

2-dimensional θ

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$



θ_i's: Parameters

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

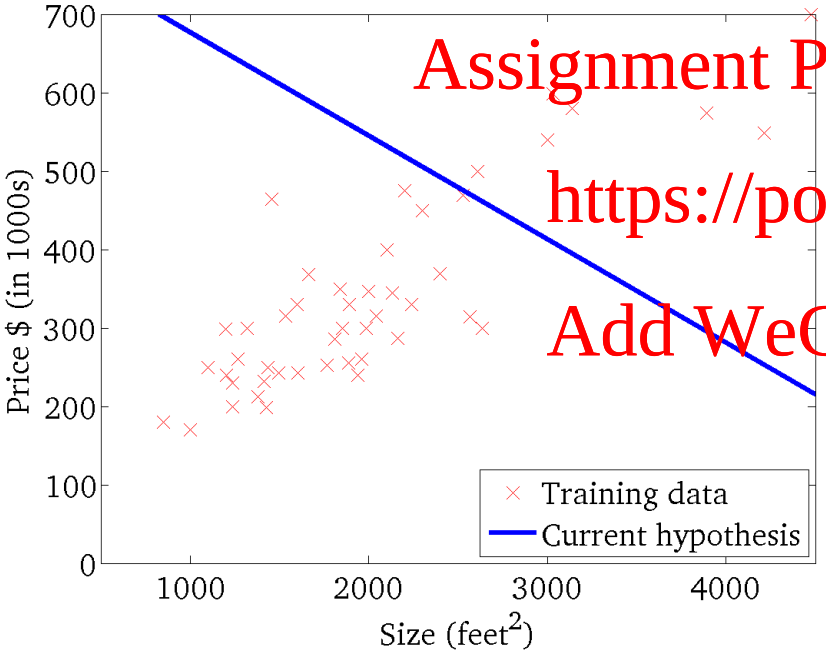
Plotting cost for 2-dimensional θ

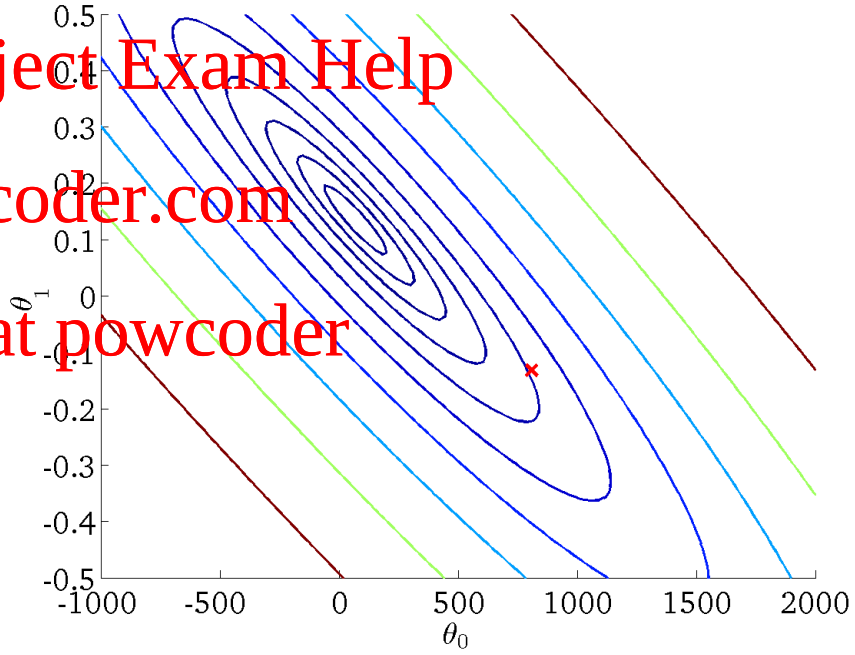
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)





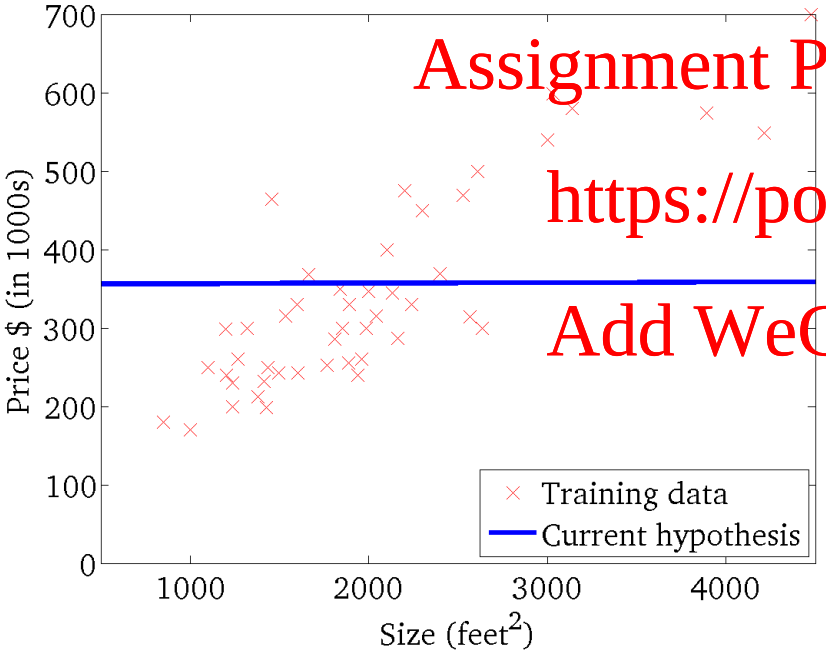
Plotting cost for 2-dimensional θ

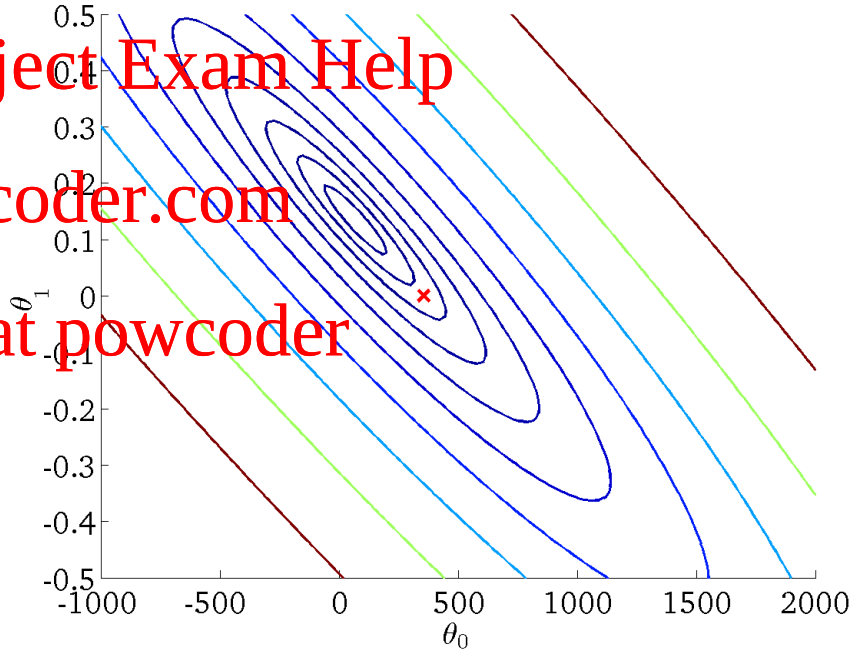
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

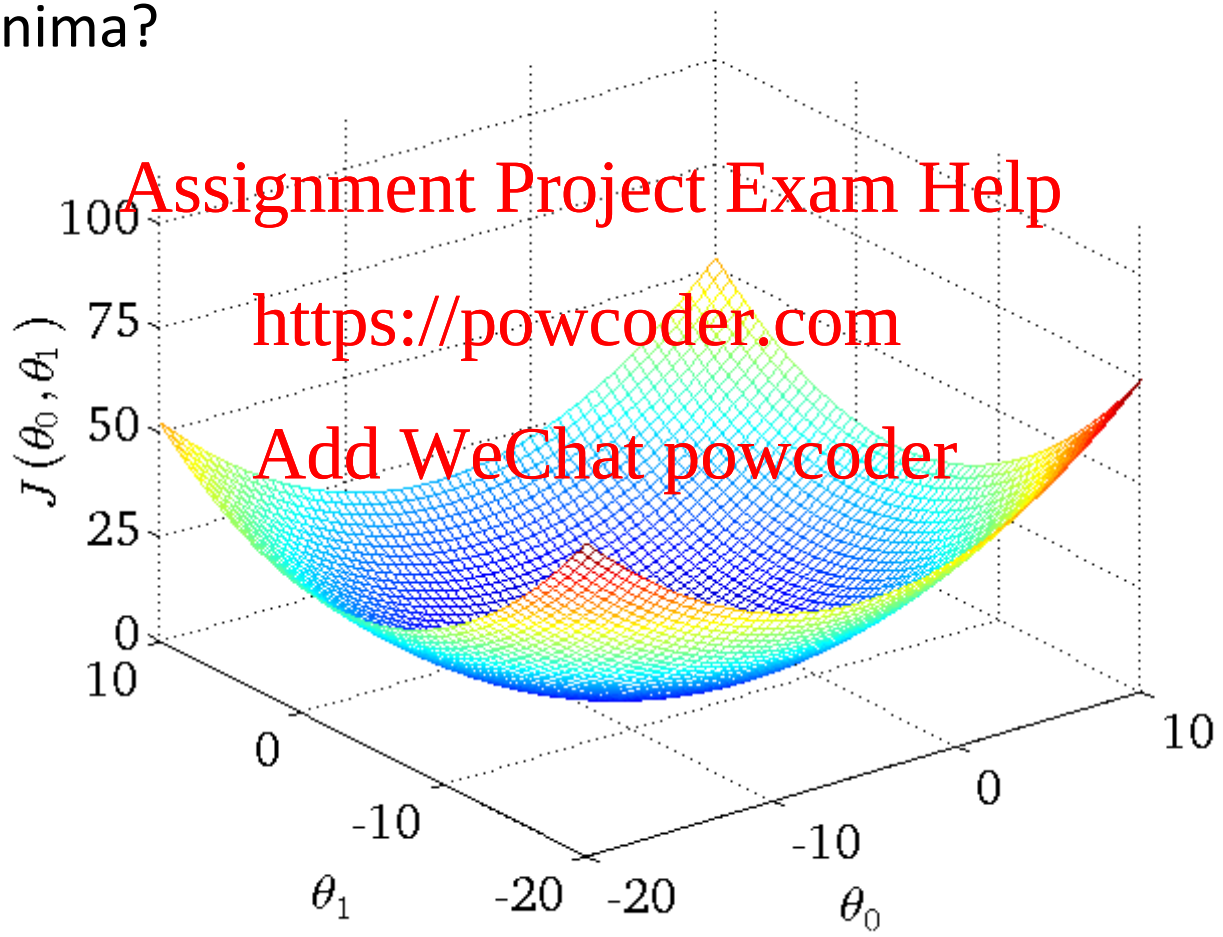




Note, squared loss cost is convex in parameters

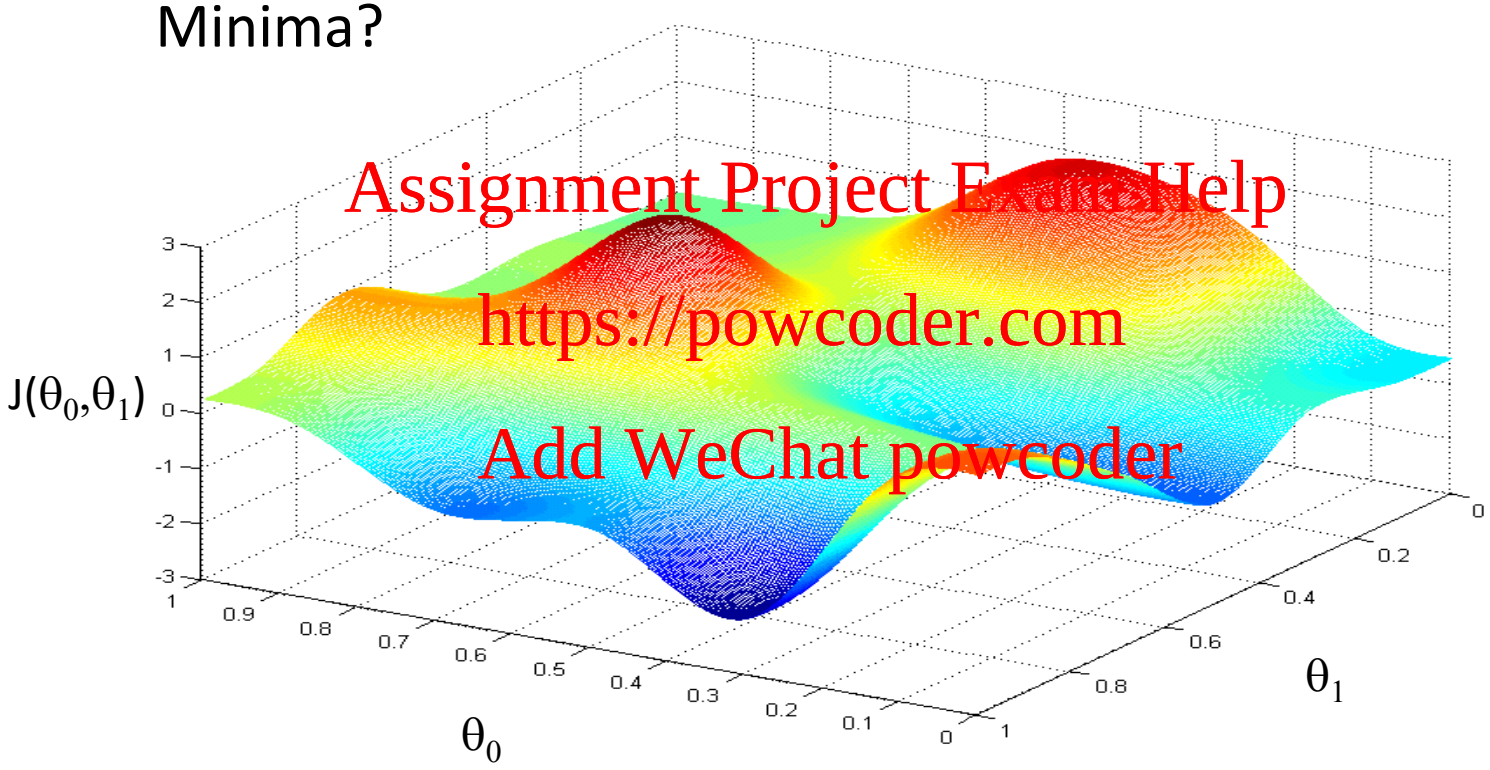
SSD cost function is convex

Minima?



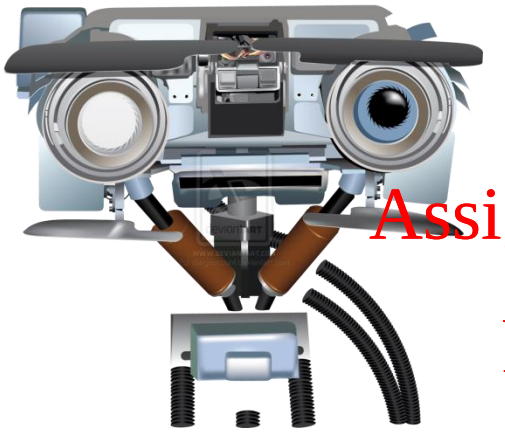
Non-convex cost function

Minima?



Later

- How to minimize the SSD cost function
  - Direct solution
  - Indirect solution



Introduction:

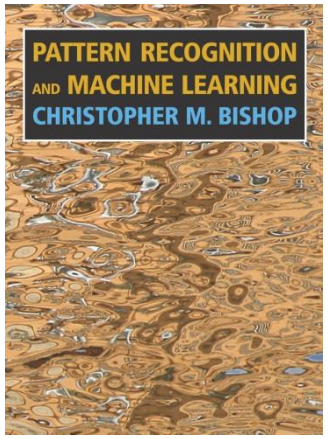
Course Overview

Class website

- Main class website

  https://piazza.com/bu/fall2020/cs542/home

Textbook



- Required textbook

Bishop, C. M. Pattern Recognition and Machine Learning. Springer. 2007

- Other suggested textbooks

Duda, R.O., Hart, P.E., and Stork, D.G. Pattern Classification. Wiley-Interscience. 2nd Edition. 2001.

Marsland, S. Machine Learning: An Algorithmic Perspective. CRC Press. 2009.Theodoridis, S. and Koutroumbas, K. Pattern Recognition. Edition 4. Academic Press, 2008.

Russell, S. and Norvig, N. Artificial Intelligence: A Modern Approach. Prentice Hall Series in Artificial Intelligence. 2003.

Bishop, C. M. Neural Networks for Pattern Recognition. Oxford University Press. 1995.

Hastie, T., Tibshirani, R. and Friedman, J. The Elements of StatisticalLearning. Springer. 2001.

Koller, D. and Friedman, N. Probabilistic Graphical Models. MIT Press. 2009.

Problem Sets

- Weekly problems sets
  - Python coding problems
  - Written math problems
  - Important to prepare you for the exams!
- Self-graded
  - you will submit code, answers, and your own grade
  - we will randomly check to verify

Class Challenge

Individual end-of-term project

- Based on a real-world problem, hosted as a Kaggle-like challenge for our class
- Goal is to design a machine learning approach and apply it to the problem
- Deliverables: github

Lecture Class Rotations

CAS 522 – 32 student capacity

- As of yesterday, 63 students have indicated they might attend in-person (or have not responded to the poll)
- Check Piazza for rotations before coming to class as they may shift during the semester
- Wipe down chairs before sitting down
- Wear a mask and be prepared to show your badge

Discussion/Lab Rotations

- Check Piazza for rotations before coming to class as they may shift during the semester
- As of yesterday, A3 and A4 require rotations, A2 and A5 don't need rotations (but may change, email me if you would like to change sections)
- Only attend the discussion section that you are registered for (especially if you want to attend in-person)
- Wipe down chairs before sitting down
- Wear a mask and be prepared to show your badge

Next Class

Preliminaries

review of expected mathematical skills for the course

- Reference reading on matrix calculus and linear algebra can be found here
- Matrix derivatives cheatsheet
- also see http://www.matrixcalculus.org/

Questions