10-301/10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Exam 3 Review

Matt Gormley
Lecture 27
Apr. 24, 2023

Reminders

- Homework 9: Learning Paradigms
  - Out: Fri, Apr. 21
  - Due: Thu, Apr. 27 at 11:59pm (only two grace/late days permitted)
- Exam 3 Practice Problems
  - Out: Tue, Apr 25
- Exam 3
  - Tue, May 2 (5:30pm – 7:30pm)
- Final Exit Poll (after Exam 3)

Crowdsourcing Exam Questions

In-Class Exercise

1. Select one of lecture-level learning objectives
   http://mlcourse.org/slides/10601-objectives.pdf
2. Write a question that assesses that objective
3. Adjust to avoid 'trivia style' question

Answer Here:

EXAM LOGISTICS

Exam 3

- **Time / Location**
  - **Time: Tue, May 2 at 5:30pm – 7:30pm**
  - **Location & Seats:** You have all been split across multiple rooms. Everyone has an assigned seat in one of these room.
  - Please watch Piazza carefully for announcements.
- **Logistics**
  - Covered material: Lectures 18 – 26
  - Format of questions:
    - Multiple choice
    - True / False (with justification)
    - Derivations
    - Short answers
    - Interpreting figures
    - Implementing algorithms on paper
  - No electronic devices
  - You are allowed to **bring** one 8½ x 11 sheet of notes (front and back)

Exam 3

- **How to Prepare**
  - Attend (or watch) this exam review session
  - Review **practice problems**
  - Review **homework problems**
  - Review the **poll questions** from each lecture
  - Consider whether you have achieved the **learning objectives** for each lecture / section
  - Write your cheat sheets

Topics for Exam 1

- Foundations
  - Probability, Linear Algebra, Geometry, Calculus
  - Optimization
- Important Concepts
  - Overfitting
  - Experimental Design
- Classification
  - Decision Tree
  - KNN
  - Perceptron
- Regression
  - Linear Regression

Topics for Exam 2

- Classification
  - Binary Logistic Regression
- Important Concepts
  - Stochastic Gradient Descent
  - Regularization
  - Feature Engineering
- Feature Learning
  - Neural Networks
  - Basic NN Architectures
  - Backpropagation
- Learning Theory
  - PAC Learning
- Generative Models
  - Generative vs. Discriminative
  - MLE / MAP
  - Naïve Bayes
- Regression
  - Linear Regression

Topics for Exam 3

- Graphical Models
  - HMMs
  - Learning and Inference
  - Bayesian Networks
- Reinforcement Learning
  - Value Iteration
  - Policy Iteration
  - Q-Learning
  - Deep Q-Learning
- Other Learning Paradigms
  - K-Means
  - PCA
  - Ensemble Methods
  - Recommender Systems

MATERIAL COVERED ON EXAM 1

Supervised Binary Classification

- Step 1: training
  - *Given*: labeled **training dataset**
  - *Goal*: learn a **classifier** from the training dataset
- Step 2: prediction
  - *Given*: unlabeled **test dat**
    : learned classifier
  - *Goal*: **predict** a label for e
    instance
- Step 3: evaluation
  - *Given*: **predictions** from
    : labeled **test datas**
  - *Goal*: compute the **test e** **rate** (i.e. error rate on th dataset)

Training Dataset:

	label	features		
index	**trash?**	**color**	**sound**	**weight**
1	+	green	crinkly	high
2	-	brown	crinkly	low
3	-	grey	none	high
4	+	clear	none	low
5	-	green	none	low

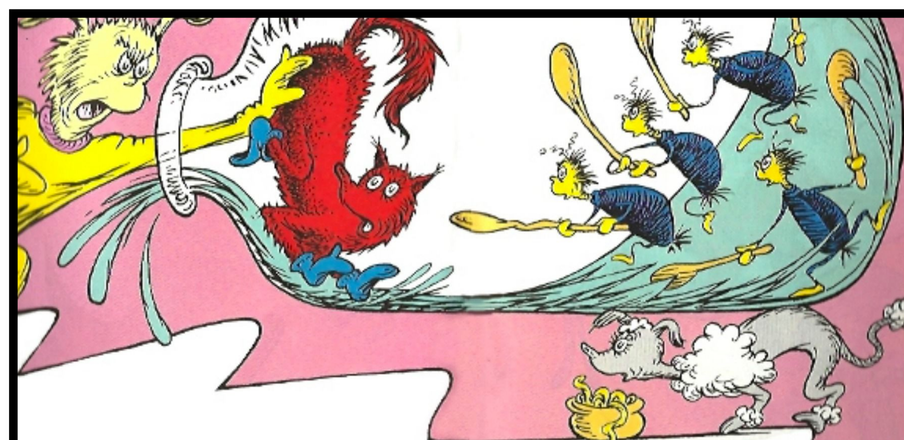
Key question in Machine Learning:

How do we learn the classifier from data?

Medical Diagnosis

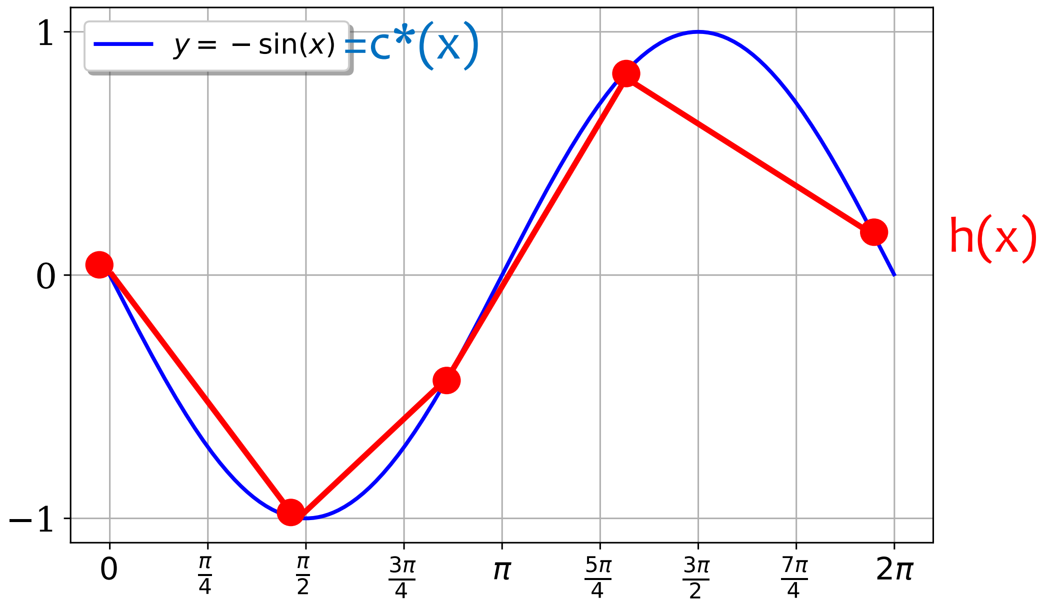
Interview Transcript
Date: Jan. 15, 2022
Parties: Matt Gormley and Doctor S.
Topic: Medical decision making

- Matt: Welcome. Thanks for interviewing with me today.
- Dr. S: Interviewing…?
- Matt: Yes. For the record, what type of doctor are you?
- Dr. S: Who said I'm a doctor?
- Matt: I thought when we set up this interview you said—
- Dr. S: I'm a preschooler.
- Matt: Good enough. Today, I'd like to learn how you would determine whether or not your little brother is allergic to cats given his symptoms.
- Dr. S: He's not allergic.
- Matt: We haven't started yet. Now, suppose he is sneezing. Does he have allergies to cats?
- Dr. S: Well, we don't even have a cat, so that doesn't make any sense.
- Matt: What if he is itchy; Does he have allergies?
- Dr. S: No, that's just a mosquito.
- [Editor's note: preschoolers unilaterally agree that itchiness is always caused by mosquitos, regardless of whether mosquitos were/are present.]
- Matt: What if he's both sneezing and itchy?
- Dr. S: Then he's allergic.
- Matt: Got it. What if your little brother is sneezing and itchy, plus he's a doctor.
- Dr. S: Then, thumbs down, he's not allergic.
- Matt: How do you know?
- Dr. S: Doctors don't get allergies.
- Matt: What if he is not sneezing, but is itchy, and he is a fox….
- Matt: …and the fox is in the bottle where the tweetle beetles battle with their paddles in a puddle on a noodle-eating poodle.
- Dr. S: Then he is must be a tweetle beetle noodle poodle bottled paddled muddled duddled fuddled wuddled fox in socks, sir. That means he's definitely allergic.
- Matt: Got it. Can I use this conversation in my lecture?
- Dr. S: Yes



Function Approximation

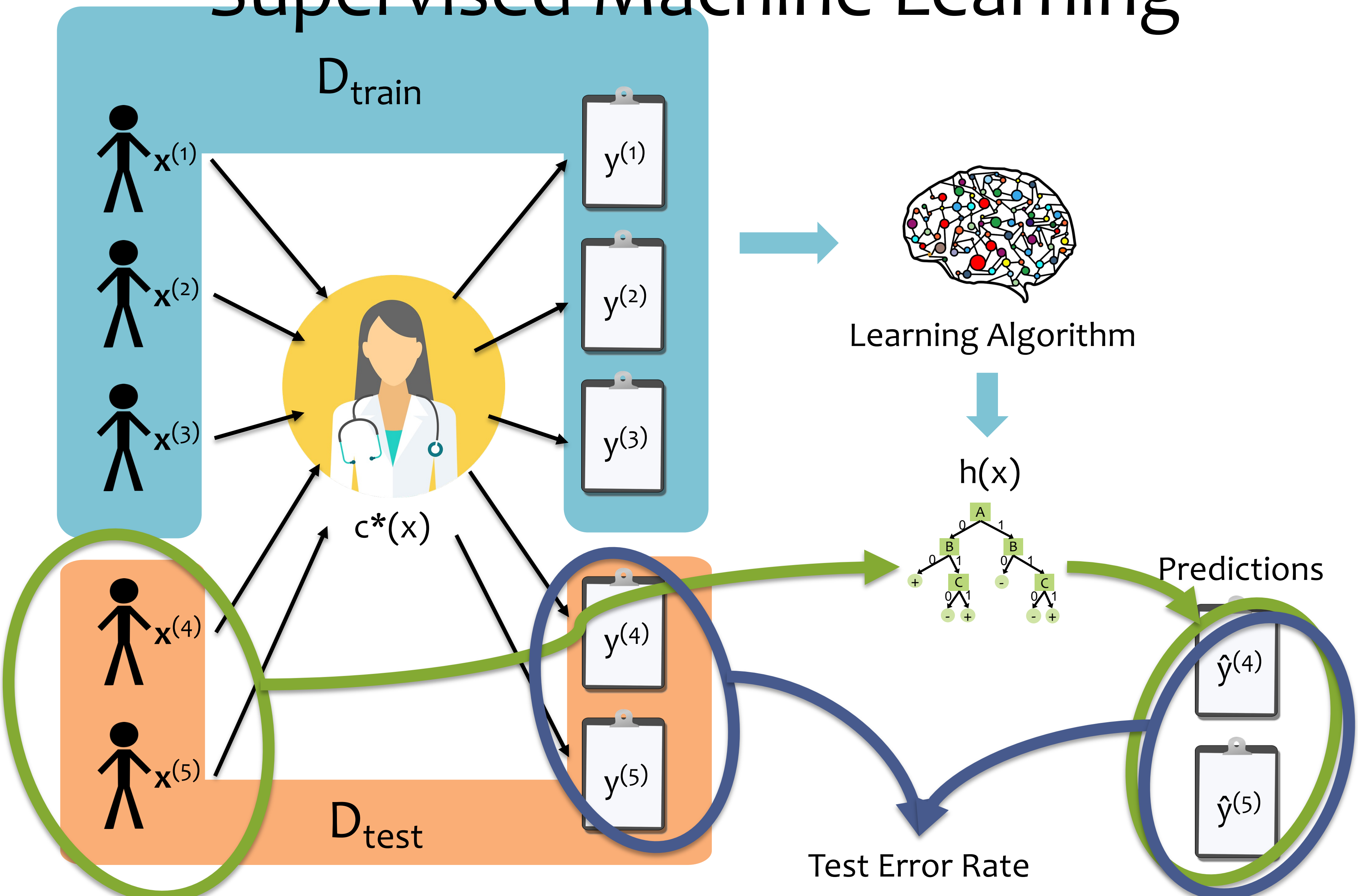
Quiz: Implement a simple function which returns -sin(x).

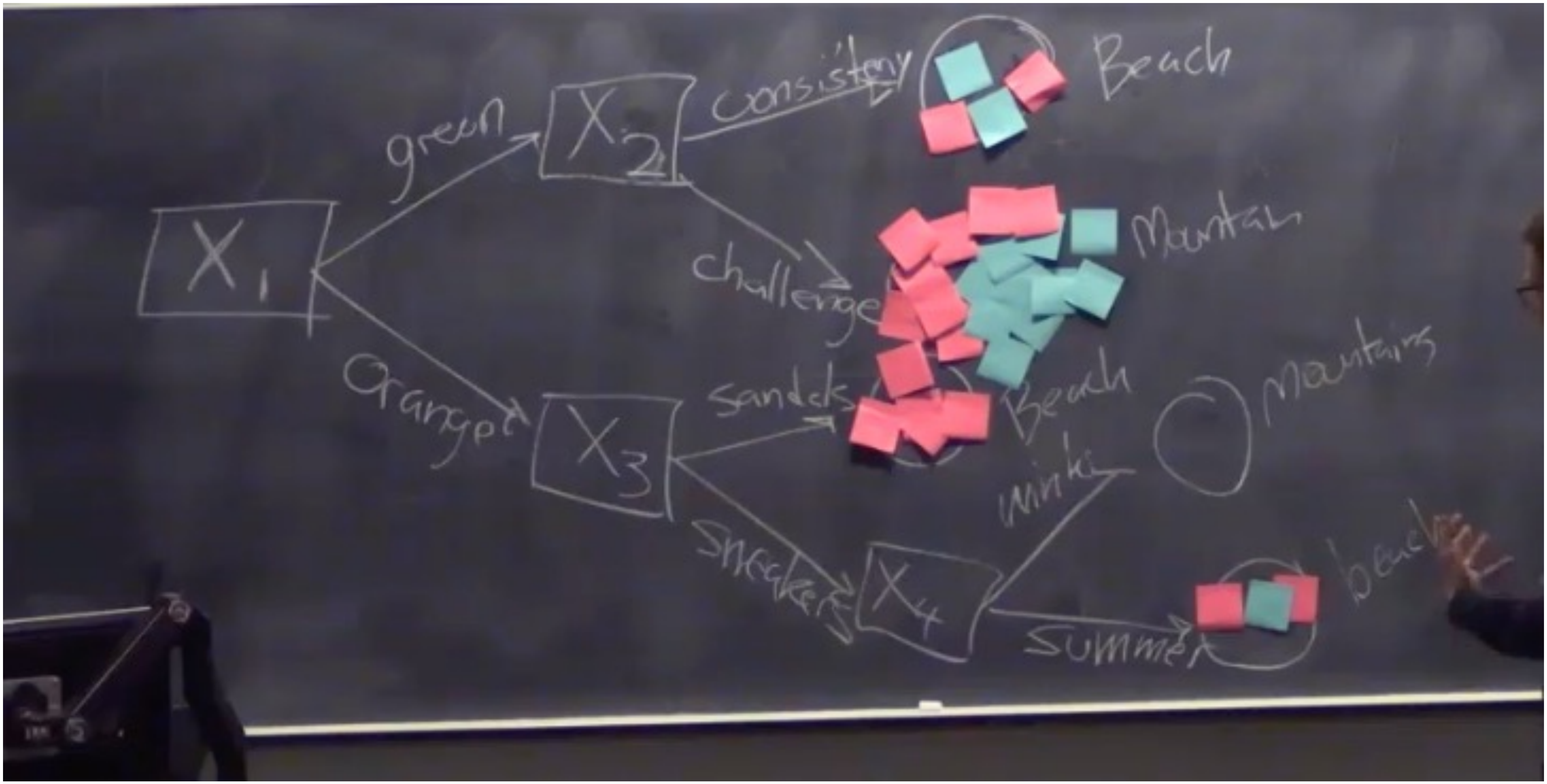


A few constraints are imposed:

1. You can't call any other trigonometric functions
2. You *can* call an existing implementation of sin(x) a few times (e.g. 100) to test your solution
3. You only need to evaluate it for x in [0, 2*pi]

Supervised Machine Learning





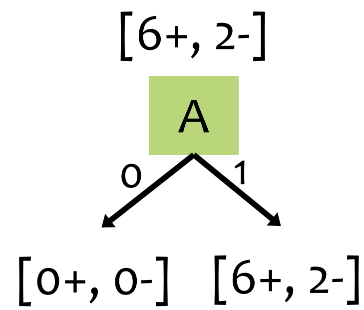
X_1
green
orange
X_2
X_3
X_4
consistency
challenge
sandals
sneakers
Beach
Mountain
Beach
mountains
winter
summer
beach

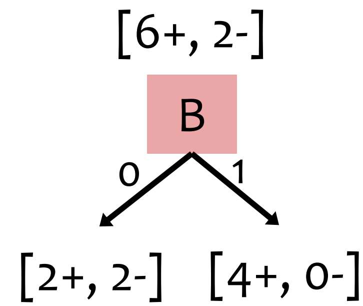
Decision Tree Learning Example

Dataset:

Output Y, Attributes A and B

Y	A	B
-	1	0
-	1	0
+	1	0
+	1	0
+	1	1
+	1	1
+	1	1
+	1	1





Mutual Information

$H(Y) = -2/8 \log(2/8) - 6/8 \log(6/8)$

$H(Y|A=0) =$ "undefined"

$H(Y|A=1) = -2/8 \log(2/8) - 6/8 \log(6/8)$
$= H(Y)$

$H(Y|A) = P(A=0)H(Y|A=0) + P(A=1)H(Y|A=1)$
$= 0 + H(Y|A=1) = H(Y)$

$I(Y; A) = H(Y) - H(Y|A=1) = 0$

$H(Y|B=0) = -2/4 \log(2/4) - 2/4 \log(2/4)$

$H(Y|B=1) = -0 \log(0) - 1 \log(1) = 0$

$H(Y|B) = 4/8(0) + 4/8(H(Y|B=0))$

$I(Y; B) = H(Y) - 4/8\, H(Y|B=0) > 0$

Overfitting in Decision Tree Learning

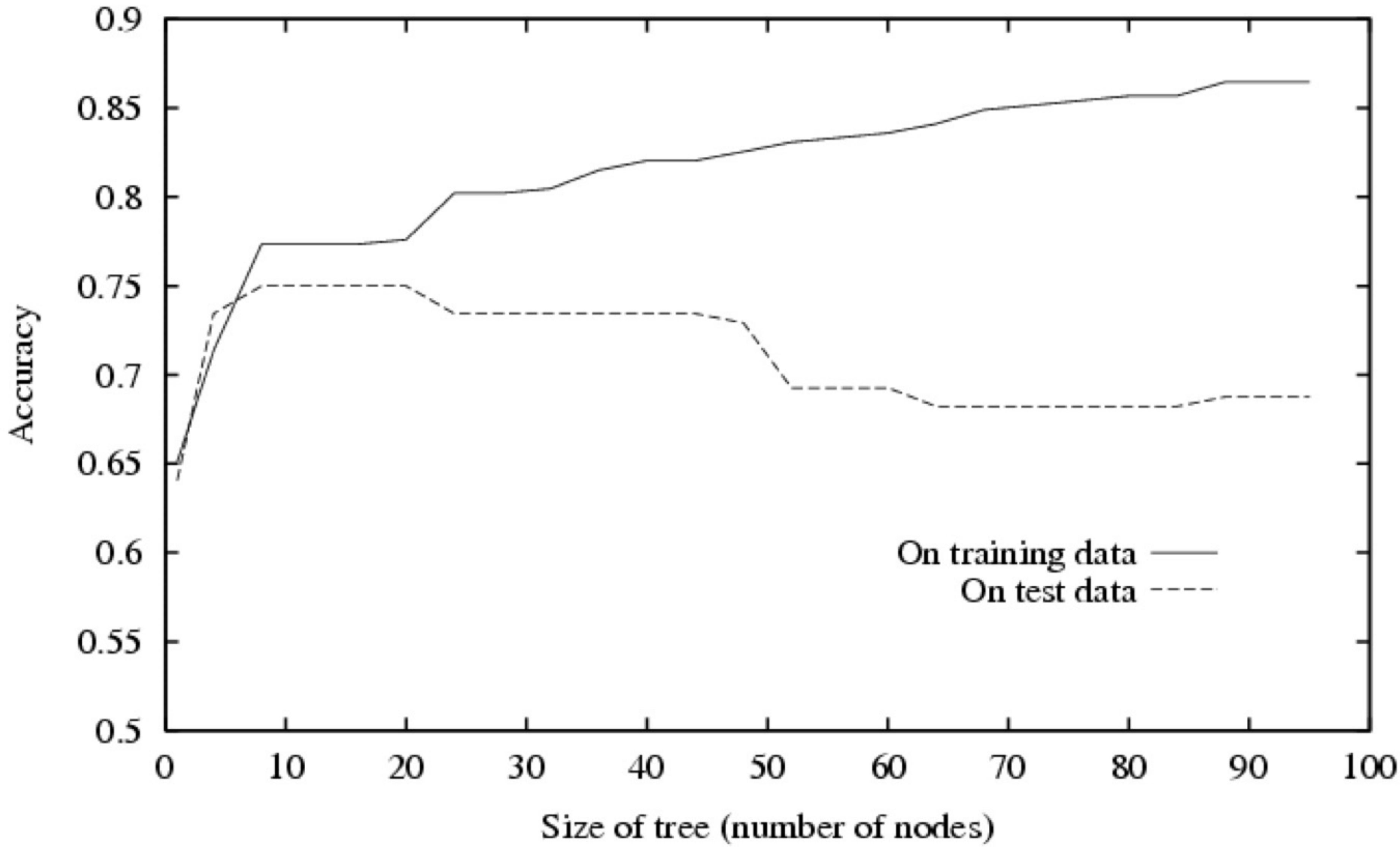
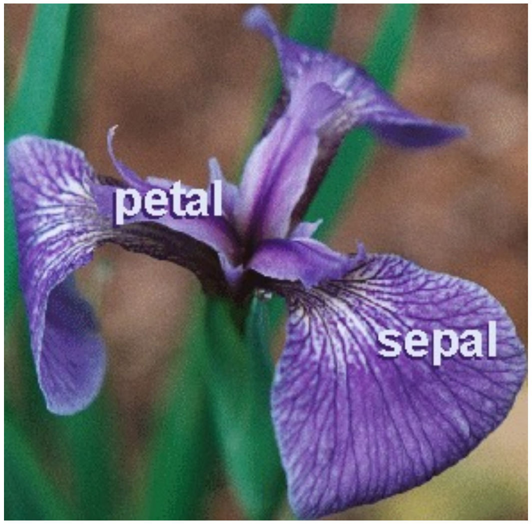


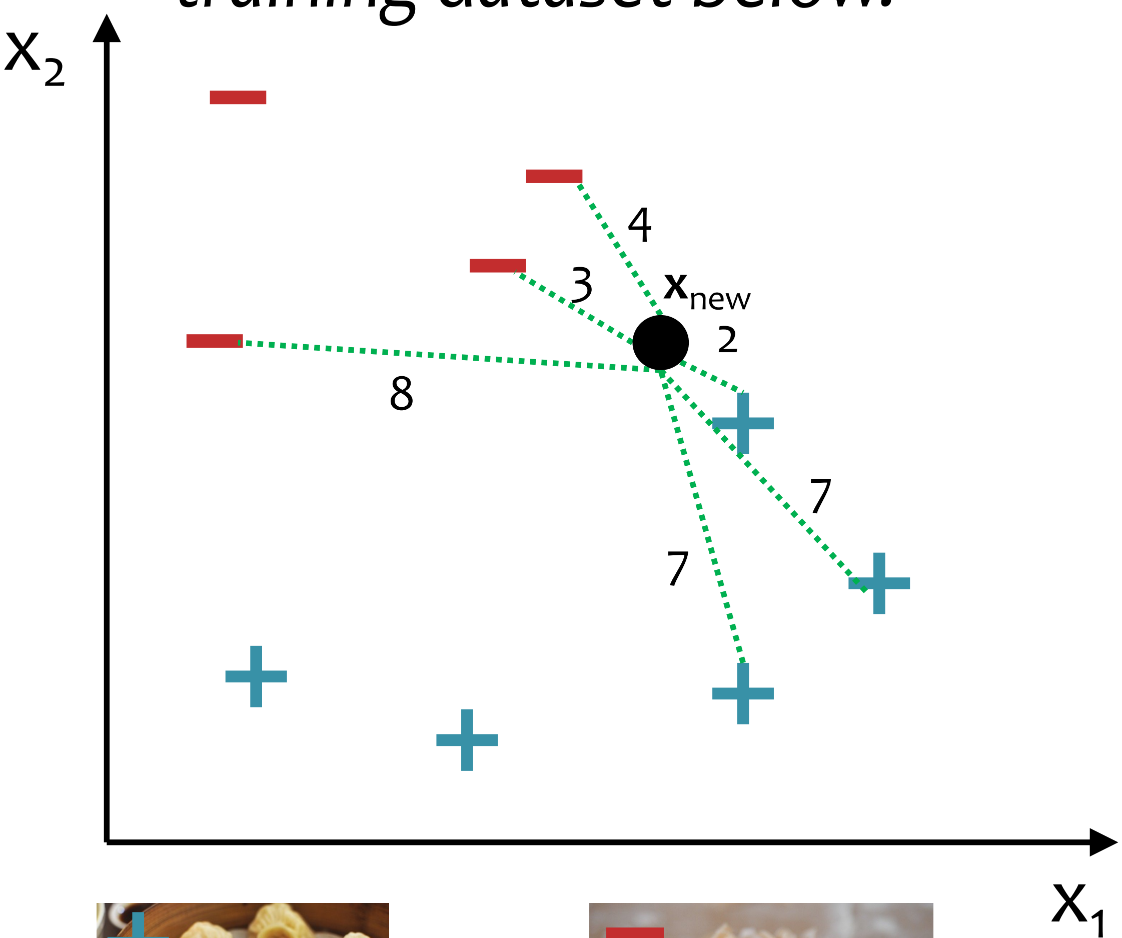
Figure from Tom Mitchell

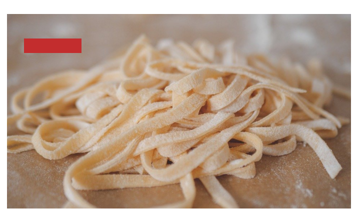


Species	Sepal Length	Sepal Width	Petal Length	Petal Width
0	4.3	3.0	1.1	0.1
0	4.9	3.6	1.4	0.1
0	5.3	3.7	1.5	0.2
1	4.9	2.4	3.3	1.0
1	5.7	2.8	4.1	1.3
1	6.3	3.3	4.7	1.6
1	6.7	3.0	5.0	1.7

k-Nearest Neighbors

Suppose we have the training dataset below.





How should we label the new point?

It depends on k:

if k=1, $h(\mathbf{x}_{new}) = +1$

if k=3, $h(\mathbf{x}_{new}) = -1$

if k=5, $h(\mathbf{x}_{new}) = +1$

Hyperparameter Optimization

Question:

True or False: given a finite amount of computation time, grid search is more likely to find good values for hyperparameters than random search.

Answer:

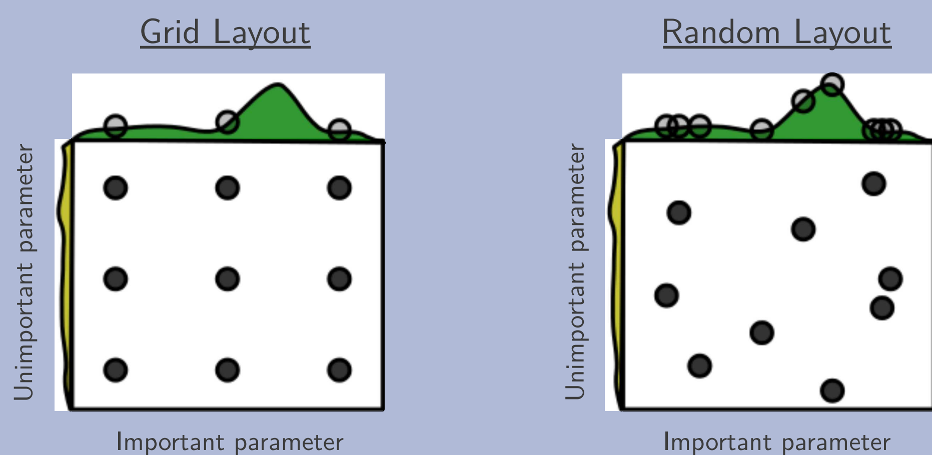
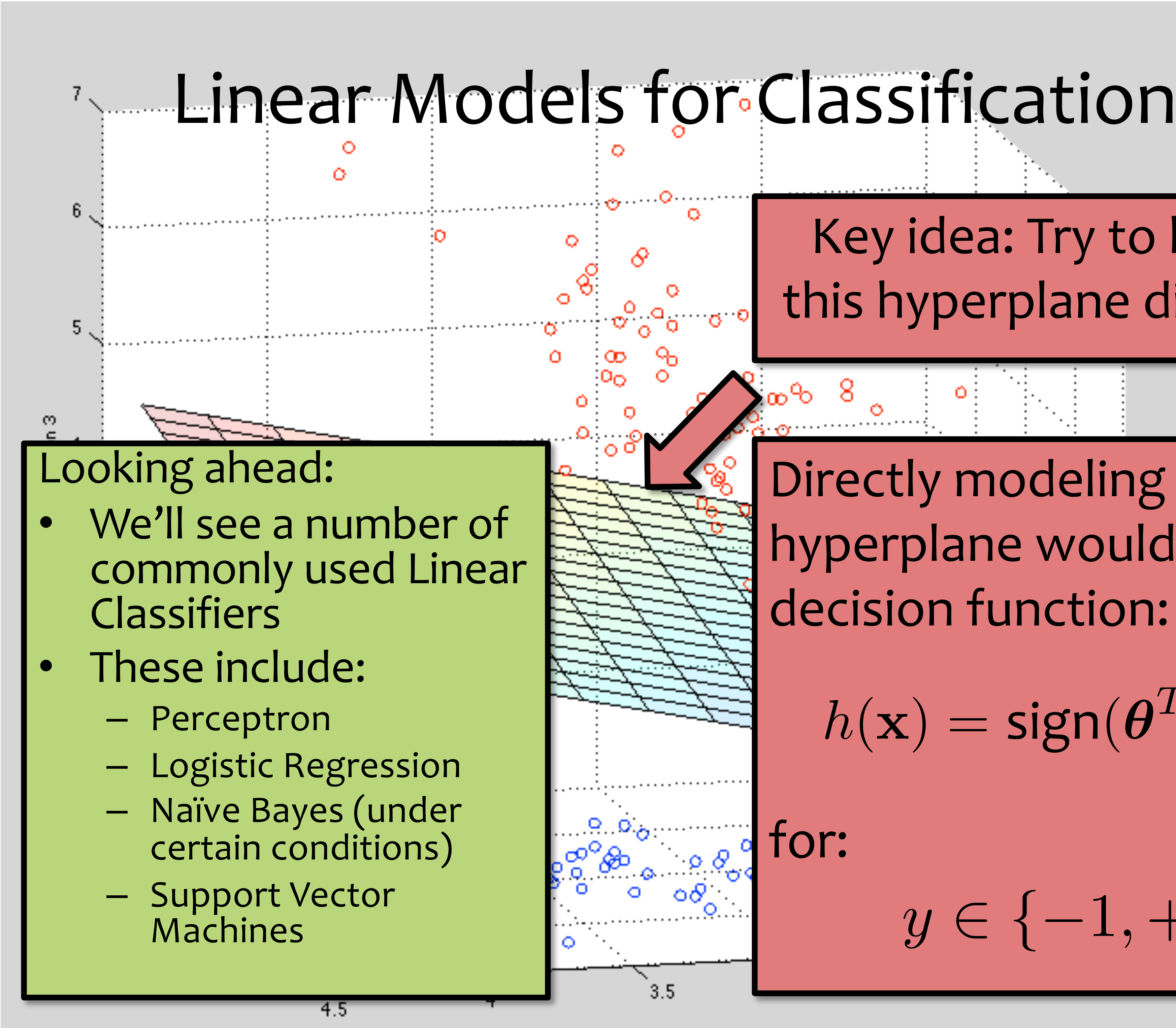


Figure 1: Grid and random search of nine trials for optimizing a function $f(x,y) = g(x) + h(y) \approx g(x)$ with low effective dimensionality. Above each square $g(x)$ is shown in green, and left of each square $h(y)$ is shown in yellow. With grid search, nine trials only test $g(x)$ in three distinct places. With random search, all nine trials explore distinct values of g. This failure of grid search is the rule rather than the exception in high dimensional hyper-parameter optimization.

Figure from Bergstra & Bengio (2012)

Linear Models for Classification



Key idea: Try to learn this hyperplane directly

Directly modeling the hyperplane would use a decision function:

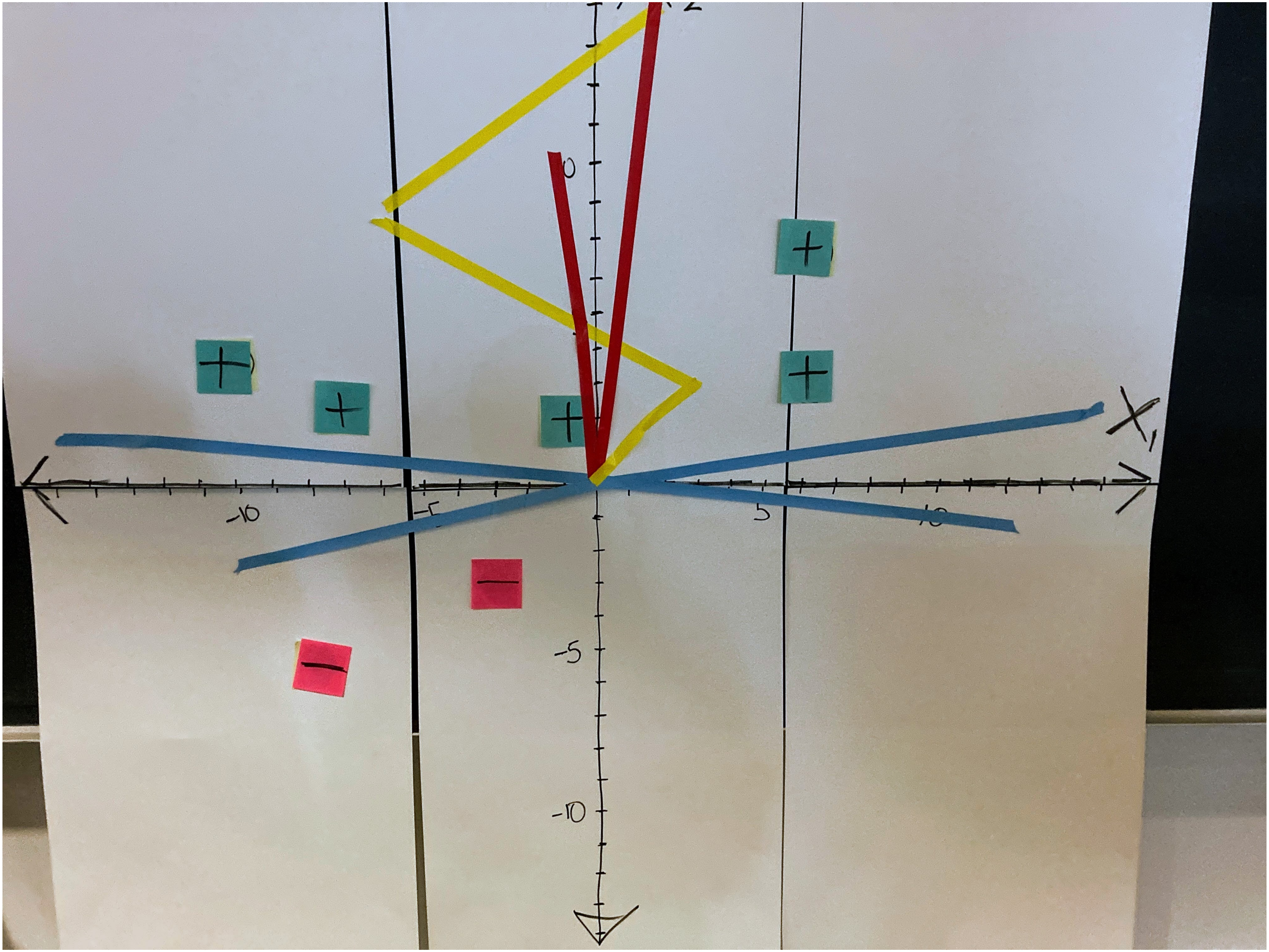
$$h(\mathbf{x}) = \text{sign}(\boldsymbol{\theta}^T \mathbf{x})$$

for:

$$y \in \{-1, +1\}$$

Looking ahead:

- We'll see a number of commonly used Linear Classifiers
- These include:
  - Perceptron
  - Logistic Regression
  - Naïve Bayes (under certain conditions)
  - Support Vector Machines

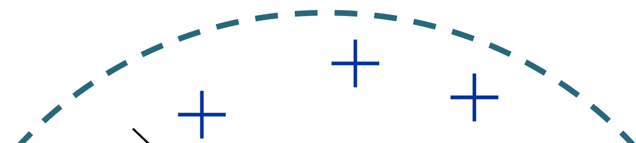


10
-10
-5
5
10
-5
-10
X₁
+
+
+
+
+
+
−
−

Perceptron Mistake Bound

Guarantee: if some data has margin γ and all points lie inside a ball of radius R, then the online Perceptron algorithm makes $\leq (R/\gamma)^2$ mistakes

(Normalized margin: multiplying all points by 100, or dividing all points by 100, doesn't change the number of mistakes! The algorithm is invariant to scaling.)



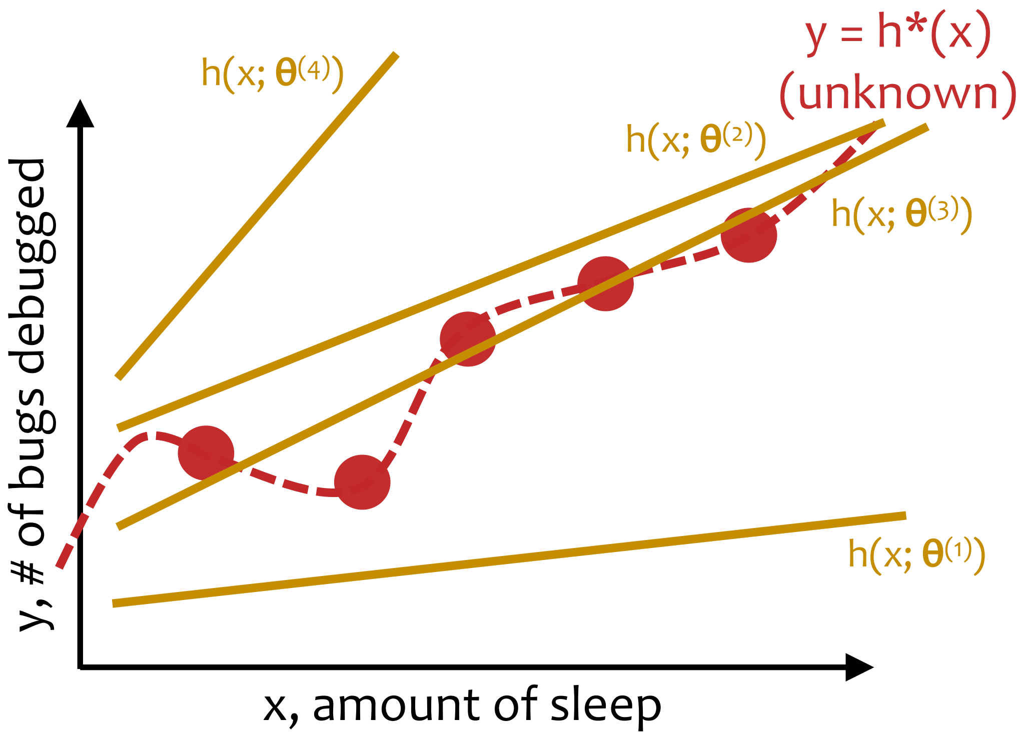
Def: We say that the (batch) perceptron algorithm has **converged** if it stops making mistakes on the training data (perfectly classifies the training data).

Main Takeaway: For **linearly separable** data, if the perceptron algorithm cycles repeatedly through the data, it will **converge** in a finite # of steps.

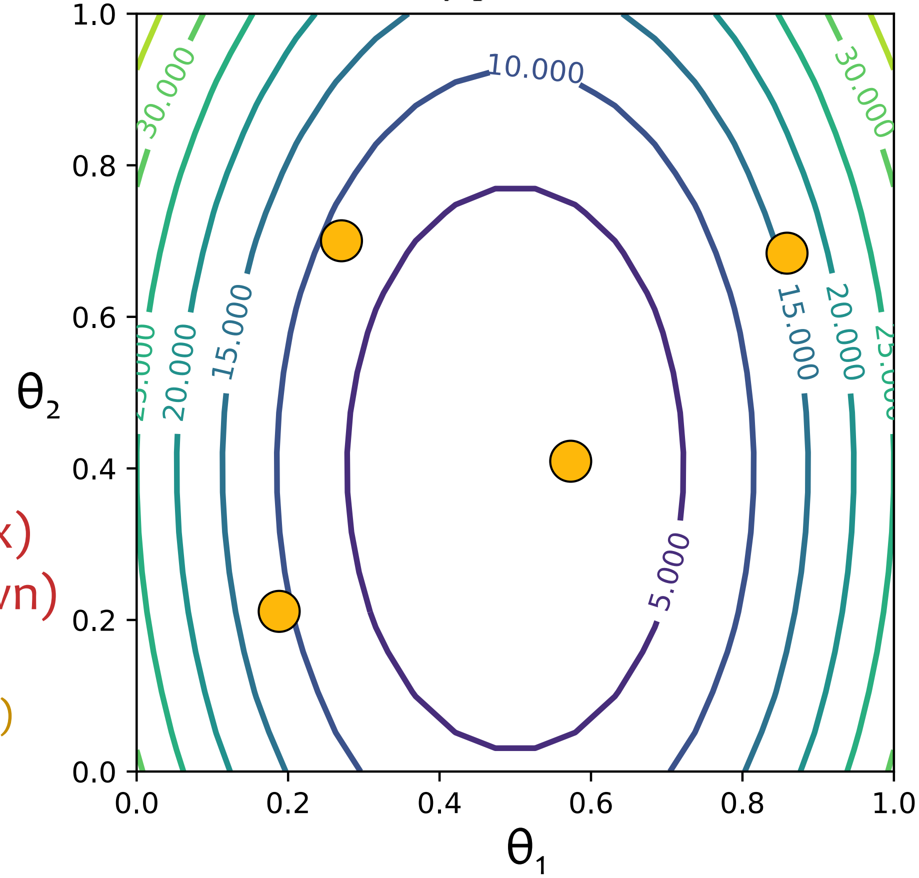
Linear Regression by Rand. Guessing

Optimization Method #0: Random Guessing

1. Pick a random $\boldsymbol{\theta}$
2. Evaluate $J(\boldsymbol{\theta})$
3. Repeat steps 1 and 2 many times
4. Return $\boldsymbol{\theta}$ that gives smallest $J(\boldsymbol{\theta})$

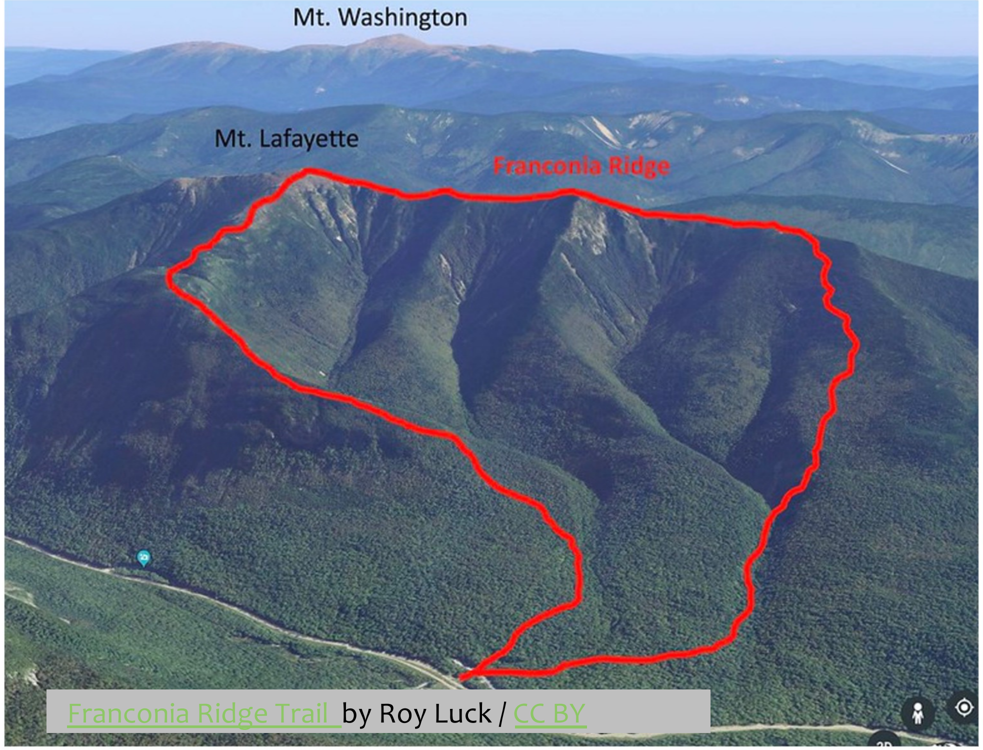


$$J(\boldsymbol{\theta}) = J(\theta_1, \theta_2) = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)^2$$

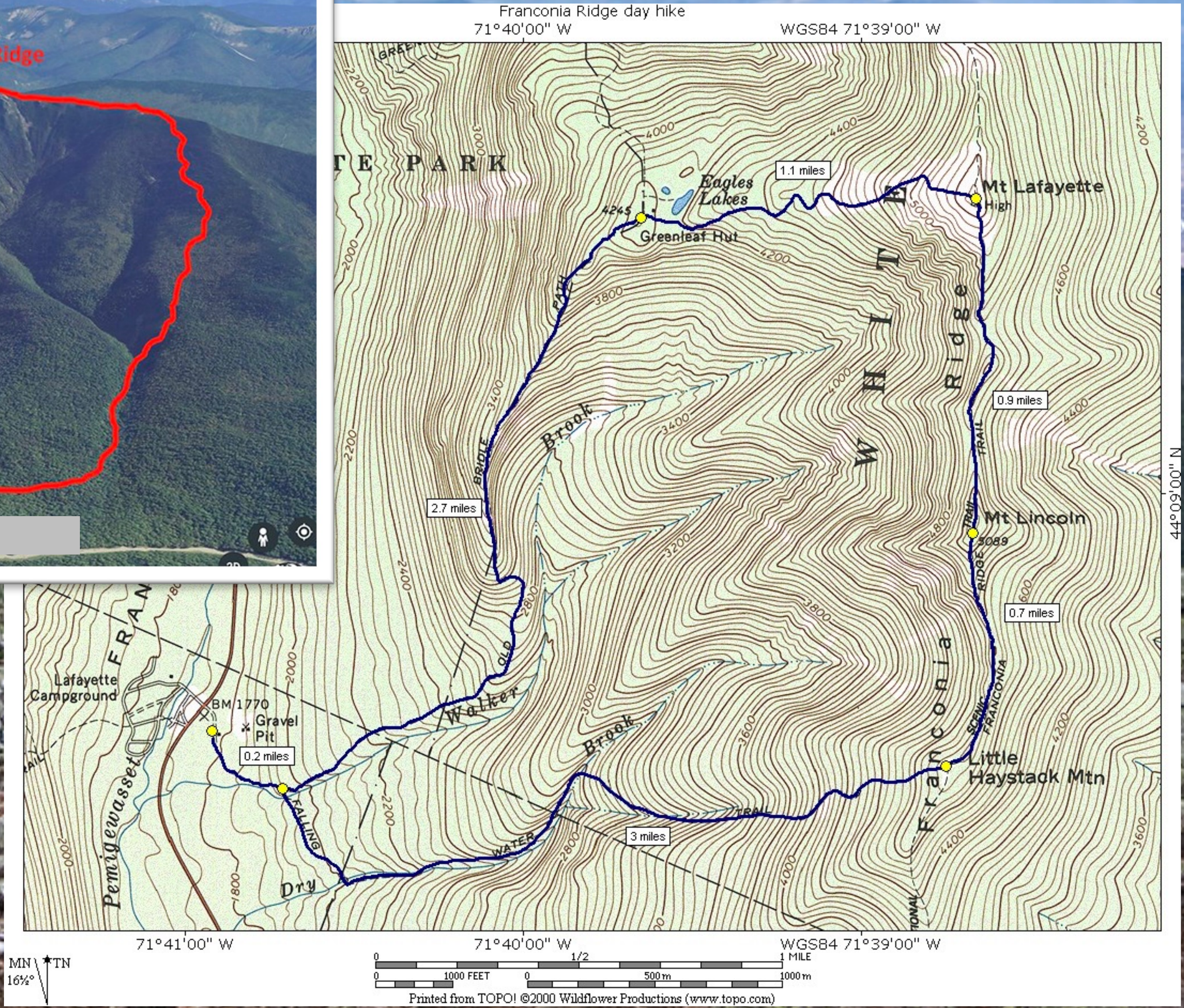


t	θ_1	θ_2	$J(\theta_1, \theta_2)$
1	0.2	0.2	10.4
2	0.3	0.7	7.2
3	0.6	0.4	1.0
4	0.9	0.7	16.2

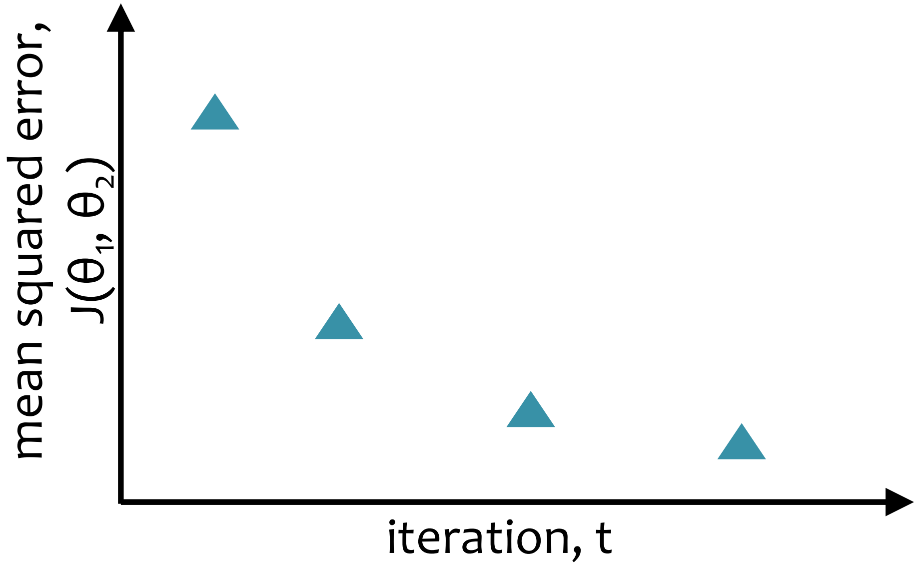
Topographical Maps

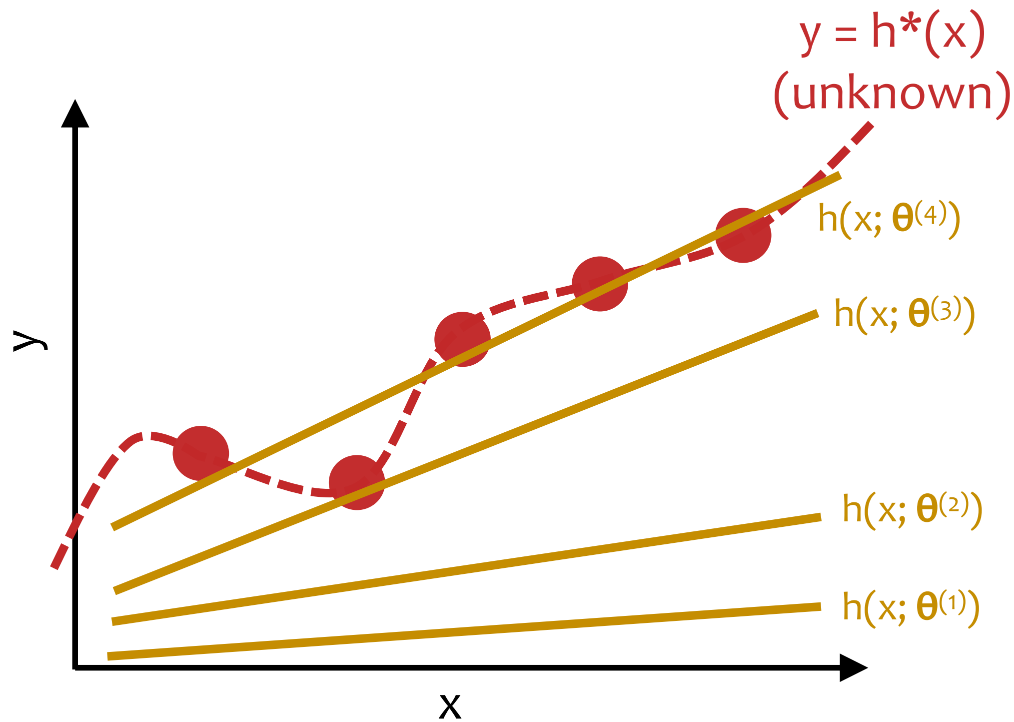


Franconia Ridge Trail by Roy Luck / CC BY

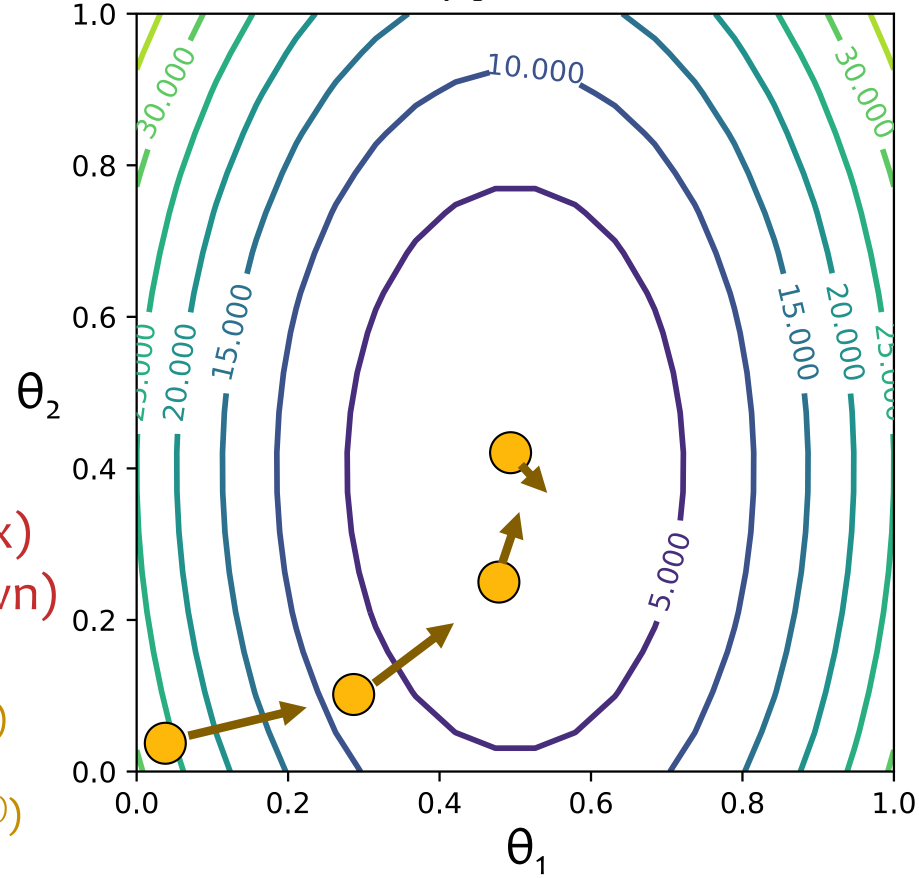


Linear Regression by Gradient Desc.





$$J(\boldsymbol{\theta}) = J(\theta_1, \theta_2) = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)^2$$

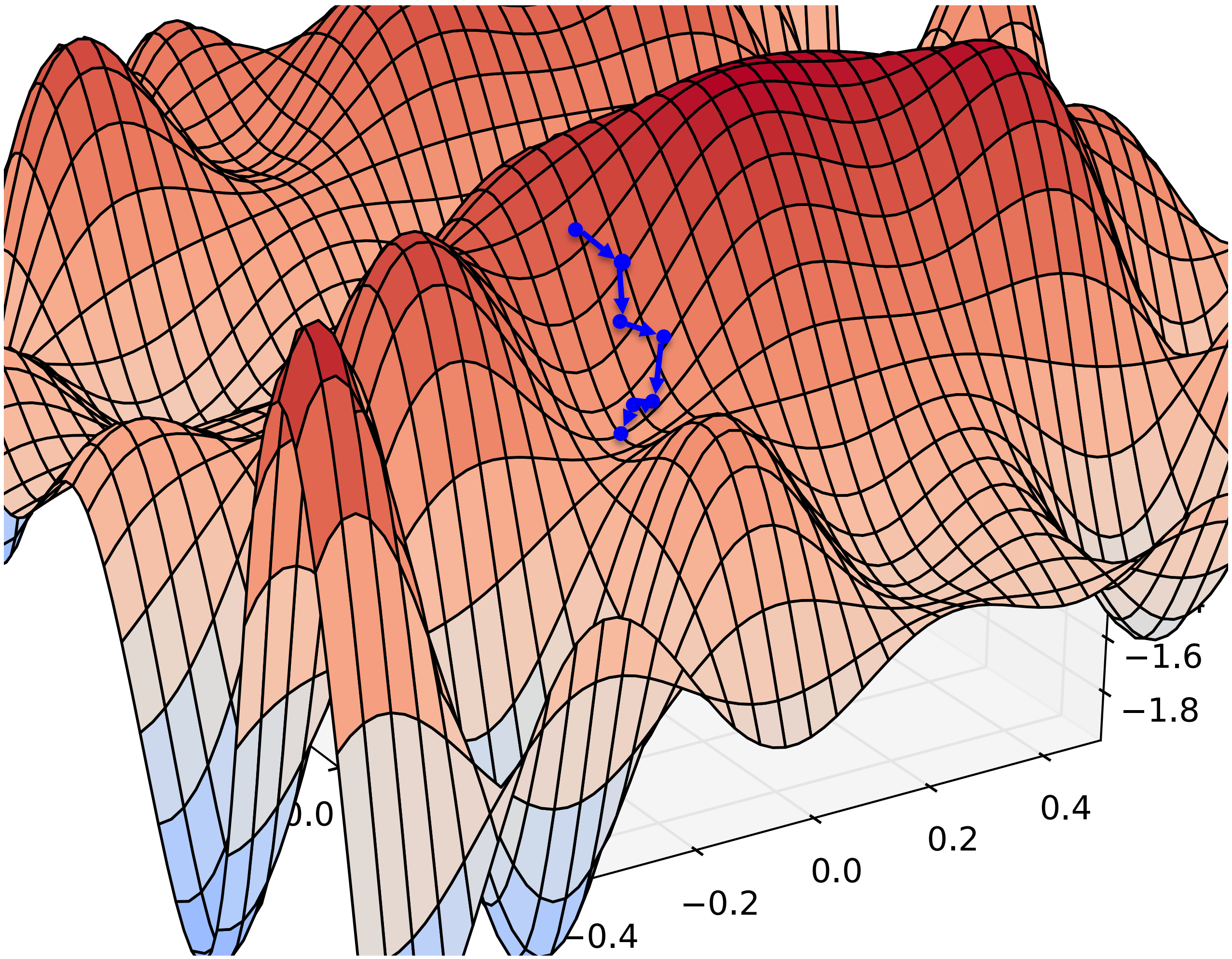


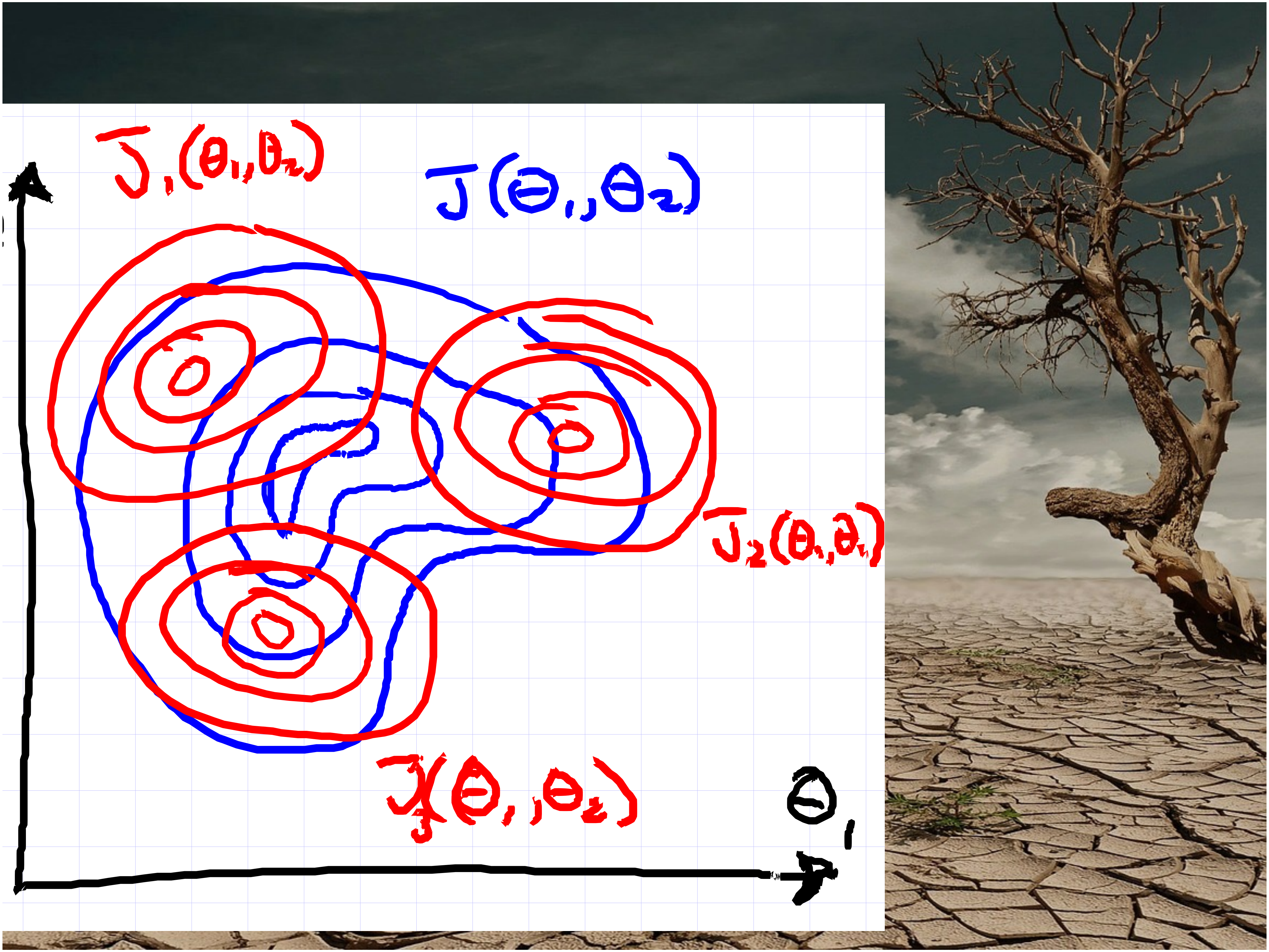
t	θ_1	θ_2	$J(\theta_1, \theta_2)$
1	0.01	0.02	25.2
2	0.30	0.12	8.7
3	0.51	0.30	1.5
4	0.59	0.43	0.2

MATERIAL COVERED ON EXAM 2

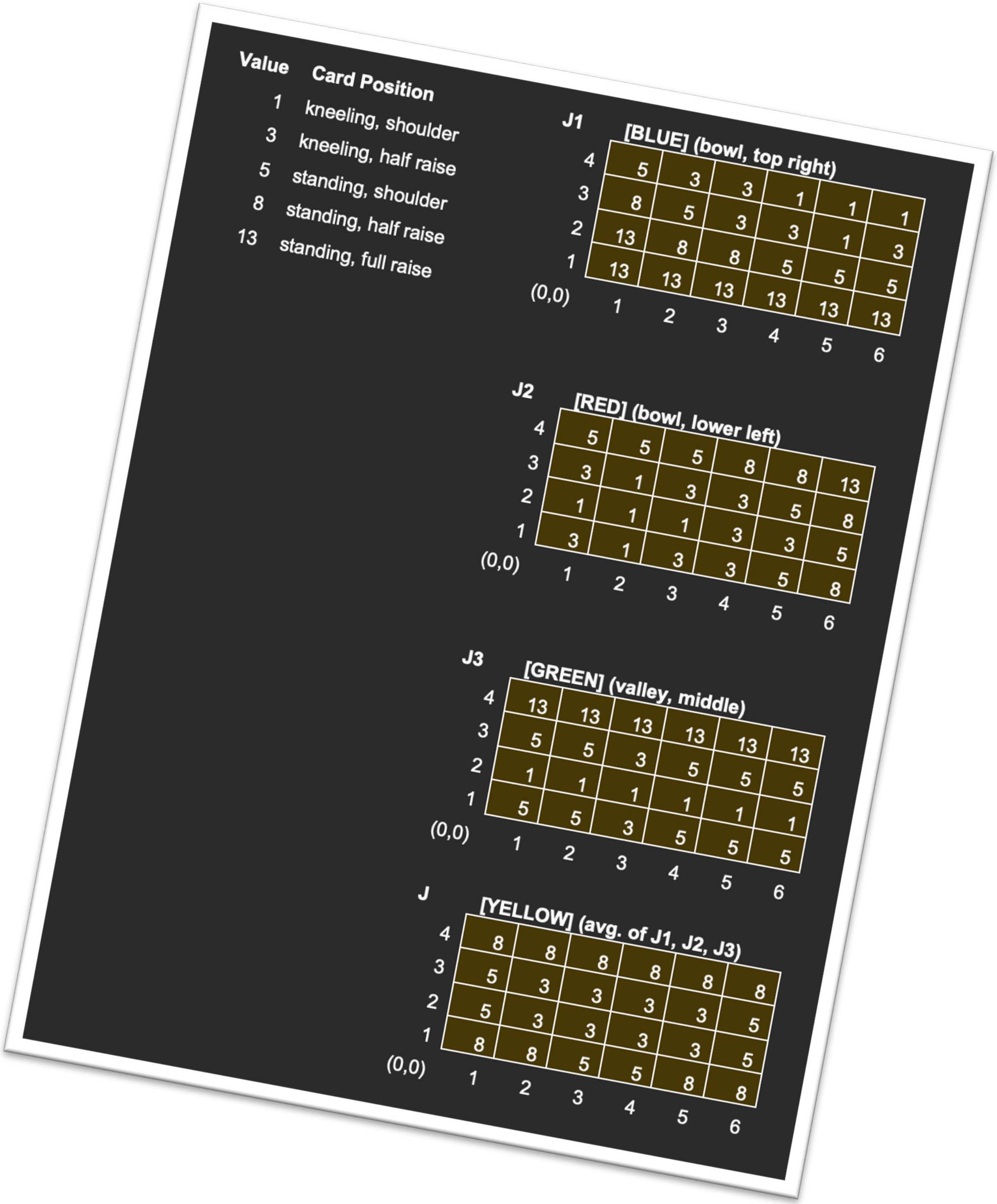
Gradient Descent & Convexity

- Gradient descent is a **local optimization algorithm**
- If the function is **nonconvex**, it will find a local minimum, not necessarily a global minimum
- If the function is **convex**, it will find a global minimum





J1(θ1,θ2)
J(θ1,θ2)
J2(θ1,θ2)
J3(θ1,θ2)
θ1



Value	Card Position
1	kneeling, shoulder
3	kneeling, half raise
5	standing, shoulder
8	standing, half raise
13	standing, full raise

J1 [BLUE] (bowl, top right)

(0,0)	1	2	3	4	5	6
4	5	3	3	1	1	1
3	8	5	3	3	1	3
2	13	8	8	5	5	5
1	13	13	13	13	13	13

J2 [RED] (bowl, lower left)

(0,0)	1	2	3	4	5	6
4	5	5	5	8	8	13
3	3	1	3	3	5	8
2	1	1	1	3	3	5
1	3	1	3	3	5	8

J3 [GREEN] (valley, middle)

(0,0)	1	2	3	4	5	6
4	13	13	13	13	13	13
3	5	5	3	5	5	5
2	1	1	1	1	1	1
1	5	5	3	5	5	5

J [YELLOW] (avg. of J1, J2, J3)

(0,0)	1	2	3	4	5	6
4	8	8	8	8	8	8
3	5	3	3	3	3	5
2	5	3	3	3	3	5
1	8	8	5	5	8	8

Probabilistic Learning

Function Approximation

Previously, we assumed that our output was generated using a **deterministic target function:**

$$\mathbf{x}^{(i)} \sim p^*(\cdot)$$
$$y^{(i)} = c^*(\mathbf{x}^{(i)})$$

Our goal was to learn a hypothesis h(**x**) that best approximates c*(**x**)

Probabilistic Learning

Today, we assume that our output is **sampled** from a conditional **probability distribution:**

$$\mathbf{x}^{(i)} \sim p^*(\cdot)$$
$$y^{(i)} \sim p^*(\cdot|\mathbf{x}^{(i)})$$

Our goal is to learn a probability distribution p(y|**x**) that best approximates p*(y|**x**)

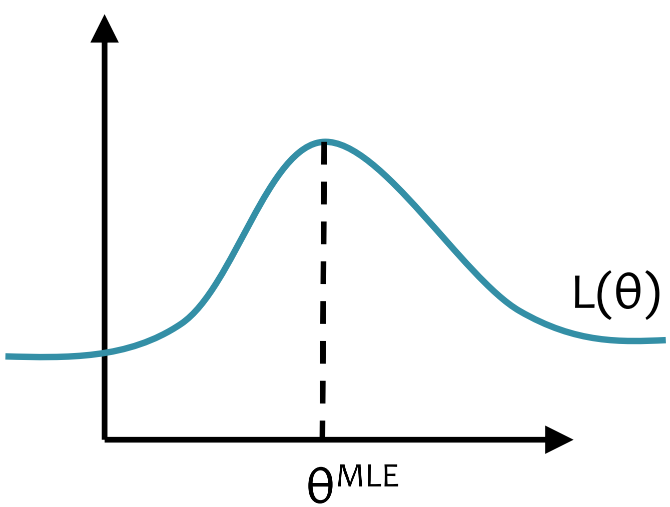
MLE

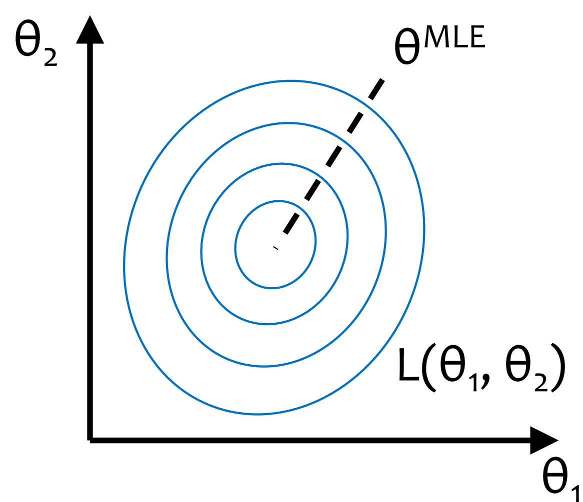
Suppose we have data $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$

Principle of Maximum Likelihood Estimation:
Choose the parameters that maximize the likelihood of the data.

$$\boldsymbol{\theta}^{\text{MLE}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)} | \boldsymbol{\theta})$$

Maximum Likelihood Estimate (MLE)





Logistic Regression

Data: Inputs are continuous vectors of length M. Outputs are discrete.

$$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N} \text{ where } \mathbf{x} \in \mathbb{R}^M \text{ and } y \in \{0, 1\}$$

Model: Logistic function applied to dot product of parameters with input vector.

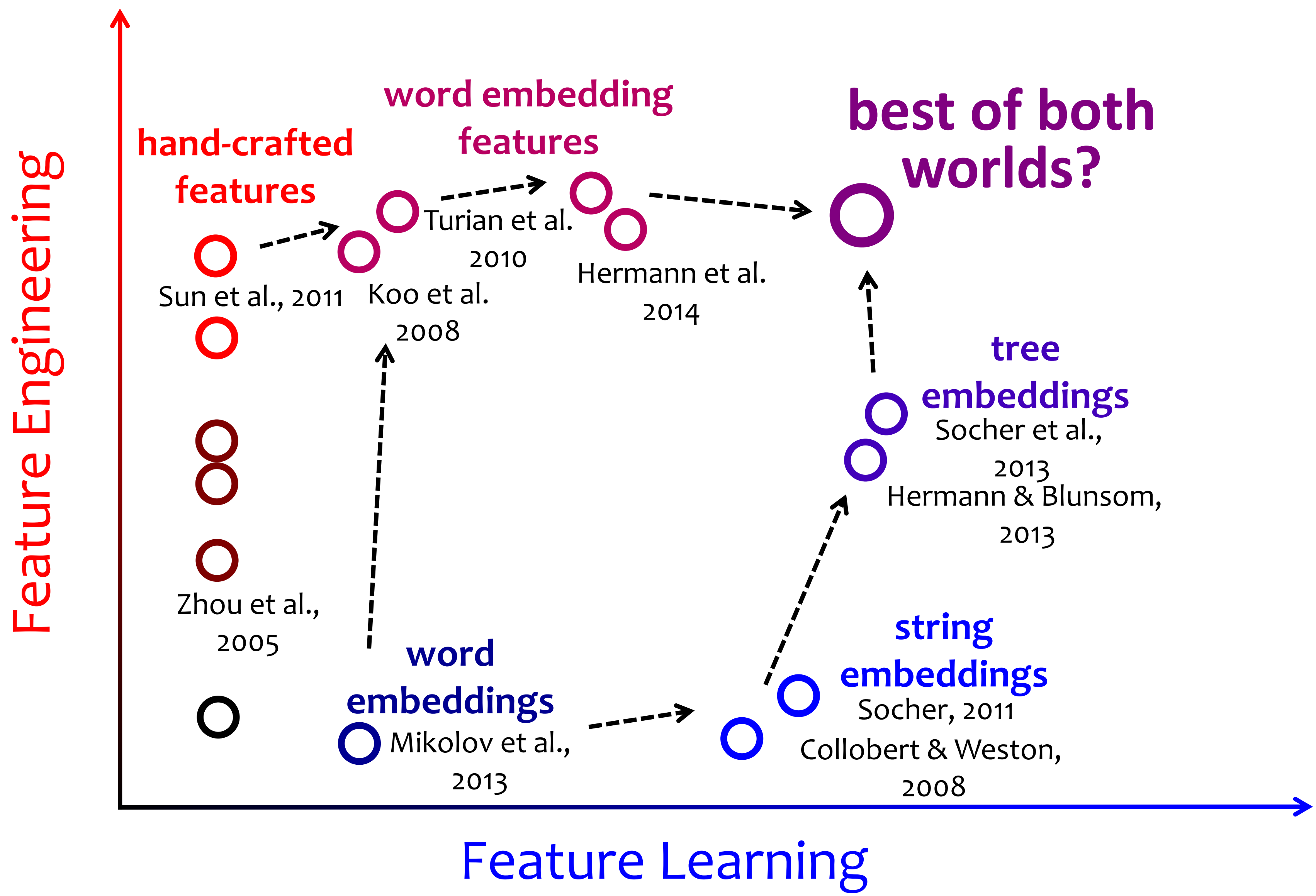
$$p_{\boldsymbol{\theta}}(y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \mathbf{x})}$$

Learning: finds the parameters that minimize some objective function.
$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

Prediction: Output is the most probable class.

$$\hat{y} = \operatorname*{argmax}_{y \in \{0,1\}} p_{\boldsymbol{\theta}}(y|\mathbf{x})$$

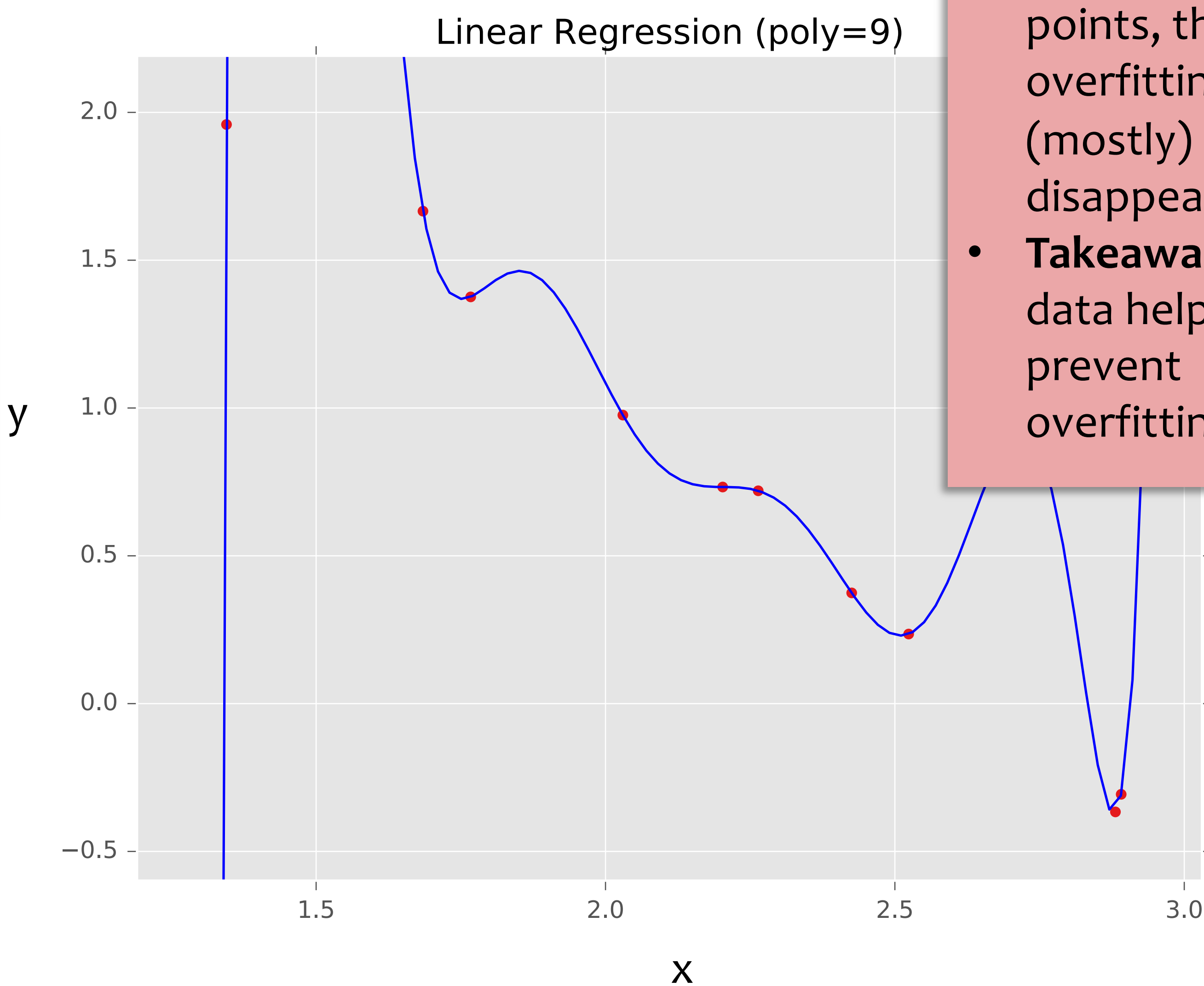
Where do features come from?



Example: Linear Regression

Goal: Learn y = $\mathbf{w}^T f(\mathbf{x}) + b$ where f(.) is a polynomial basis function

i	y	x	…	x^9
1	2.0	1.2	…	$(1.2)^9$
2	1.3	1.7	…	$(1.7)^9$
…	…	…	…	…
10	1.1	1.9	…	$(1.9)^9$

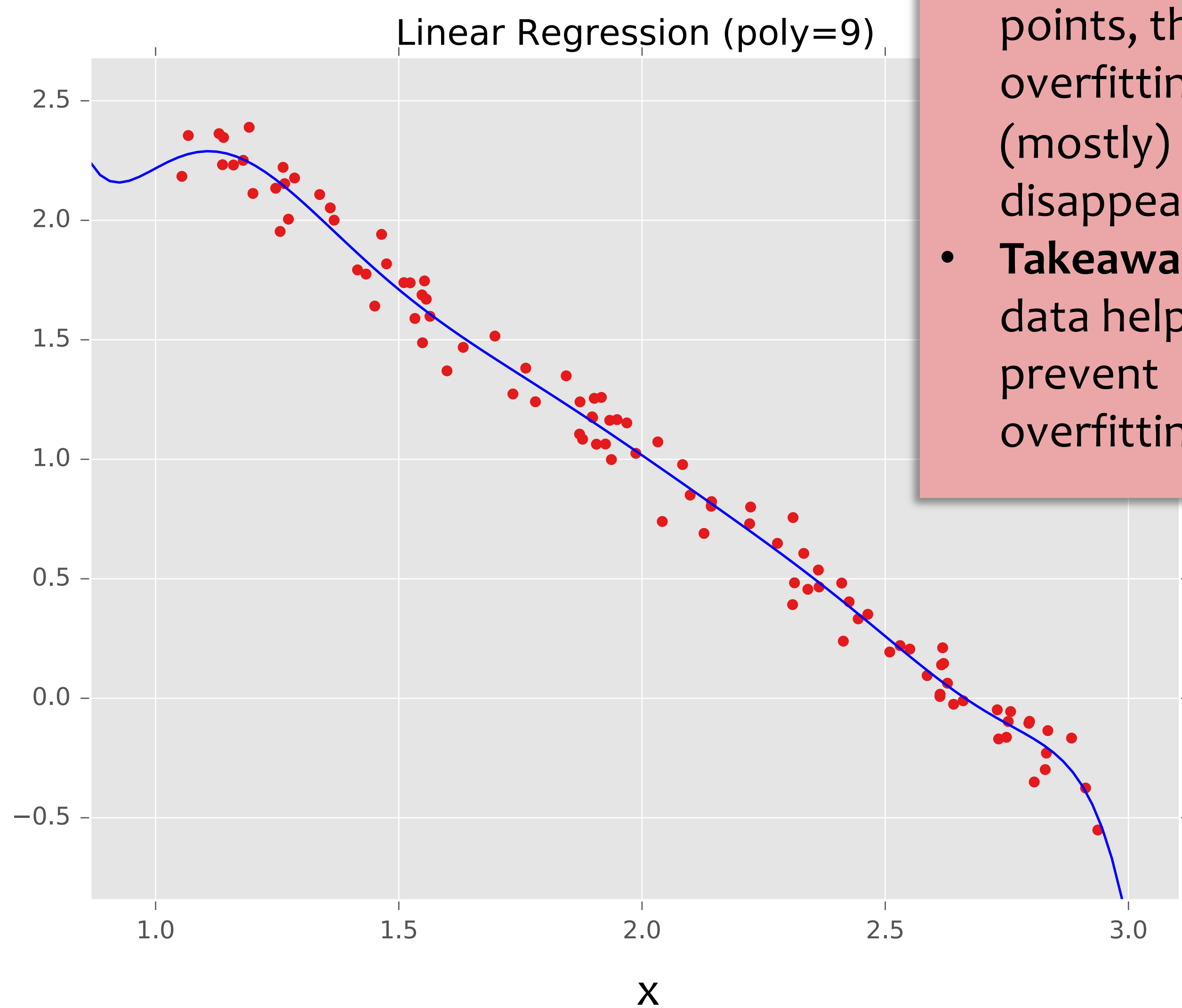


- With just N = 10 points we overfit!
- But with N = 100 points, the overfitting (mostly) disappears
- **Takeaway:** more data helps prevent overfitting

Example: Linear Regression

Goal: Learn y = $\mathbf{w}^T f(\mathbf{x}) + b$ where f(.) is a polynomial basis function

i	y	x	…	x^9
1	2.0	1.2	…	$(1.2)^9$
2	1.3	1.7	…	$(1.7)^9$
3	0.1	2.7	…	$(2.7)^9$
4	1.1	1.9	…	$(1.9)^9$
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
98	…	…	…	…
99	…	…	…	…
100	0.9	1.5	…	$(1.5)^9$



- With just N = 10 points we overfit!
- But with N = 100 points, the overfitting (mostly) disappears
- **Takeaway:** more data helps prevent overfitting

Regularization

- **Given** objective function: J(θ)
- **Goal** is to find: $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} J(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta})$
- **Key idea:** Define regularizer r(θ) s.t. we tradeoff between fitting the data and keeping the model simple
- **Choose form of r(θ):**
  - Example: q-norm (usually p-norm): $||\boldsymbol{\theta}||_q = \left(\sum_{m=1}^{M} |\theta_m|\right)^{\frac{1}{q}}$

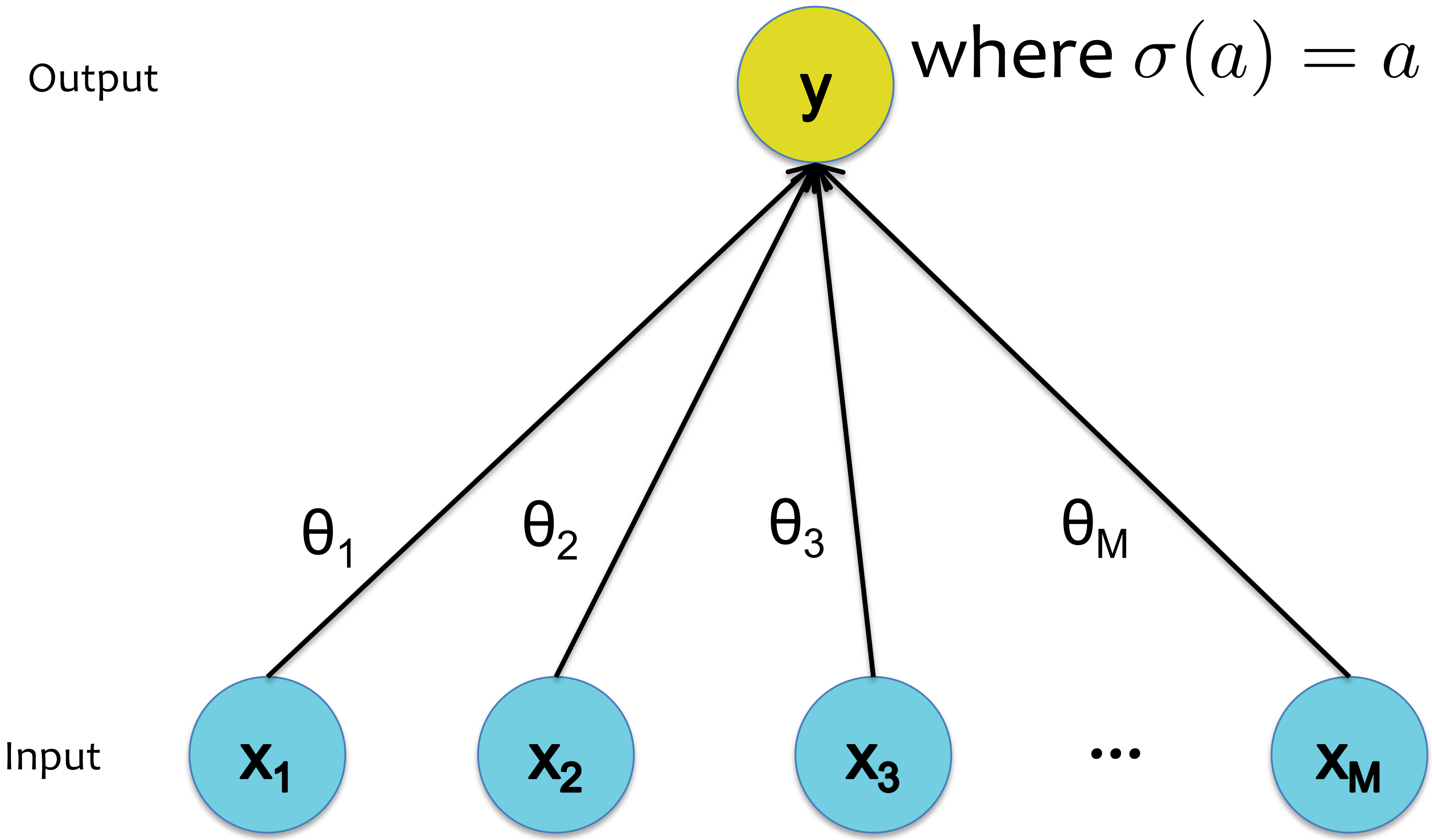
q	$r(\boldsymbol{\theta})$	yields parameters that are...	name	optimization notes
0	$\|\|\boldsymbol{\theta}\|\|_0 = \sum \mathbb{1}(\theta_m \neq 0)$	zero values	L0 reg.	no good computational solutions
1	$\|\|\boldsymbol{\theta}\|\|_1 = \sum \|\theta_m\|$	zero values	L1 reg.	subdifferentiable
2	$(\|\|\boldsymbol{\theta}\|\|_2)^2 = \sum \theta_m^2$	small values	L2 reg.	differentiable

Decision Functions

Linear Regression

$$y = h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sigma(\boldsymbol{\theta}^T \boldsymbol{x})$$

where $\sigma(a) = a$

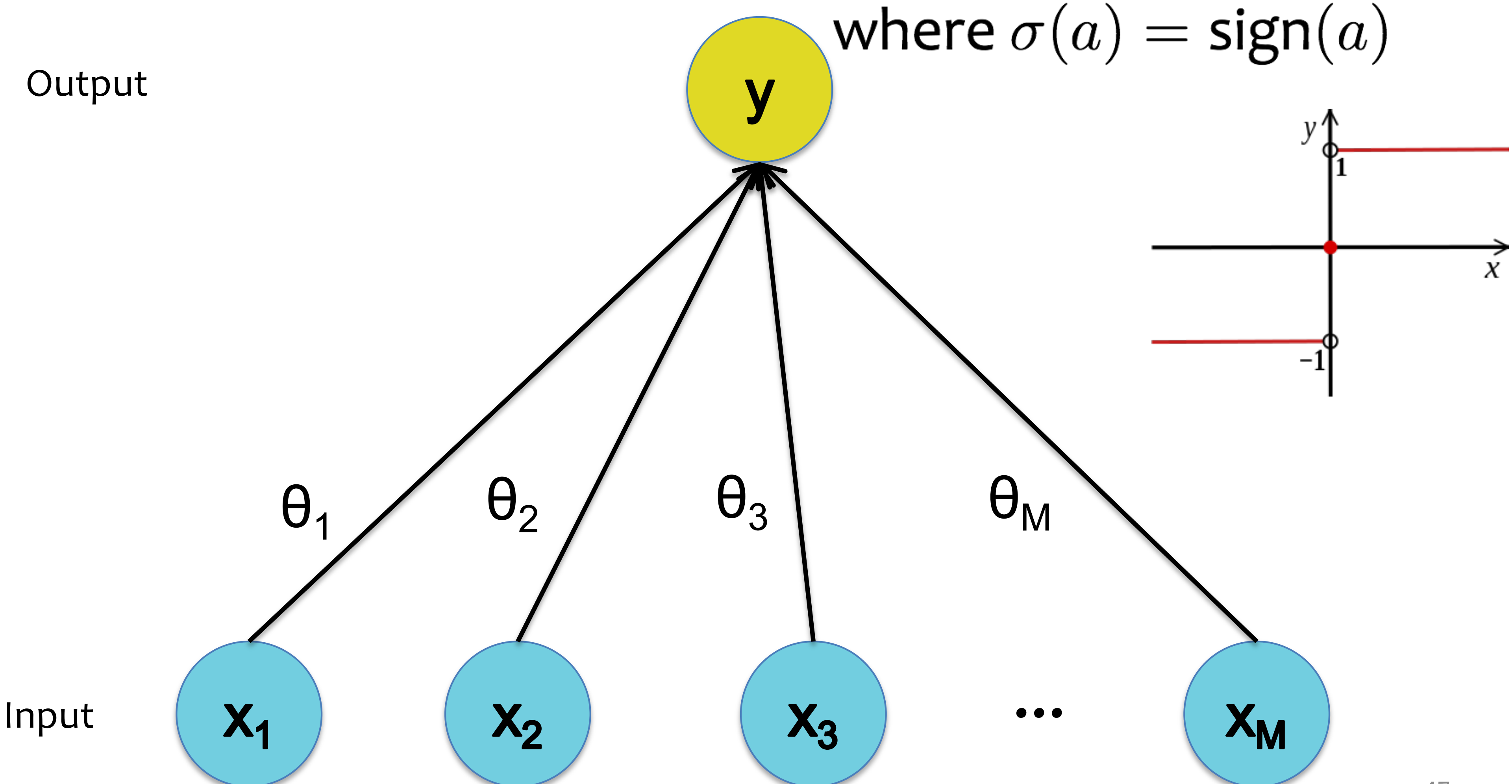


Decision Functions

Perceptron

$$y = h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sigma(\boldsymbol{\theta}^T \boldsymbol{x})$$

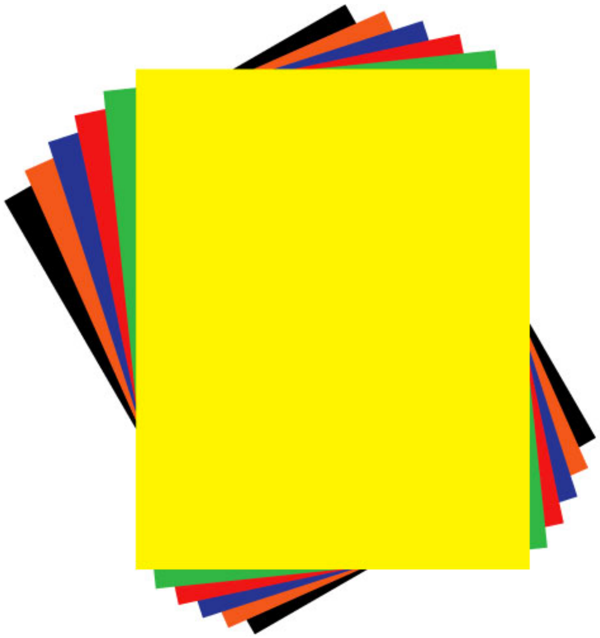
where $\sigma(a) = \text{sign}(a)$

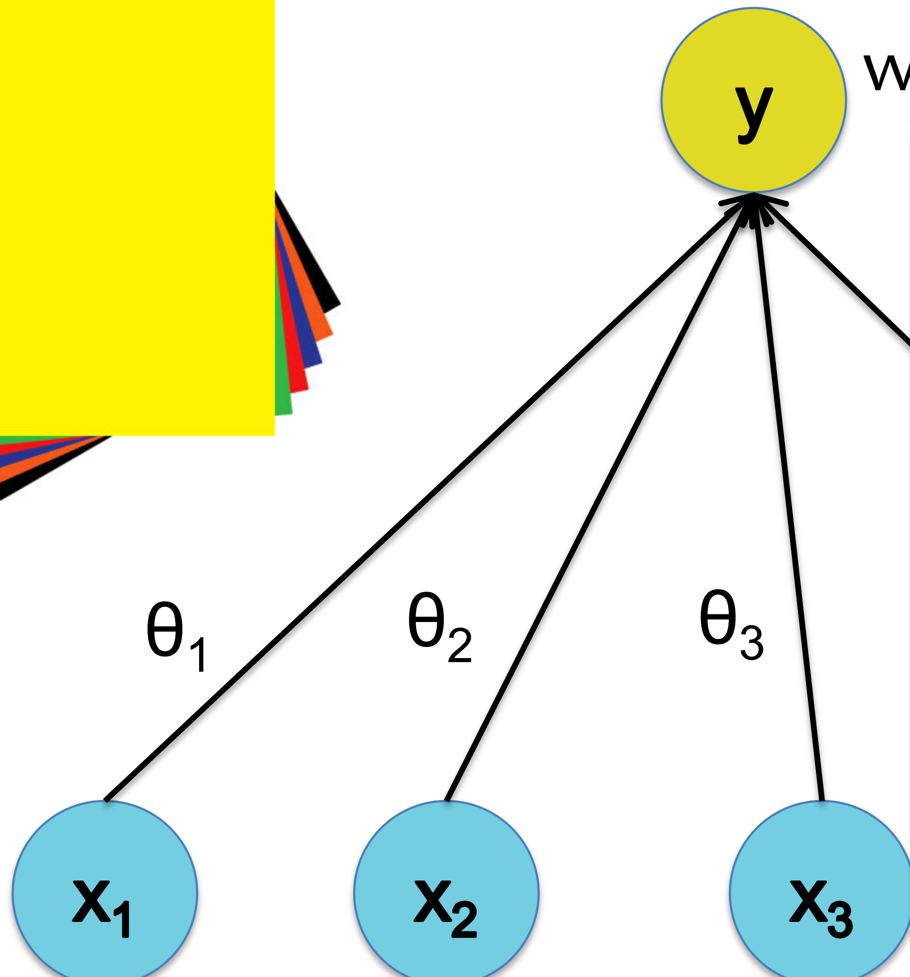


Decision Functions

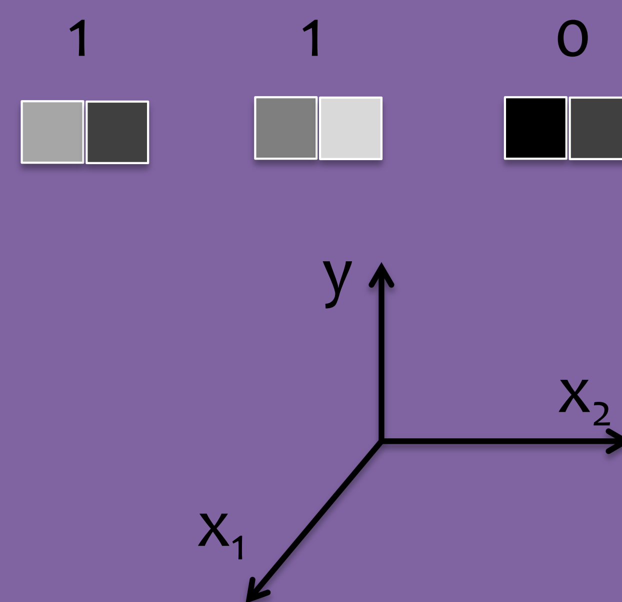
Logistic Regression

$$y = h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sigma(\boldsymbol{\theta}^T \boldsymbol{x})$$



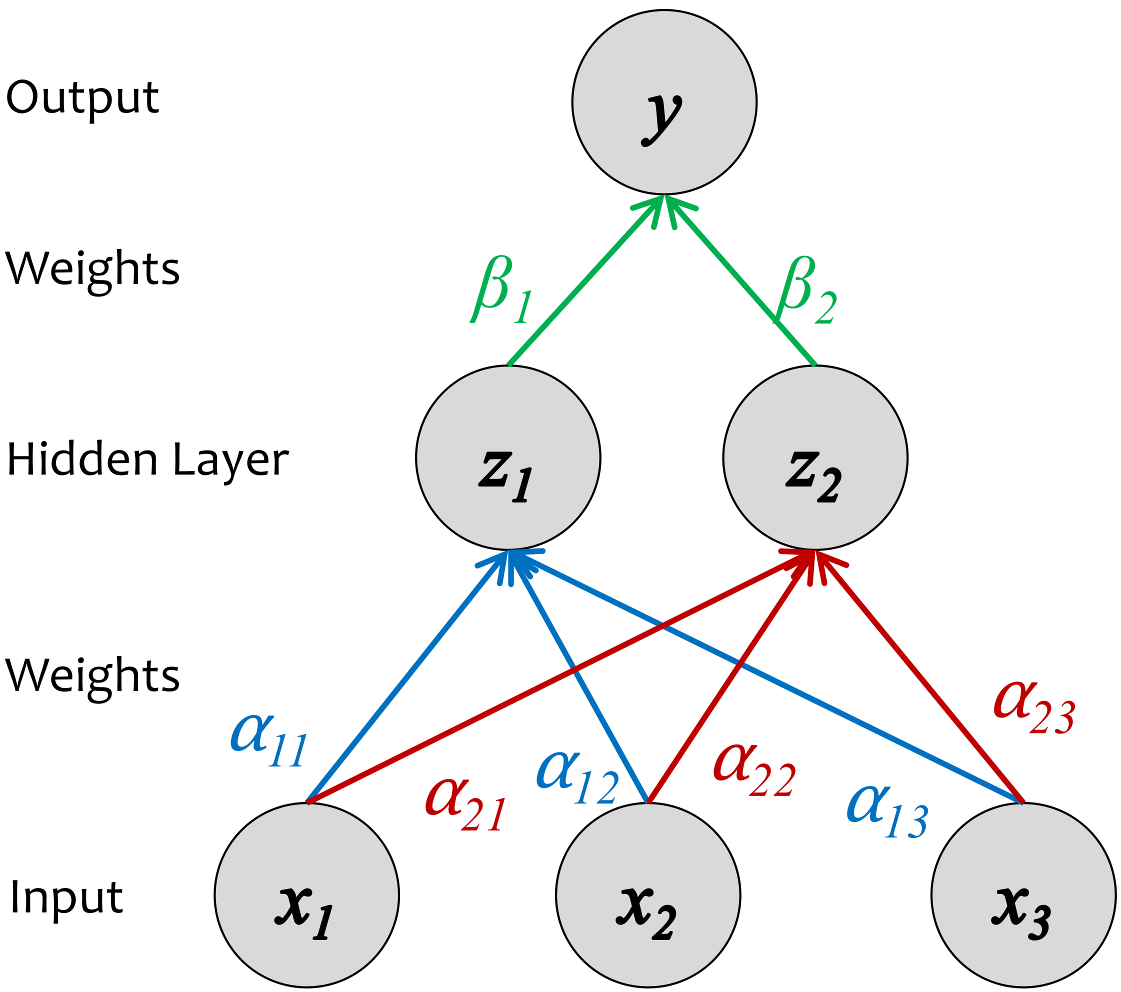


In-Class Example



Decision Functions

Neural Network

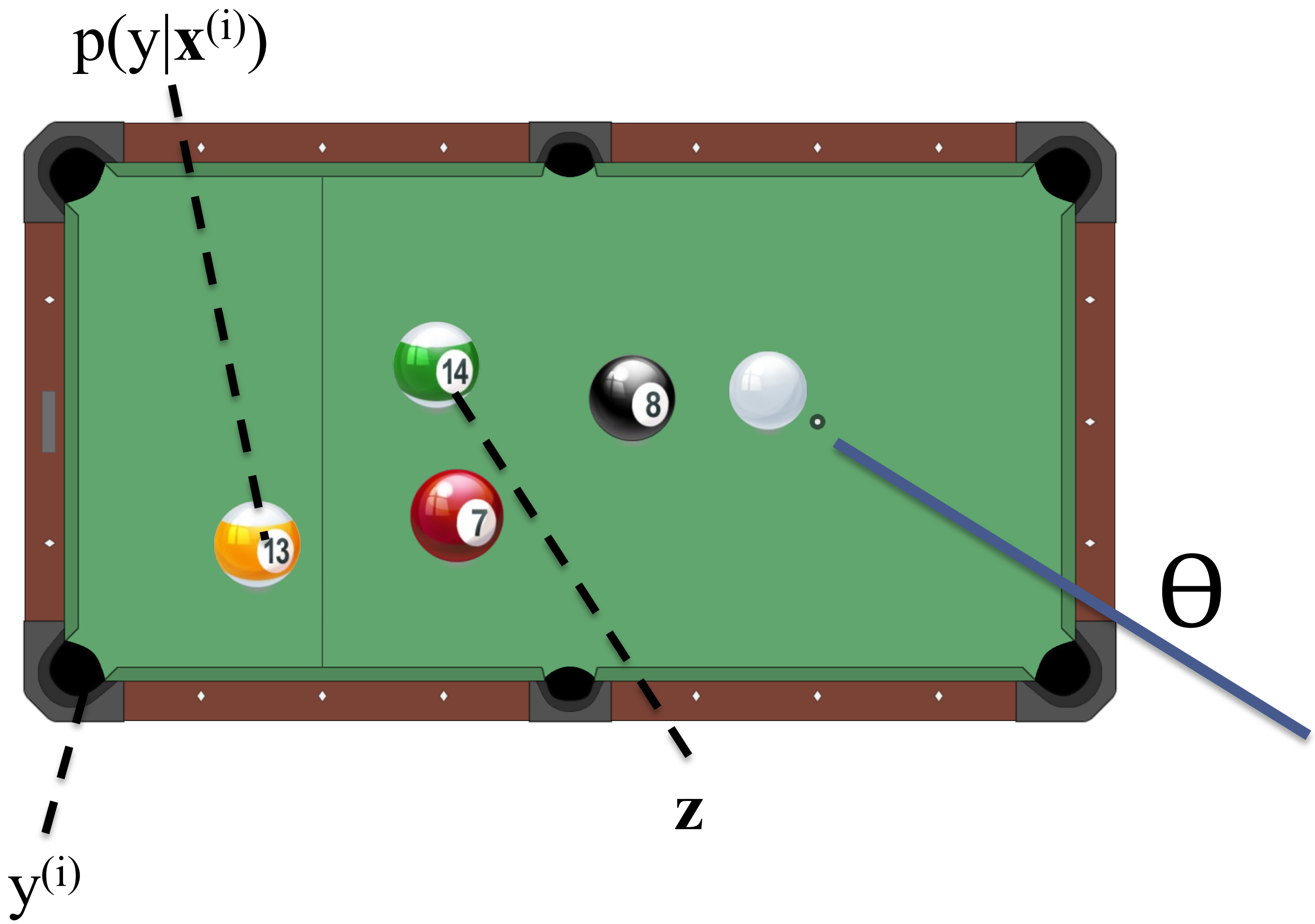


$$y = \sigma(\boldsymbol{\beta}^T \mathbf{z})$$

$$z_2 = \sigma(\boldsymbol{\alpha}_{2,\cdot}^T \mathbf{x})$$

$$z_1 = \sigma(\boldsymbol{\alpha}_{1,\cdot}^T \mathbf{x})$$

Error Back-Propagation



Slide from (Stoyanov & Eisner, 2012)

Training

Differentiation Quiz

Speed Quiz:
2 minute time limit.

Differentiation Quiz #1:

Suppose x = 2 and z = 3, what are dy/dx and dy/dz for the function below? **Round your answer to the nearest integer**.

$$y = \exp(xz) + \frac{xz}{\log(x)} + \frac{\sin(\log(x))}{xz}$$

Answer: *Answers below are in the fo*

A. [42, -72]
B. [72, -42]
C. [100, 127]
D. [127, 100]
E. [12
F. [81
G. [15
H. [94

```
from math import *

# Define function
def f(x, z):
    return exp(x*z) + x*z/log(x) + sin(log(x)) / (x*z)

# Inputs
x = 2; z = 3; e = 1e-8

# Finite difference check
dydx = (f(x+e, z) - f(x-e, z)) / (2*e)
dydz = (f(x, z+e) - f(x, z-e)) / (2*e)
print("dydx =", dydx)
print("dydz =", dydz)
```

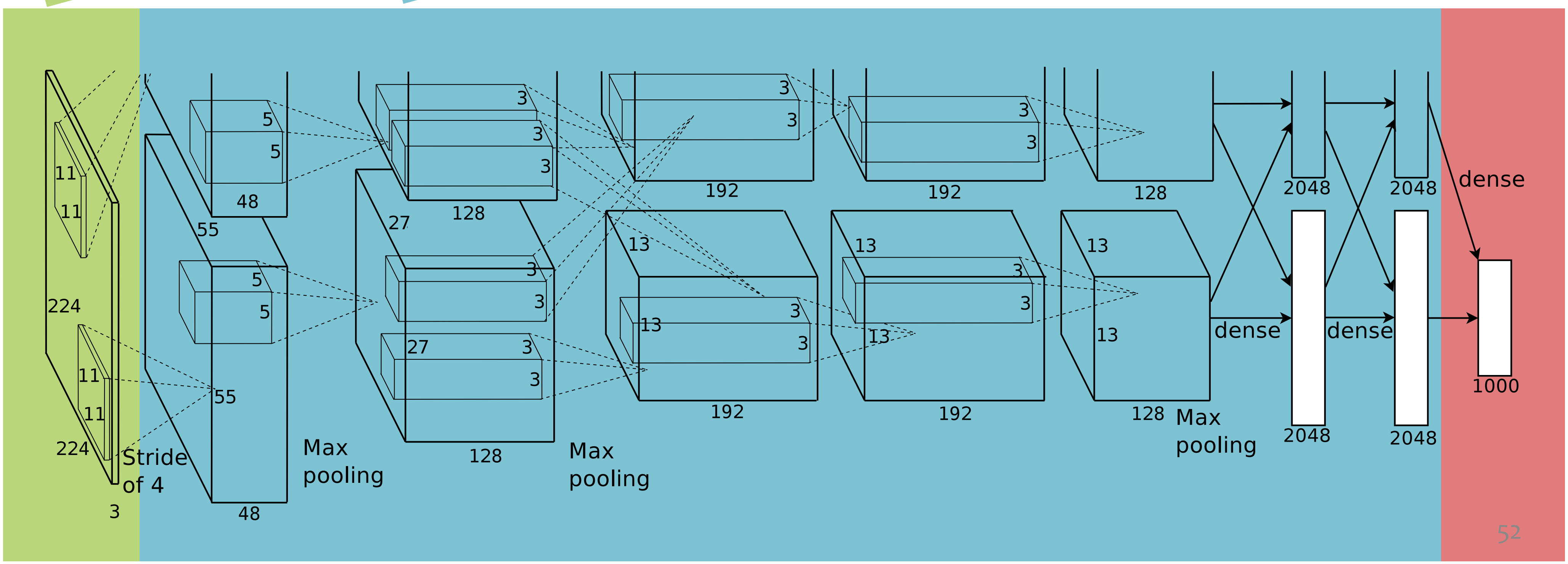
Architecture #2: AlexNet

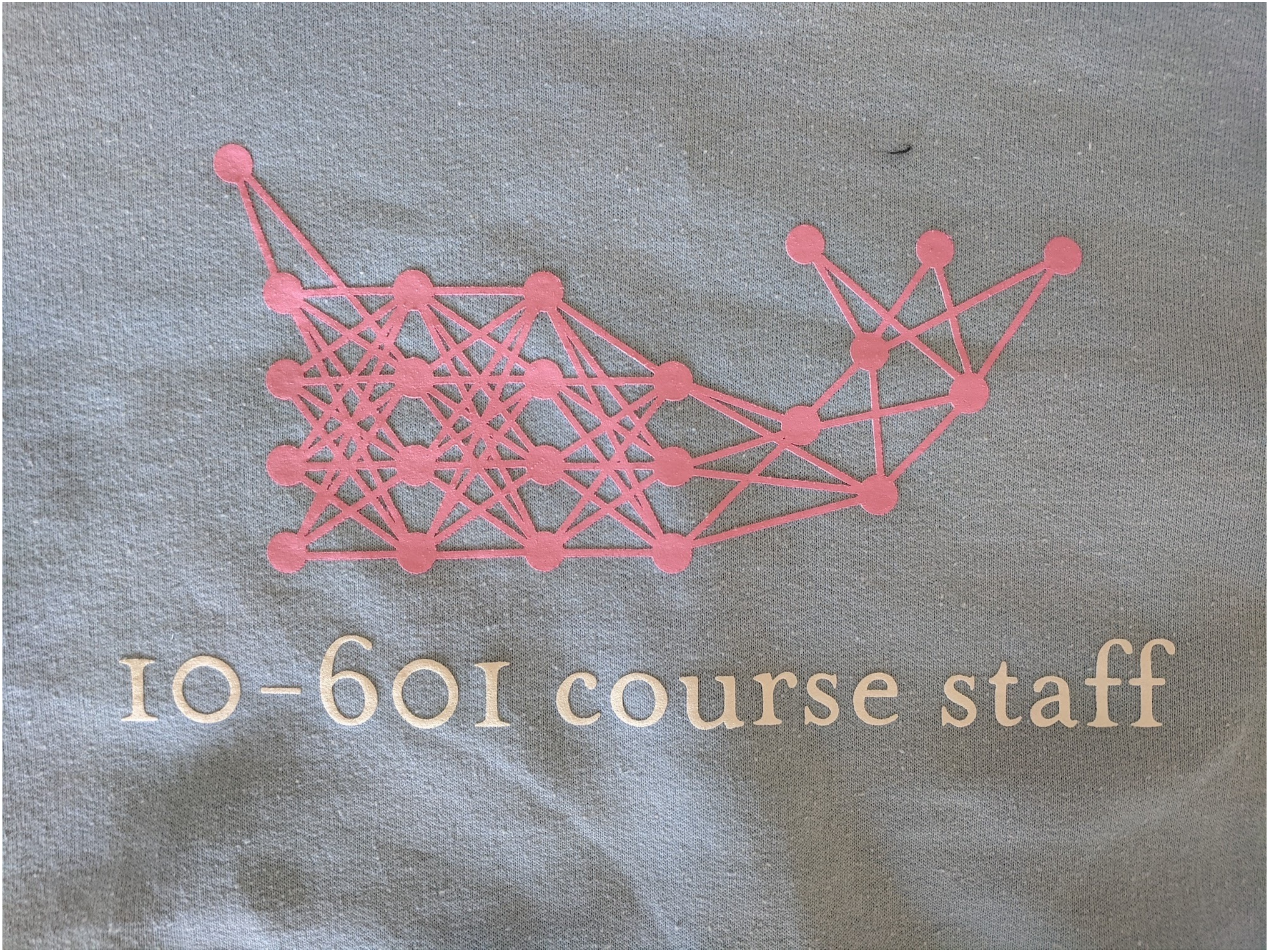
CNN for Image Classification
(Krizhevsky, Sutskever & Hinton, 2012)
15.3% error on ImageNet LSVRC-2012 contest

Input image (pixels)

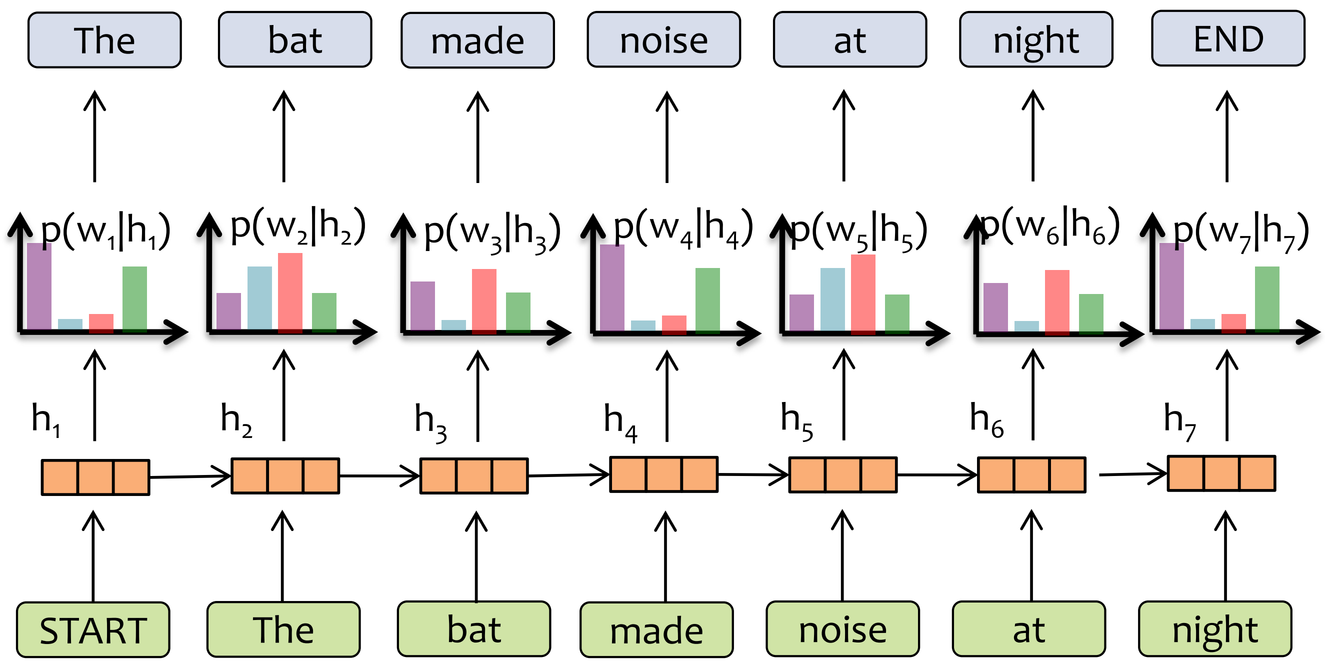
- Five convolutional layers (w/max-pooling)
- Three fully connected layers

1000-way softmax




10-601 course staff

RNN Language Model



Key Idea:
(1) convert all previous words to a **fixed length vector**
(2) define distribution $p(w_t \mid f_\theta(w_{t-1}, \ldots, w_1))$ that conditions on the vector $\mathbf{h}_t = f_\theta(w_{t-1}, \ldots, w_1)$

Sampling from an RNN-LM

??

VIOLA: Why, Salisbury must find his flesh and thought That which I am not aps, not a man and in fire, To show the reining of the raven and the wars To grace my hand reproach within, and not a fair are hand, That Caesar and my goodly father's world; When I was heaven of presence and our fleets, We spare with hours, but cut thy council I am great, Murdered a
master's ready there My powe
so much as hell: Some service i
bondman here, Would show hi

KING LEAR: O, if you were a feeble sight, the courtesy of your law, Your sight and several breath, will wear the gods With his heads, and my hands are wonder'd at the deeds, So drop upon your lordship's head, and your opinion Shall be against your honour.

??

CHARLES: Marry, do I, sir; and I came to acquaint you with a matter. I am given, sir, secretly to understand that your younger brother Orlando hath a disposition to come in disguised against me to try a fall. To-morrow, sir, I wrestle for my credit; and he that escapes me without some broken limb shall acquit him
is but young and tender; and,
uld be loath to foil him, as I
honour, if he come in:
ny love to you, I came hither
to acquaint you withal, that either you might stay him from his intendment or brook such disgrace well as he shall run into, in that it is a thing of his own search and altogether against my will.

TOUCHSTONE: For my part, I had rather bear with you than bear you; yet I should bear no cross if I did bear you, for I think you have no money in your purse.

Which is the real Shakespeare?!

Example from http://karpathy.github.io/2015/05/21/rnn-effectiveness/

PAC-MAN Learning

For some hypothesis $h \in \mathcal{H}$:

1. True Error

$$R(h)$$

2. Training Error

$$\hat{R}(h)$$

Question 2:
What is the expected number of PAC-MAN levels Matt will complete before a **Game-Over**?

A. 1-10
B. 11-20
C. 21-30

Sample Complexity Results

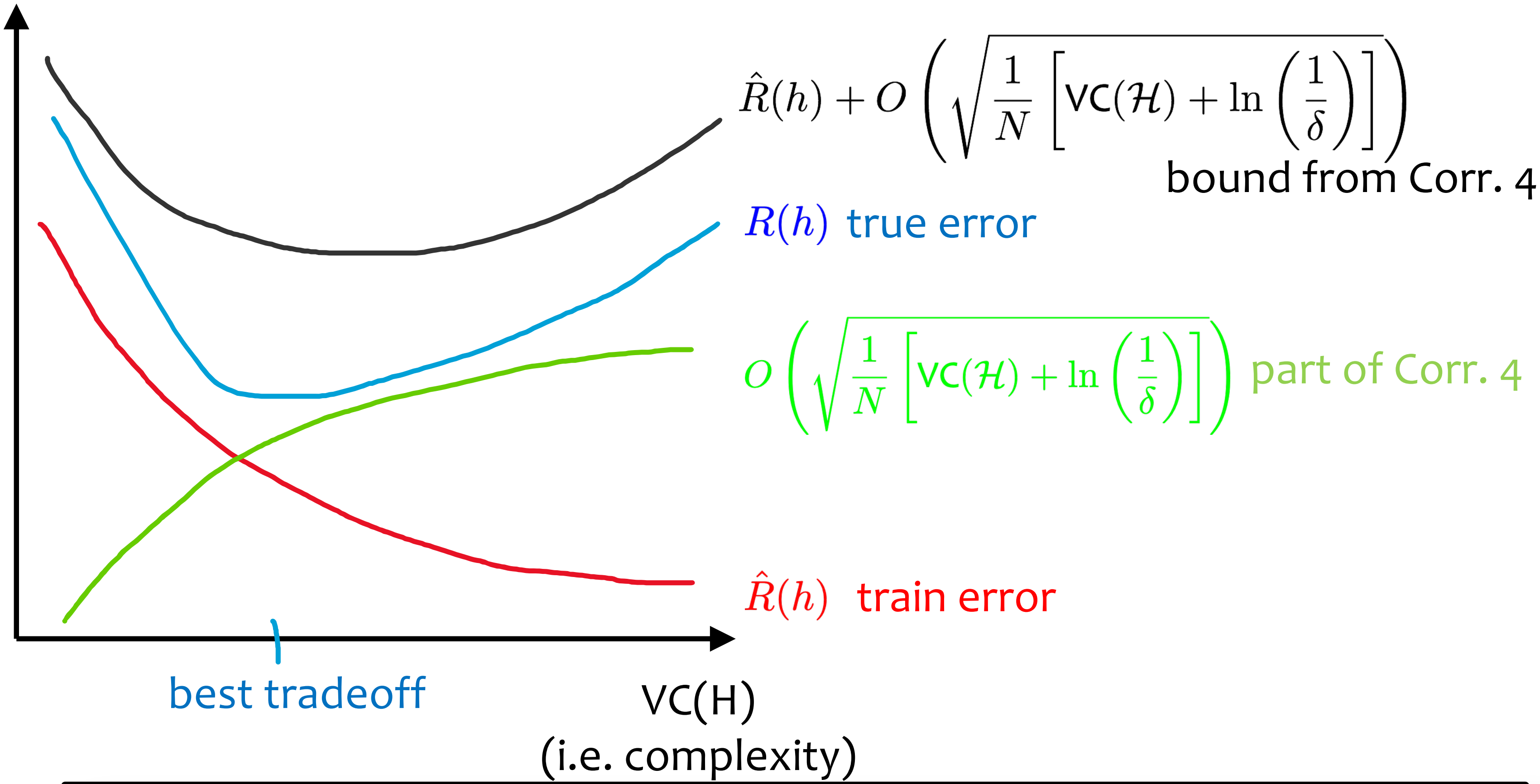
Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about...

	Realizable	Agnostic
Finite $\|\mathcal{H}\|$	**Thm. 1** $N \geq \frac{1}{\epsilon}\left[\log(\|\mathcal{H}\|) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.	**Thm. 2** $N \geq \frac{1}{2\epsilon^2}\left[\log(\|\mathcal{H}\|) + \log(\frac{2}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\|R(h) - \hat{R}(h)\| \leq \epsilon$.
Infinite $\|\mathcal{H}\|$	**Thm. 3** $N = O(\frac{1}{\epsilon}\left[\text{VC}(\mathcal{H})\log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta})\right])$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.	**Thm. 4** $N = O(\frac{1}{\epsilon^2}\left[\text{VC}(\mathcal{H}) + \log(\frac{1}{\delta})\right])$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\|R(h) - \hat{R}(h)\| \leq \epsilon$.

Learning Theory & Model Selection

error
(i.e. lower ➔
good data fit)



Key Point: we want to tradeoff between low training error and keeping H simple (low VC-Dim)

Q: Is Corollary 4 useful?
A: Yes!

Ex: H = Linear Separators in R^M

VC(H) = M+1

Q: In practice, how do we tradeoff between error and VC(H)?

A: Use a regularizer! That is, reducing the number of (effective) features reduces the VC dimension. More features usually leads to a better fit to the data.

Text Data


The TOOTH Book
POCKET ROCKS
HOP ON POP
By Dr. Seuss
The SIMPLEST SEUSS for YOUNGEST USE
THE CAT IN THE HAT
Touch, Feel
Go, Dog. Go!
by P.D. Eastman
by P.D Eastman
Chicka Chicka Boom Boom
by Bill Martin Jr
and John Archambault
illustrated by Lois Ehlert
The CRAYONS' Book of FEELINGS
Big Dog... Little Dog
by P.D. Eastman
幼儿启蒙大全
YOU ER QI MENG DA QUAN
RED FISH
BLUE FISH
dinosaur
Lois Ehlert

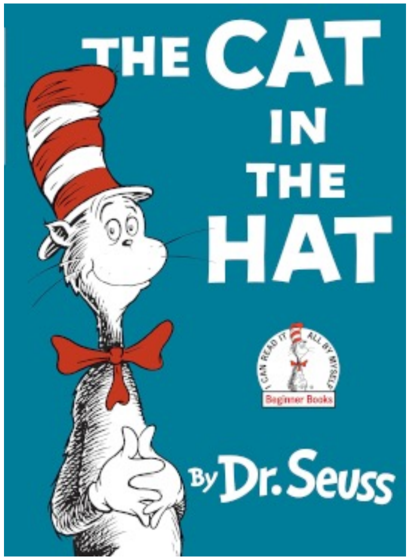
Bag-of-Words Model

x_1 ("hat")	x_2 ("cat")	x_3 ("dog")	x_4 ("fish")	x_5 ("mom")	x_6 ("dad")	y (Dr. Seuss)

Bag-of-Words Model

x_1 ("hat")	x_2 ("cat")	x_3 ("dog")	x_4 ("fish")	x_5 ("mom")	x_6 ("dad")	y (Dr. Seuss)
1	1	0	0	0	0	1

The Cat in the Hat
(by Dr. Seuss)

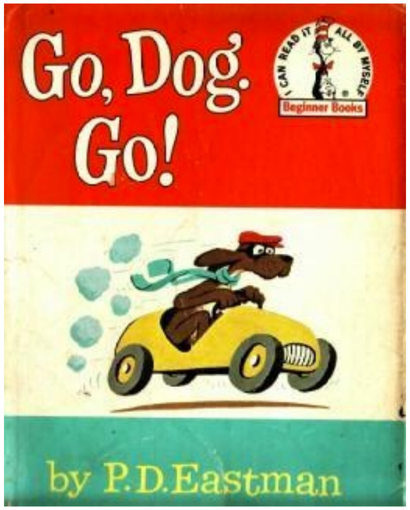


Source: https://en.wikipedia.org/wiki/The_Cat_in_the_Hat#/media/File:The_Cat_in_the_Hat.png

Bag-of-Words Model

x_1 ("hat")	x_2 ("cat")	x_3 ("dog")	x_4 ("fish")	x_5 ("mom")	x_6 ("dad")	y (Dr. Seuss)
1	1	0	0	0	0	1
0	0	1	0	0	0	0

Go, Dog. Go!
(by P. D. Eastman)

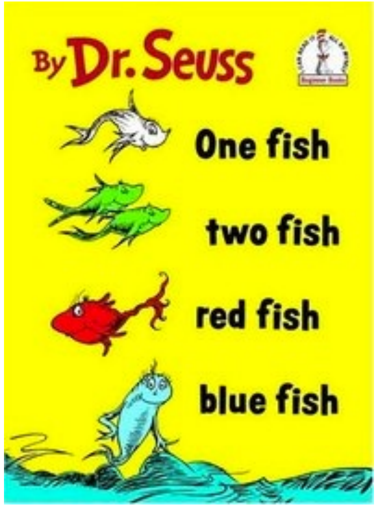


Source: https://en.wikipedia.org/wiki/Go,_Dog._Go!#/media/File:Go_Dog_Go.jpg

Bag-of-Words Model

x_1 ("hat")	x_2 ("cat")	x_3 ("dog")	x_4 ("fish")	x_5 ("mom")	x_6 ("dad")	y (Dr. Seuss)
1	1	0	0	0	0	1
0	0	1	0	0	0	0
0	0	0	1	0	0	1

One Fish, Two Fish, Red Fish, Blue Fish (by Dr. Seuss)

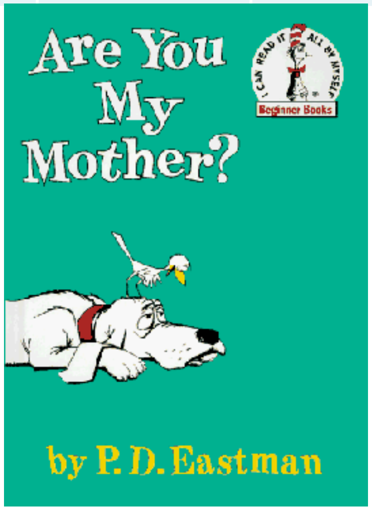


Source: https://en.wikipedia.org/wiki/One_Fish,_Two_Fish,_Red_Fish,_Blue_Fish#/media/File:One_Fish_Two_Fish_Red_Fish_Blue_Fish_(cover_art).jpg

Bag-of-Words Model

x_1 ("hat")	x_2 ("cat")	x_3 ("dog")	x_4 ("fish")	x_5 ("mom")	x_6 ("dad")	y (Dr. Seuss)
1	1	0	0	0	0	1
0	0	1	0	0	0	0
0	0	0	1	0	0	1
0	0	0	0	1	0	0

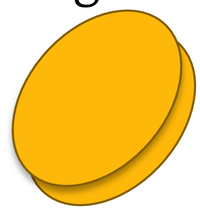
Are You My Mother?
(by P. D. Eastman)



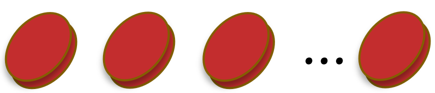
Source: https://en.wikipedia.org/wiki/Are_You_My_Mother%3F#/media/File:Areyoumymother.gif

Model 1: Bernoulli Naïve Bayes

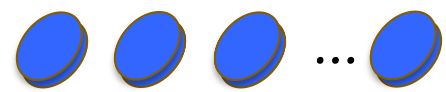
Flip weighted coin



If HEADS, flip
each red coin



If TAILS, flip
each blue coin



y	x_1	x_2	x_3	…	x_M
0	1	0	1	…	1
1	0	1	0	…	1
1	1	1	1	…	1
0	0	0	1	…	1
0	1	0	1	…	0
1	1	0	1	…	0

We can **generate** data in this fashion. Though in practice we never would since our data is **given**.

Instead, this provides an explanation of **how** the data was generated (albeit a terrible one).

Each red coin corresponds to an x_m

Recipe for Closed-form MLE

1. Assume data was generated i.i.d. from some model (i.e. write the generative story)

   $x^{(i)} \sim p(x|\boldsymbol{\theta})$

2. Write log-likelihood

   $\ell(\boldsymbol{\theta}) = \log p(x^{(1)}|\boldsymbol{\theta}) + \ldots + \log p(x^{(N)}|\boldsymbol{\theta})$

3. Compute partial derivatives (i.e. gradient)

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_1 = \ldots$

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_2 = \ldots$

   $\ldots$

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_M = \ldots$

4. Set derivatives to zero and solve for $\boldsymbol{\theta}$

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_m = 0$ for all $m \in \{1, \ldots, M\}$

   $\boldsymbol{\theta}^{MLE} =$ solution to system of M equations and M variables

5. Compute the second derivative and check that $\ell(\boldsymbol{\theta})$ is concave down at $\boldsymbol{\theta}^{MLE}$

Recipe for Closed-form MAP Estimation

1. Assume data was generated i.i.d. from some model (i.e. write the generative story)

   $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$ and then for all i: $x^{(i)} \sim p(x|\boldsymbol{\theta})$

2. Write log-likelihood

   $\ell_{MAP}(\boldsymbol{\theta}) = \log p(\boldsymbol{\theta}) + \log p(x^{(1)}|\boldsymbol{\theta}) + \ldots + \log p(x^{(N)}|\boldsymbol{\theta})$

3. Compute partial derivatives (i.e. gradient)

   $\partial \ell_{MAP}(\boldsymbol{\theta})/\partial \theta_1 = \ldots$

   $\partial \ell_{MAP}(\boldsymbol{\theta})/\partial \theta_2 = \ldots$

   …

   $\partial \ell_{MAP}(\boldsymbol{\theta})/\partial \theta_M = \ldots$

4. Set derivatives to zero and solve for $\boldsymbol{\theta}$

   $\partial \ell_{MAP}(\boldsymbol{\theta})/\partial \theta_m = 0$ for all $m \in \{1, \ldots, M\}$

   $\boldsymbol{\theta}^{MAP}$ = solution to system of M equations and M variables

5. Compute the second derivative and check that $\ell(\boldsymbol{\theta})$ is concave down at $\boldsymbol{\theta}^{MAP}$

Classification and Regression: The Big Picture

Recipe for Machine Learning

1. Given data $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$
2. (a) Choose a decision function $h_{\boldsymbol{\theta}}(\mathbf{x}) = \cdots$ (parameterized by $\boldsymbol{\theta}$)
   (b) Choose an objective function $J_{\mathcal{D}}(\boldsymbol{\theta}) = \cdots$ (relies on data)
3. Learn by choosing parameters that optimize the objective $J_{\mathcal{D}}(\boldsymbol{\theta})$

$$\hat{\boldsymbol{\theta}} \approx \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, J_{\mathcal{D}}(\boldsymbol{\theta})$$

4. Predict on new test example $\mathbf{x}_{\text{new}}$ using $h_{\boldsymbol{\theta}}(\cdot)$

$$\hat{y} = h_{\boldsymbol{\theta}}(\mathbf{x}_{\text{new}})$$

Decision Functions

- Perceptron: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \text{sign}(\boldsymbol{\theta}^T\mathbf{x})$
- Linear Regression: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T\mathbf{x}$
- Discriminative Models: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \underset{y}{\operatorname{argmax}}\, p_{\boldsymbol{\theta}}(y \mid \mathbf{x})$
  - Logistic Regression: $p_{\boldsymbol{\theta}}(y = 1 \mid \mathbf{x}) = \sigma(\boldsymbol{\theta}^T\mathbf{x})$
  - Neural Net (classification): $p_{\boldsymbol{\theta}}(y = 1 \mid \mathbf{x}) = \sigma((\mathbf{W}^{(2)})^T\sigma((\mathbf{W}^{(1)})^T\mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})$
- Generative Models: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \underset{y}{\operatorname{argmax}}\, p_{\boldsymbol{\theta}}(\mathbf{x}, y)$
  - Naive Bayes: $p_{\boldsymbol{\theta}}(\mathbf{x}, y) = p_{\boldsymbol{\theta}}(y) \prod_{m=1}^{M} p_{\boldsymbol{\theta}}(x_m \mid y)$

Optimization Method

- Gradient Descent: $\boldsymbol{\theta} \to \boldsymbol{\theta} - \gamma\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$
- SGD: $\boldsymbol{\theta} \to \boldsymbol{\theta} - \gamma\nabla_{\boldsymbol{\theta}} J^{(i)}(\boldsymbol{\theta})$
  for $i \sim \text{Uniform}(1, \ldots, N)$
  where $J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} J^{(i)}(\boldsymbol{\theta})$
- mini-batch SGD
- closed form
  1. compute partial derivatives
  2. set equal to zero and solve

Objective Function

- MLE: $J(\boldsymbol{\theta}) = -\sum_{i=1}^{N} \log p(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$
- MCLE: $J(\boldsymbol{\theta}) = -\sum_{i=1}^{N} \log p(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)})$
- L2 Regularized: $J'(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \lambda||\boldsymbol{\theta}||_2^2$
  (same as Gaussian prior $p(\boldsymbol{\theta})$ over parameters)
- L1 Regularized: $J'(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \lambda||\boldsymbol{\theta}||_1$
  (same as Laplace prior $p(\boldsymbol{\theta})$ over parameters)

MATERIAL COVERED ON EXAM 3

Totoro’s Tunnel


SQUIRREL
HILL SOUTH
Covode St
Nicholson St
Phillips Ave
Crombie St
Melvin St
Taylor Allderdice
High School
Tilbury Ave
English Ln
Beechwood Blvd
Mt Royal Rd
376
Alderson St
Forward Ave
Beechwood Blvd
Flemington St
Inez Way
Mirror St
Burchfield Ave
Inez Way
Burchfield Ave
376
Forward Ave
Ira Way
Anita Ave
376
Monitor St
Kennebec St
Monitor St
Summerset Dr
Nine Mile Run
Ebdy St
Fernwald Rd
Windsor St
Beechwood Blvd
Saline St
Shady Ave
Caton St
Murray Ave
Lilac St
Rosemoor St
Melbourne St
Rosemont Ln
Landview Rd
Map data ©2018 Google
US FOODS

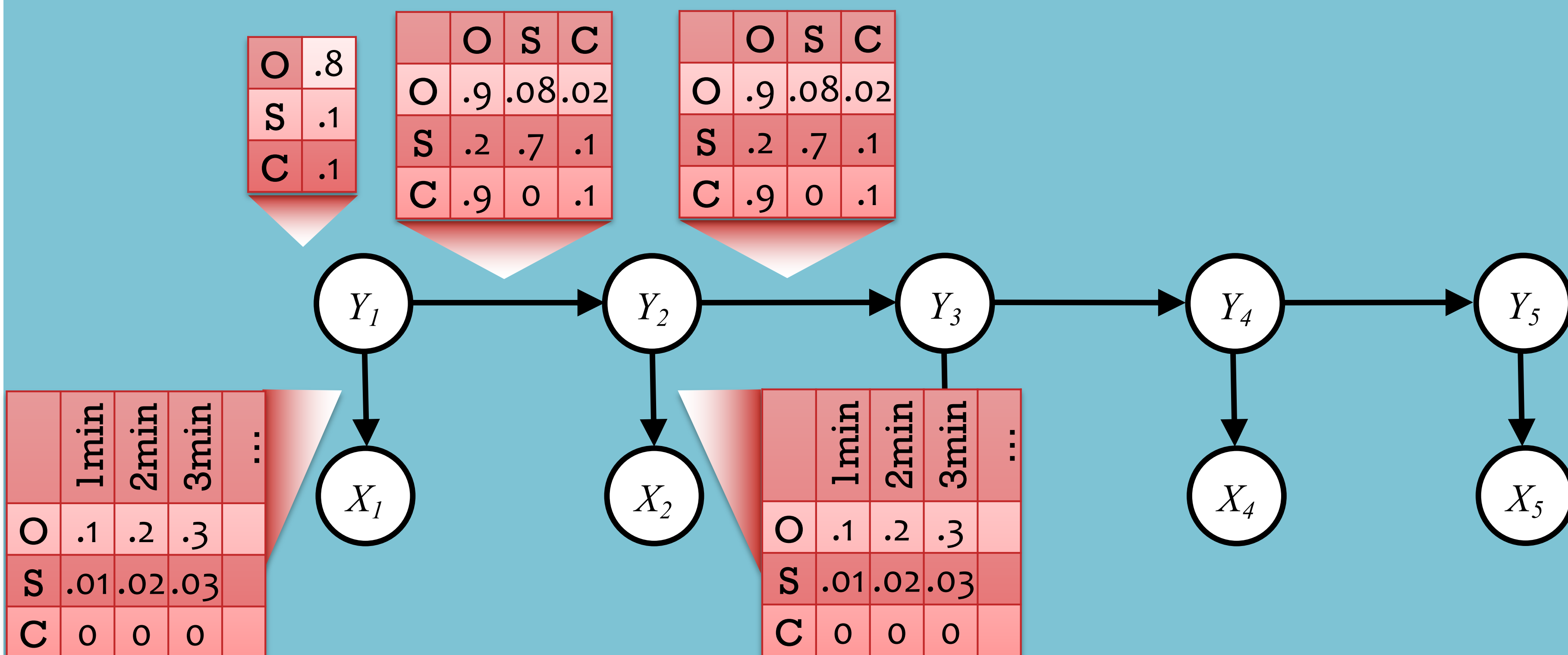
Hidden Markov Model

HMM Parameters:

Emission matrix, $\mathbf{A}$, where $P(X_t = k | Y_t = j) = A_{j,k}, \forall t, k$

Transition matrix, $\mathbf{B}$, where $P(Y_t = k | Y_{t-1} = j) = B_{j,k}, \forall t, k$

Initial probs, $\mathbf{C}$, where $P(Y_1 = k) = C_k, \forall k$



O	.8
S	.1
C	.1

	O	S	C
O	.9	.08	.02
S	.2	.7	.1
C	.9	0	.1

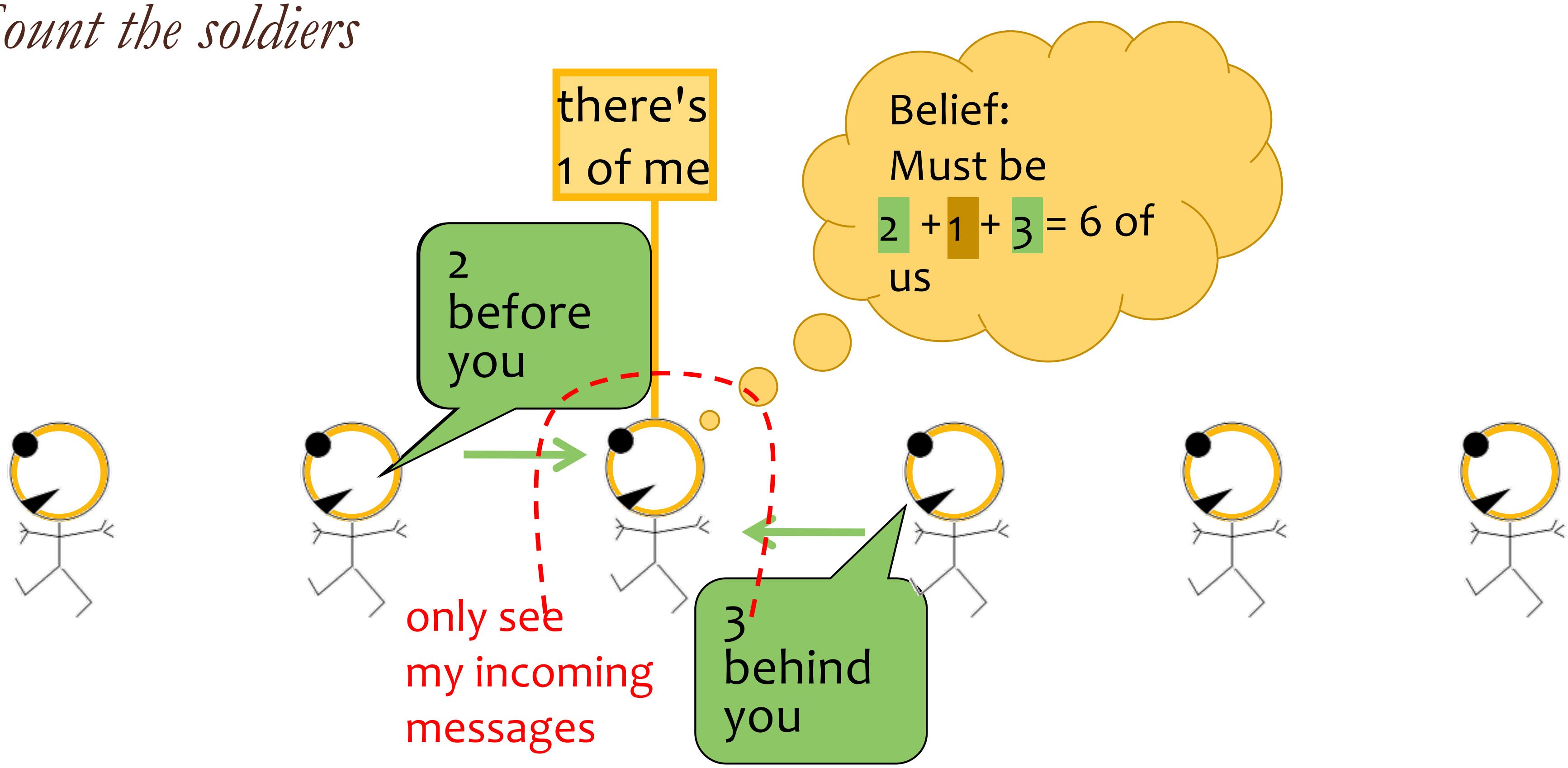
	O	S	C
O	.9	.08	.02
S	.2	.7	.1
C	.9	0	.1

	1min	2min	3min	…
O	.1	.2	.3	
S	.01	.02	.03	
C	0	0	0	

	1min	2min	3min	…
O	.1	.2	.3	
S	.01	.02	.03	
C	0	0	0	

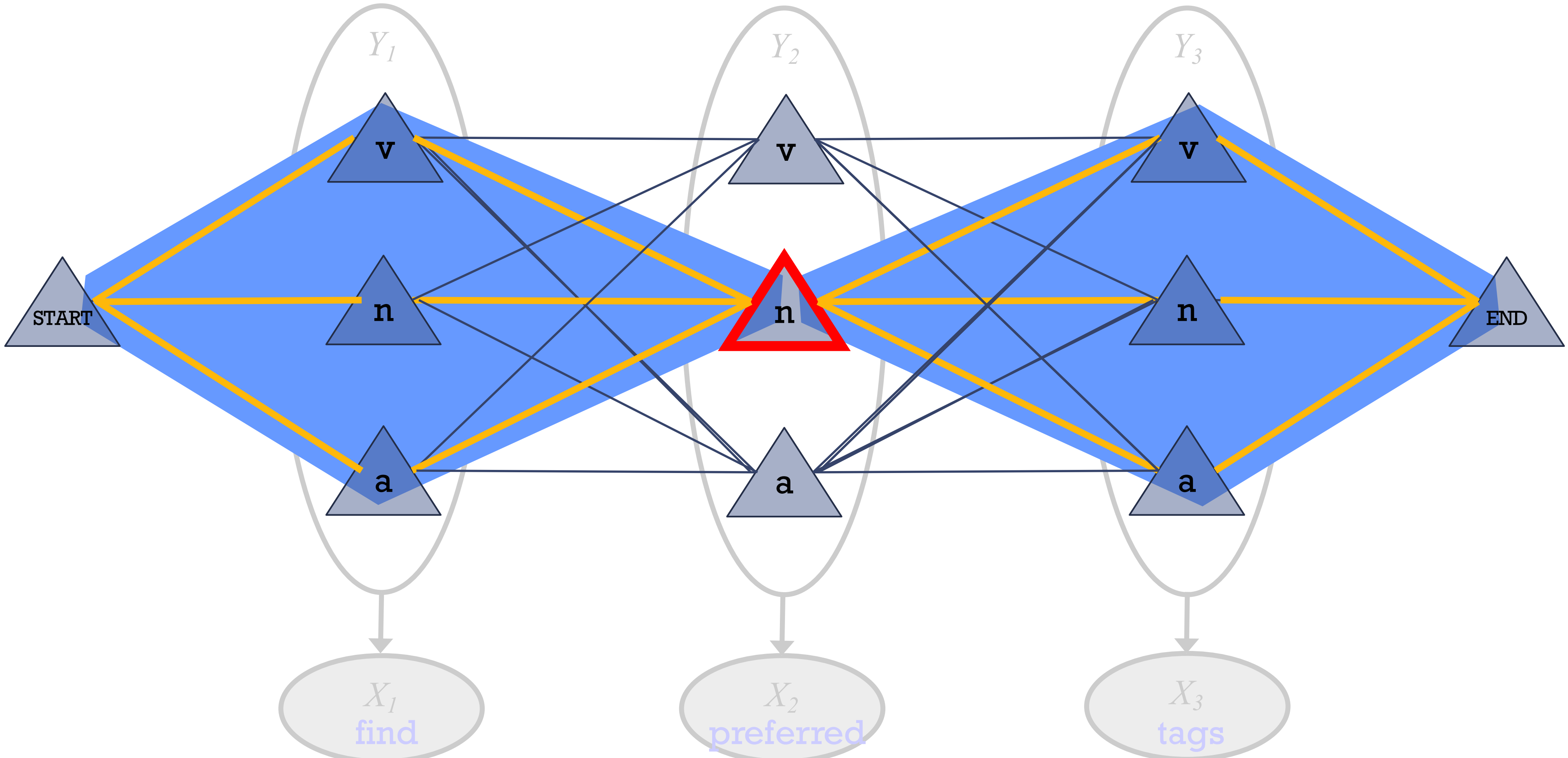
Great Ideas in ML: Message Passing

Count the soldiers



adapted from MacKay (2003) textbook

Forward-Backward Algorithm: Finds Marginals



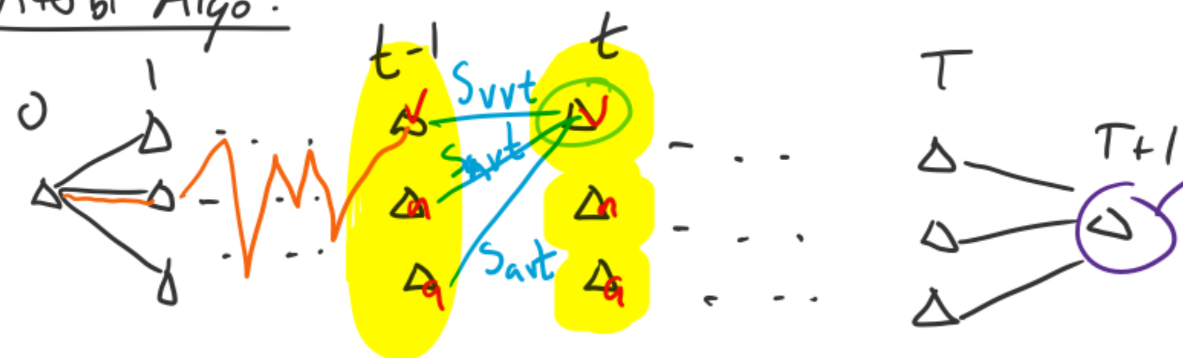
$\alpha_2(\mathbf{n})$ = total weight of these path *prefixes* (a + b + c)

$\beta_2(\mathbf{n})$ = total weight of these path *suffixes* (x + y + z)

Product gives ax+ay+az+bx+by+bz+cx+cy+cz = total weight of paths

Viterbi Algorithm

Viterbi Algo:



$W_T(\text{END})$ = weight of maximum weight path in trellis

$$w_t(v) = \max\big(w_{t-1}(v)\, s_{vvt},\; w_{t-1}(n)\, s_{nvt},\; w_{t-1}(a)\, s_{avt} \big)$$

for $t = 1, \dots, T$:

for $k = 1, \dots, K$:

$$w_t(k) = \max_{j \in \{1,\dots,K\}} w_{t-1}(j)\, s_{kjt}$$

$$b_t(k) = \operatorname*{argmax}_{j \in \{1,\dots,K\}} w_{t-1}(j)\, s_{kjt}$$

For HMM:

$$s_{kjt} = p(y_t = k \mid y_{t-1} = j)\; p(x_t \mid y_t = k)$$

backpointer: tag at timestep $t-1$ that yielded the max weight path into node (t,k)

Sample Questions

4 Hidden Markov Models

1. Given the POS tagging data shown, what are the parameter values learned by an HMM?

Verb	Noun	Verb
see	spot	run

Verb	Noun	Verb
run	spot	run

Adj.	Adj.	Noun
funny	funny	spot

Sample Questions

4 Hidden Markov Models

1. Given the POS tagging data shown, what are the parameter values learned by an HMM?

Verb	Noun	Verb
see	spot	run

Verb	Noun	Verb
run	spot	run

Adj.	Adj.	Noun
funny	funny	spot

2. Suppose you a learning an HMM POS Tagger, how many POS tag sequences of length 23 are there?

3. How does an HMM efficiently search for the most probable tag sequence given a 23-word sentence?

Example: CMU Mission Control

Bloomberg Subscribe

Businessweek | Technology

College Students Are About to Put a Robot on the Moon Before NASA

A commercial spaceflight in May will take a Carnegie Mellon-designed rover, named Iris, to the lunar surface.

An engineering model of the Iris rover at Carnegie Mellon University's Robotics Institute. *Source: Carnegie Mellon University*

By Katrina Manson
March 29, 2023 at 8:00 AM EDT

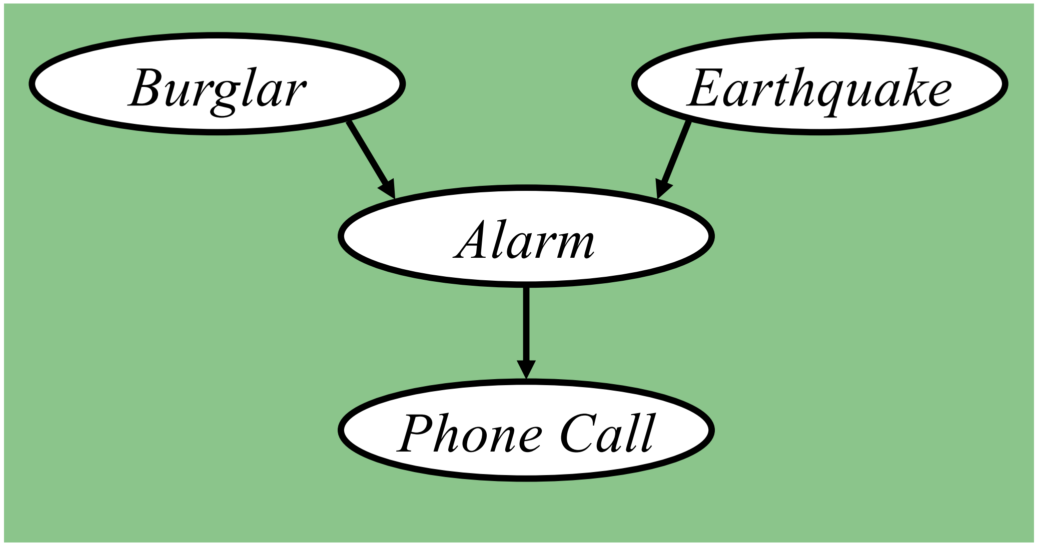
The "Burglar Alarm" example

- After you get this phone call, suppose you learn that there was a medium-sized earthquake in your neighborhood. Oh, whew! Probably not a burglar after all.
- Earthquake "explains away" the hypothetical burglar.
- But then it must **not** be the case that

$$Burglar \perp\!\!\!\perp Earthquake \mid PhoneCall$$

even though

$$Burglar \perp\!\!\!\perp Earthquake$$



Example: Tornado Alarms

Hacking Attack Woke Up Dallas With Emergency Sirens, Officials Say

By **ELI ROSENBERG** and **MAYA SALAM** APRIL 8, 2017

Warning sirens in Dallas, meant to alert the public to emergencies like severe weather, started sounding around 11:40 p.m. Friday, and were not shut off until 1:20 a.m. Rex C. Curry for The New York Times

1. Imagine that you work at the 911 call center in Dallas
2. You receive six calls informing you that the Emergency Weather Sirens are going off
3. What do you conclude?

Figure from https://www.nytimes.com/2017/04/08/us/dallas-emergency-sirens-hacking.html

Sample Questions

(a) [2 pts.] Write the expression for the joint distribution.

5 Graphical Models [16 pts.]

We use the following Bayesian network to model the relationship between studying (S), being well-rested (R), doing well on the exam (E), and getting an A grade (A). All nodes are binary, i.e., $R, S, E, A \in \{0, 1\}$.

$$
\begin{tikzcd}
S \arrow[dr] & & R \arrow[dl] \\
& E \arrow[d] & \\
& A &
\end{tikzcd}
$$

Figure 5: Directed graphical model for problem 5.

Sample Questions

(b) [2 pts.] How many parameters are necessary to describe the joint distribution?

5 Graphical Models [16 pts.]

We use the following Bayesian network to model the relationship between studying (S), being well-rested (R), doing well on the exam (E), and getting an A grade (A). All nodes are binary, i.e., $R, S, E, A \in \{0, 1\}$.

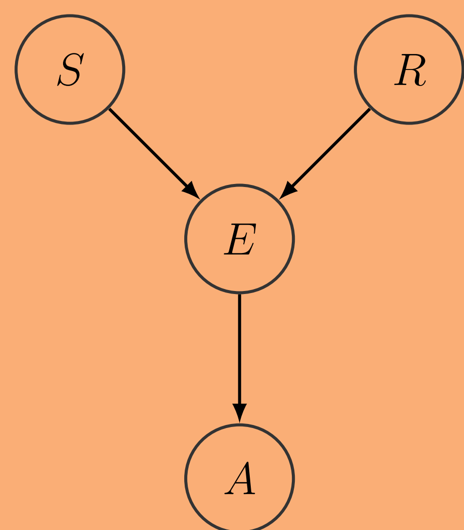


Figure 5: Directed graphical model for problem 5.

Sample Questions

Poll

(d) [2 pts.] Is S marginally independent of R? Is S conditionally independent of R given E? Answer yes or no to each questions and provide a brief explanation why.

5 Graphical Models [16 pts.]

We use the following Bayesian network to model the relationship between studying (S), being well-rested (R), doing well on the exam (E), and getting an A grade (A). All nodes are binary, i.e., $R, S, E, A \in \{0, 1\}$.

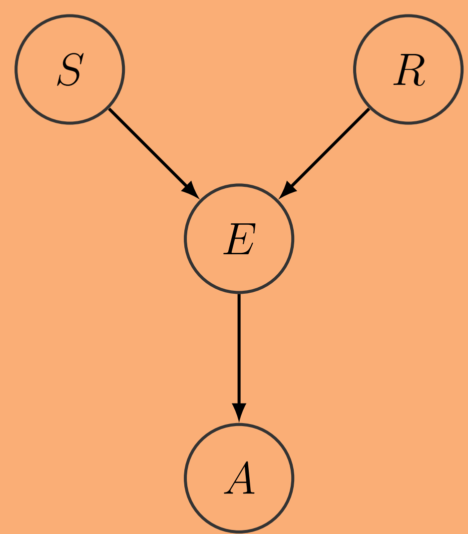


Figure 5: Directed graphical model for problem 5.

Sample Questions

5 Graphical Models

(f) [3 pts.] Give two reasons why the graphical models formalism is convenient when compared to learning a full joint distribution.

A Few Problems for Bayes Nets

Suppose we already have the parameters of a Bayesian Network…

1. How do we compute the probability of a specific assignment to the variables?
   $P(T=t, H=h, A=a, C=c)$

2. How do we draw a sample from the joint distribution?
   $t,h,a,c \sim P(T, H, A, C)$

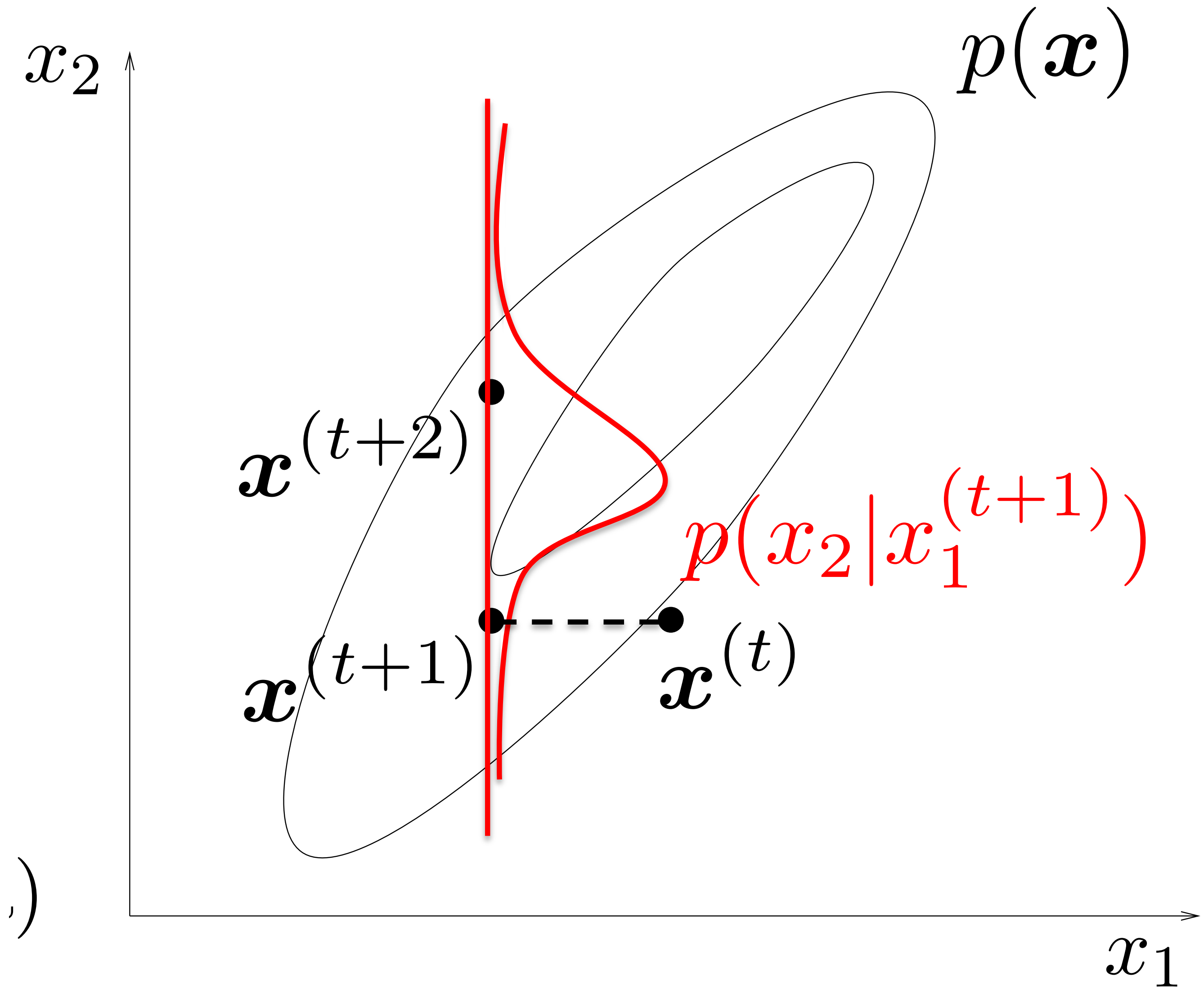
3. How do we compute marginal probabilities?
   $P(A) = \ldots$

4. How do we draw samples from a conditional distribution?
   $t,h,a \sim P(T, H, A \mid C = c)$

5. How do we compute conditional marginal probabilities?
   $P(H \mid C = c) = \ldots$

Can we use samples ?

Gibbs Sampling



RL: Components

From the Environment (i.e. the MDP)

- State space, $\mathcal{S}$
- Action space, $\mathcal{A}$
- Reward function, $R(s, a),\ R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
- Transition probabilities, $p(s' \mid s, a)$
  - Deterministic transitions:

$$p(s' \mid s, a) = \begin{cases} 1 \text{ if } \delta(s, a) = s' \\ 0 \text{ otherwise} \end{cases}$$

where $\delta(s, a)$ is a transition function

Markov Assumption

$$p(s_{t+1} \mid s_t, a_t, \ldots, s_1, a_1) = p(s_{t+1} \mid s_t, a_t)$$

From the Model

- Policy, $\pi : \mathcal{S} \to \mathcal{A}$
- Value function, $V^{\pi} : \mathcal{S} \to \mathbb{R}$
  - Measures the expected total payoff of starting in some state s and *executing* policy π

MDP Example: Multi-armed bandit

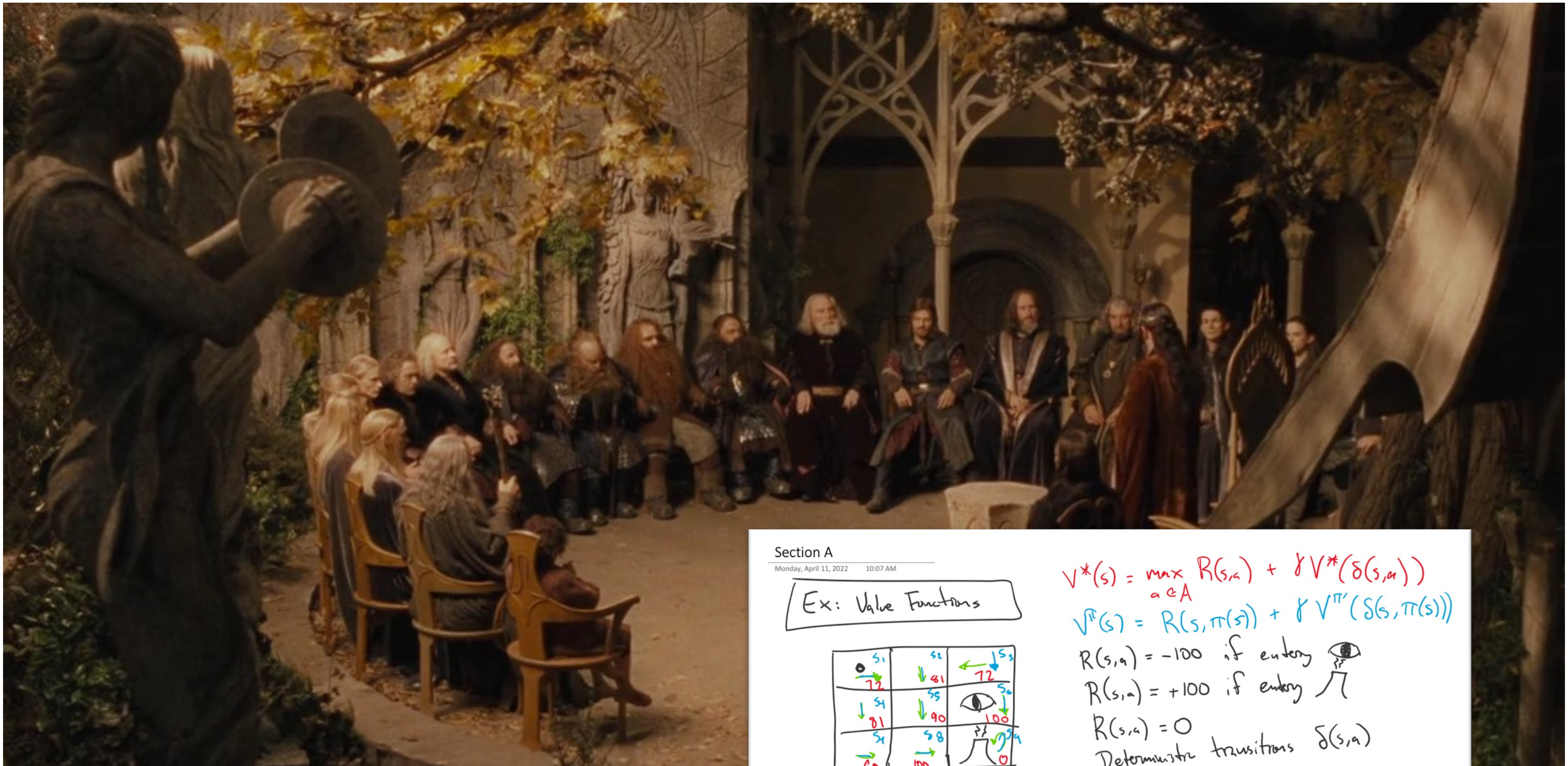
- Single state:
  $$|\mathcal{S}| = 1$$
- Three actions:
  $$\mathcal{A} = \{1, 2, 3\}$$
- Rewards are stochastic

Example: Path Planning

Section A

Monday, April 11, 2022 10:07 AM

Ex: Value Functions



$$V^*(s) = \max_{a \in A} R(s,a) + \gamma V^*(\delta(s,a))$$

$$V^{\pi}(s) = R(s,\pi(s)) + \gamma V^{\pi'}(\delta(s,\pi(s)))$$

$R(s,a) = -100$ if entering

$R(s,a) = +100$ if entering

$R(s,a) = 0$

Deterministic transitions $\delta(s,a)$

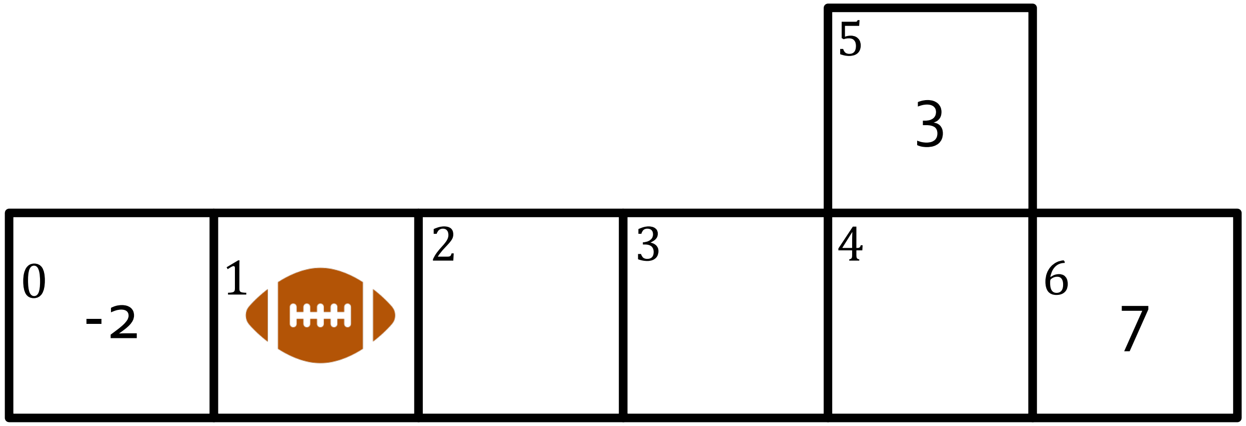
V^* values, Assume $\gamma = 0.9$

← an optimal policy, π^*

← another policy, π'

$$V^{\pi'}(s_3) = -100 + 0.9(100) = -10$$

RL: Value Function Example



$$R(s,a) = \begin{cases} -2 \text{ if entering state 0 (safety)} \\ 3 \text{ if entering state 5 (field goal)} \\ 7 \text{ if entering state 6 (touch down)} \\ 0 \text{ otherwise} \end{cases}$$

$\gamma = 0.9$

Learning $Q^*(s,a)$

- Algorithm 3: ϵ-greedy online learning of Q^* (table form)
  - Inputs: discount factor γ,
    an initial state s,
    greediness parameter $\epsilon \in [0,1]$,
    learning rate $\alpha \in [0,1]$ ("mistrust parameter")
  - Initialize $Q(s,a) = 0 \ \forall\, s \in \mathcal{S}, a \in \mathcal{A}$
    (Q is a $|\mathcal{S}| \times |\mathcal{A}|$ table or array)
  - While TRUE, do
    - With probability $1-\epsilon$, take the greedy action $a = \underset{a' \in \mathcal{A}}{\operatorname{argmax}}\ Q(s,a')$. Otherwise (with probability ϵ), take a random action a
    - Receive reward $r = R(s,a)$
    - Observe the new state $s' \sim p(S' \mid s, a)$
    - Update Q and s

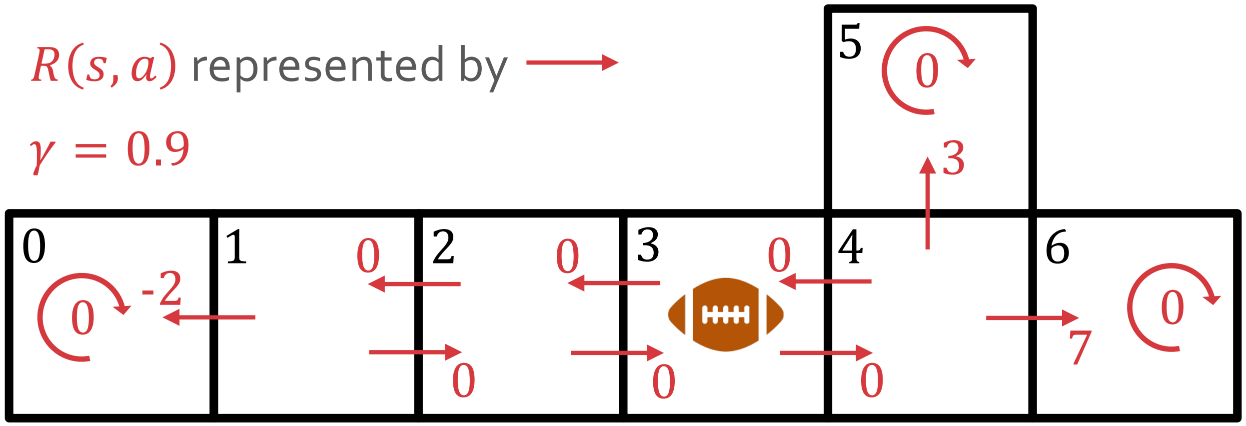
$$Q(s,a) \leftarrow (1-\alpha)\underbrace{Q(s,a)}_{\text{Current value}} + \alpha\underbrace{\left(r + \gamma \max_{a'} Q(s',a')\right)}_{\text{Update w/ deterministic transitions}}$$

$$s \leftarrow s'$$

Learning $Q^*(s, a)$: Example

$R(s, a)$ represented by $\longrightarrow$

$\gamma = 0.9$



$$Q(3, \rightarrow) \leftarrow 0 + (0.9) \max_{a' \in \{\rightarrow, \leftarrow, \uparrow, \circlearrowright\}} Q(4, a') = 2.7$$

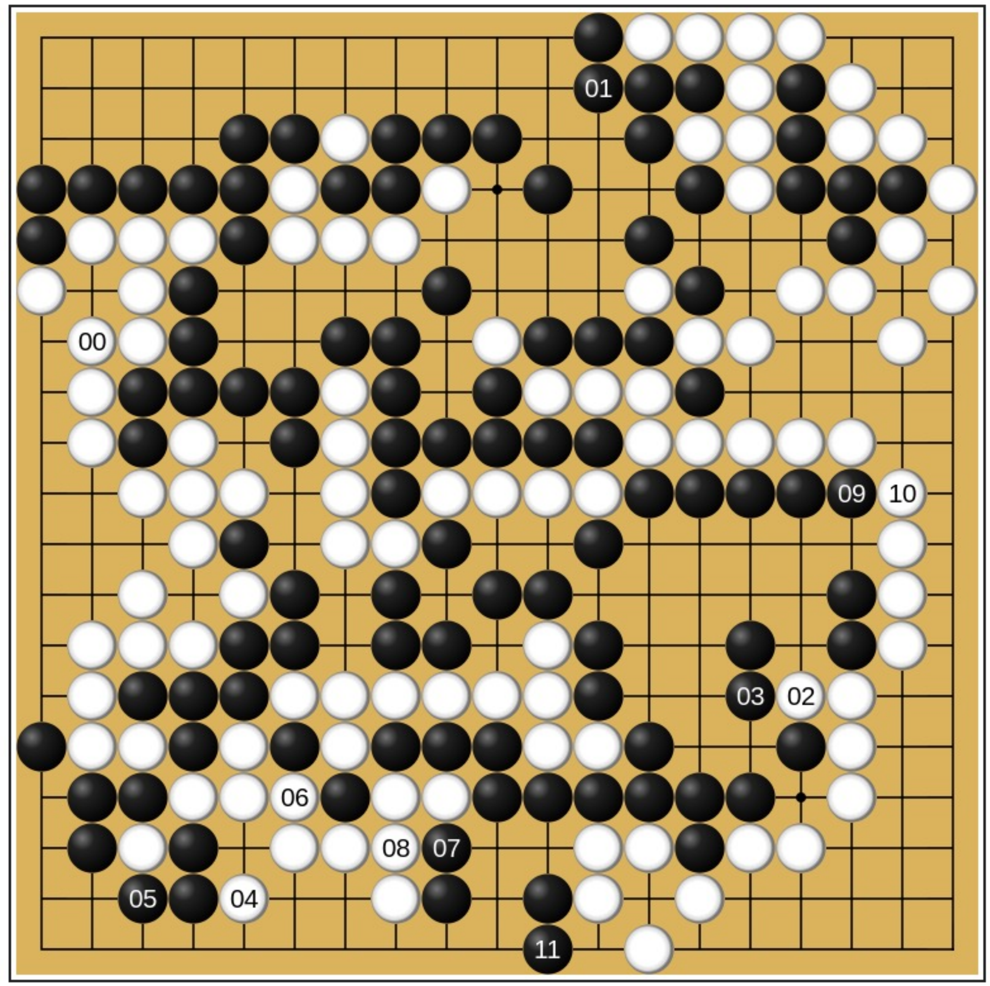
$Q(s,a)$	$\rightarrow$	$\leftarrow$	$\uparrow$	$\circlearrowright$
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	2.7	0	0	0
4	0	0	3	0
5	0	0	0	0
6	0	0	0	0

Alpha Go

Game of Go (圍棋)

- 19x19 **board**
- Players alternately play black/white **stones**
- **Goal** is to fully encircle the largest region on the board
- **Simple** rules, but **extremely complex** game play

AlphaGo (Black) vs. Lee Sedol (White) - Game 2
Final position (AlphaGo wins in 211 moves)



Source: https://en.wikipedia.org/wiki/AlphaGo_versus_Lee_Sedol

Deep Q-Learning

Question: *What if our state space S is too large to represent with a table?*

Examples:

- s_t = pixels of a video game
- s_t = continuous values of a sensors in a manufacturing robot
- s_t = sensor output from a self-driving car

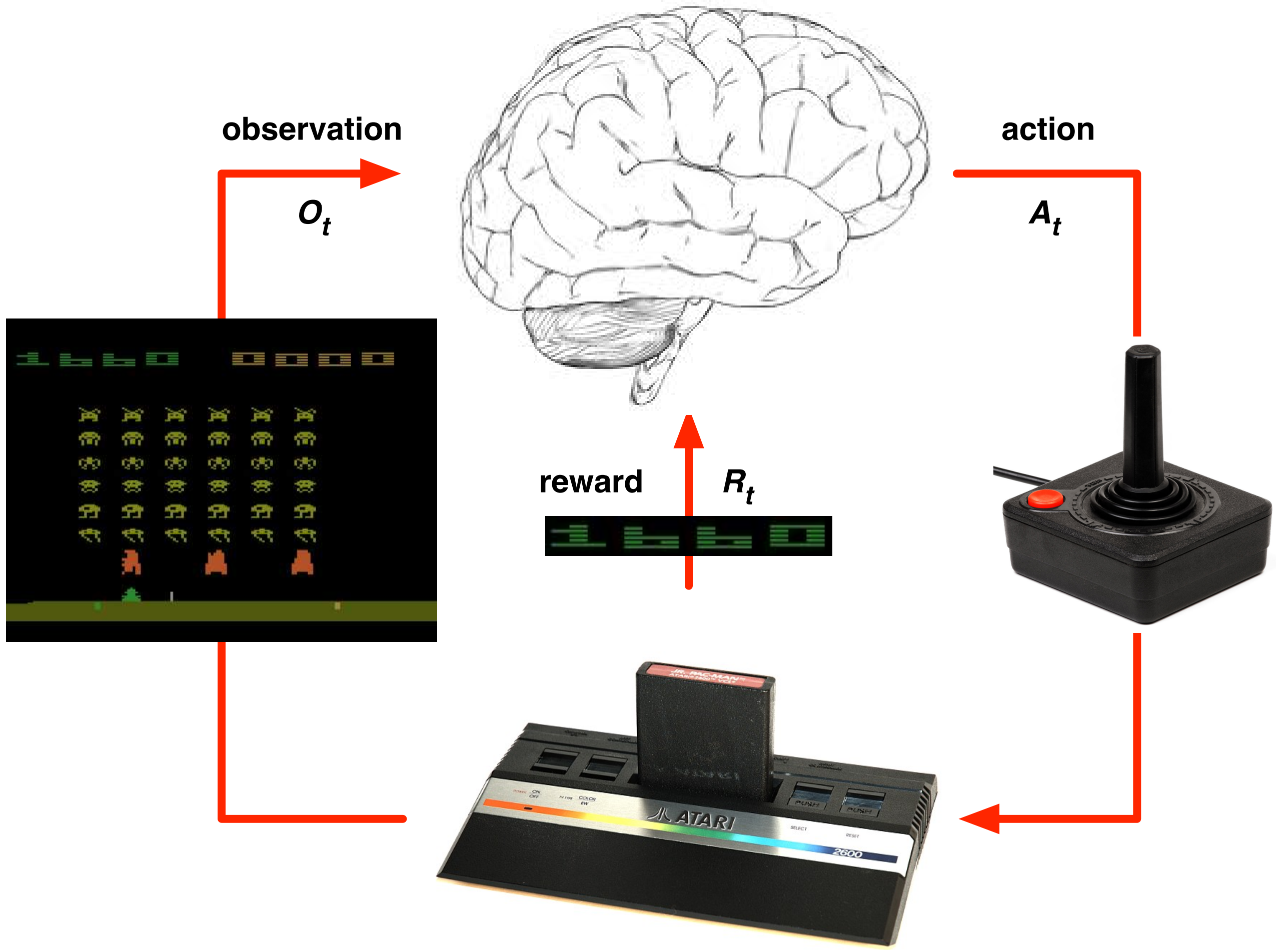
Answer: Use a parametric function to approximate the table entries

Key Idea:

1. Use a neural network $Q(s,a; \theta)$ to approximate $Q^*(s,a)$
2. Learn the parameters θ via SGD with training examples $< s_t, a_t, r_t, s_{t+1} >$

Playing Atari with Deep RL

- Setup: RL system observes the pixels on the screen
- It receives rewards as the game score
- Actions decide how to move the joystick / buttons



Figures from David Silver (Intro RL lecture)

Sample Questions

Poll

7.1 Reinforcement Learning

3. (1 point) **Please select one statement that is true for reinforcement learning and supervised learning.**

   ◯ Reinforcement learning is a kind of supervised learning problem because you can treat the reward and next state as the label and each state, action pair as the training data.

   ◯ Reinforcement learning differs from supervised learning because it has a temporal structure in the learning process, whereas, in supervised learning, the prediction of a data point does not affect the data you would see in the future.

Sample Questions

Poll

7.1 Reinforcement Learning

3. (1 point) **Please select one statement that is true for reinforcement learning and supervised learning.**

   ◯ Reinforcement learning is a kind of supervised learning problem because you can treat the reward and next state as the label and each state, action pair as the training data.

   ◯ Reinforcement learning differs from supervised learning because it has a temporal structure in the learning process, whereas, in supervised learning, the prediction of a data point does not affect the data you would see in the future.

4. (1 point) **True or False:** Value iteration is better at balancing exploration and exploitation compared with policy iteration.

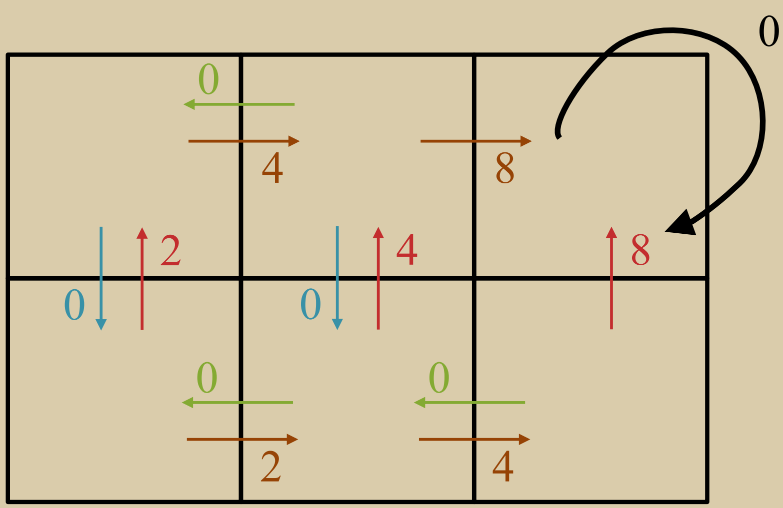
   ◯ True

   ◯ False

Sample Questions

7.1 Reinforcement Learning

1. For the R(s,a) values shown on the arrows below, what is the corresponding optimal policy? Assume the discount factor is 0.1



2. For the R(s,a) values shown on the arrows below, which are the corresponding V*(s) values? Assume the discount factor is 0.1

3. For the R(s,a) values shown on the arrows below, which are the corresponding Q*(s,a) values? Assume the discount factor is 0.1

4. Could we change R(s,a) such that all the V*(s) values change but the optimal policy stays the same? If so, show how and if not, briefly explain why not.

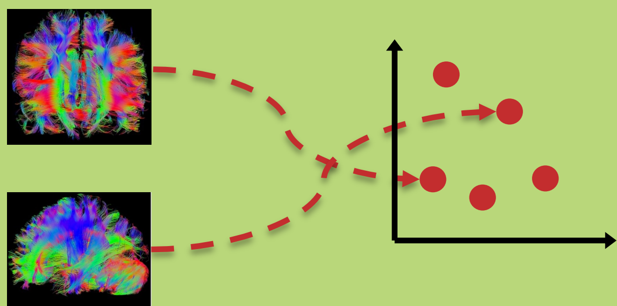
Shortcut Example

https://www.youtube.com/watch?v=MlJN9pEfPfE

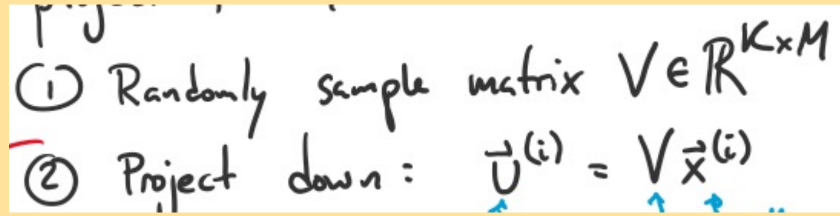
Photo from https://www.springcarnival.org/booth.shtml

PCA section in one slide…

1. Dimensionality reduction:



2. Random Projection:



3. Definition of PCA:

Choose the matrix V that either…

1. minimizes reconstruction error
2. consists of the K eigenvectors with largest eigenvalue

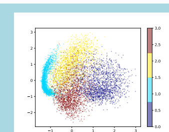
The above are equivalent definitions.

4. Algorithm for PCA:

The option we'll focus on:

Run Singular Value Decomposition (SVD) to obtain all the eigenvectors. Keep just the top-K to form V. Play some tricks to keep things efficient.

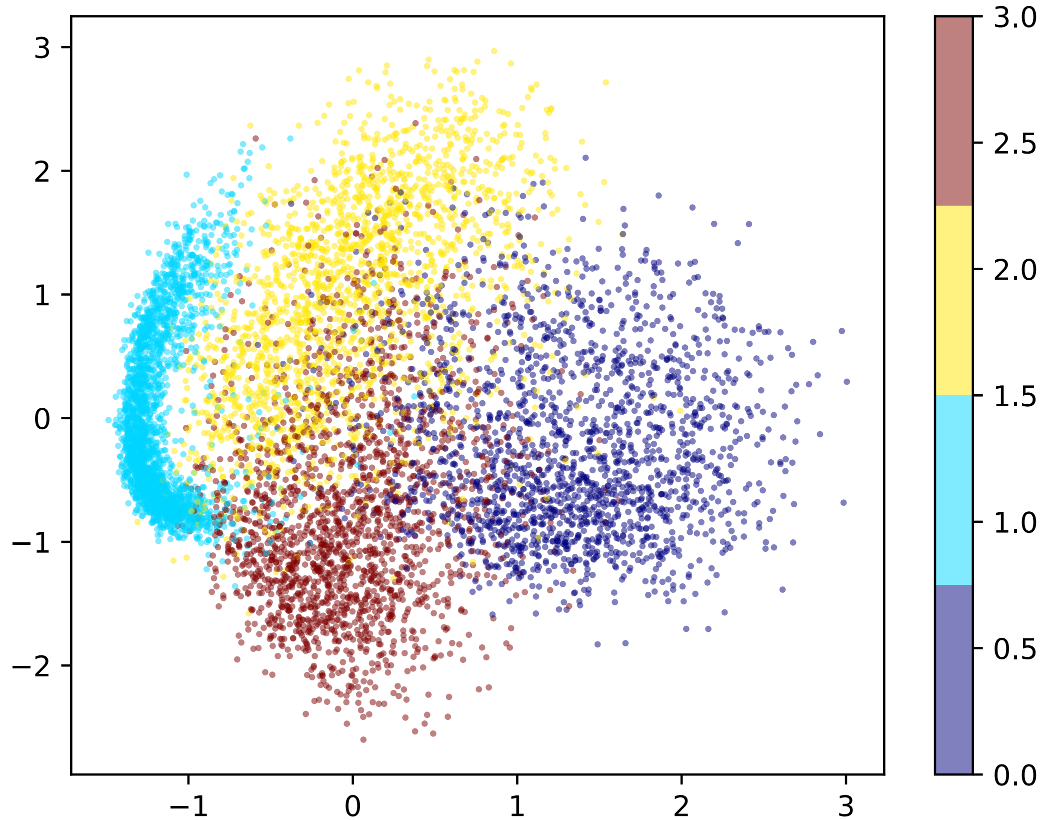
5. An Example



Projecting MNIST digits

Task Setting:

1. Take 25x25 images of digits and project them down to 2 components
2. Plot the 2 dimensional points



Sample Questions

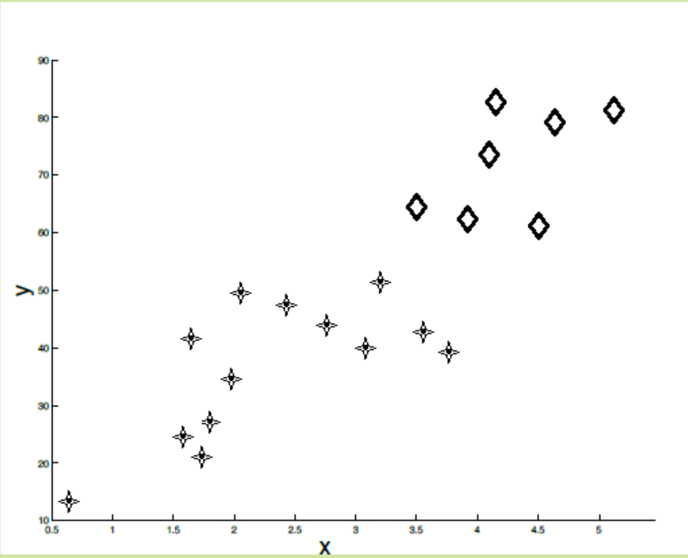
4 Principal Component Analysis [16 pts.]

(a) In the following plots, a train set of data points X belonging to two classes on $\mathbb{R}^2$ are given, where the original features are the coordinates (x, y). For each, answer the following questions:

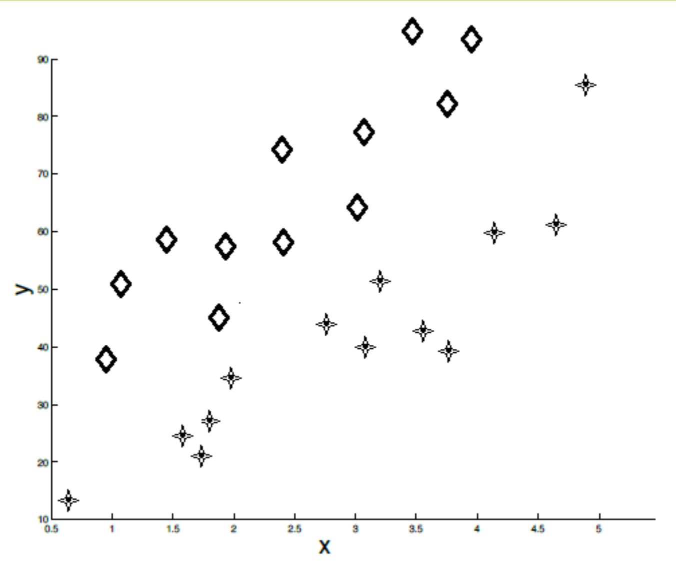
   (i) [3 pt.] Draw all the principal components.

   (ii) [6 pts.] Can we correctly classify this dataset by using a threshold function after projecting onto one of the principal components? If so, which principal component should we project onto? If not, explain in 1–2 sentences why it is not possible.

Dataset 1:

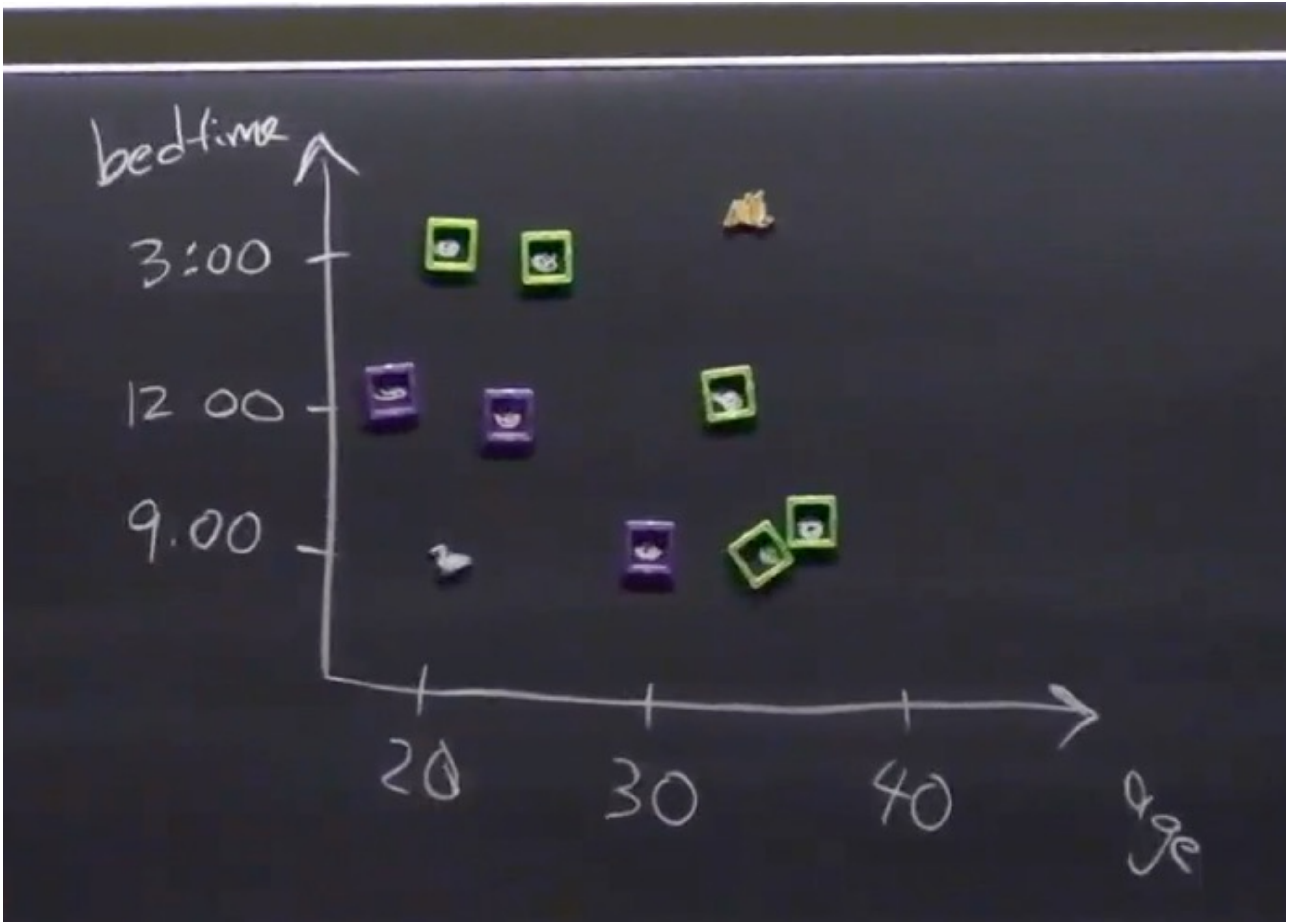


Dataset 2:

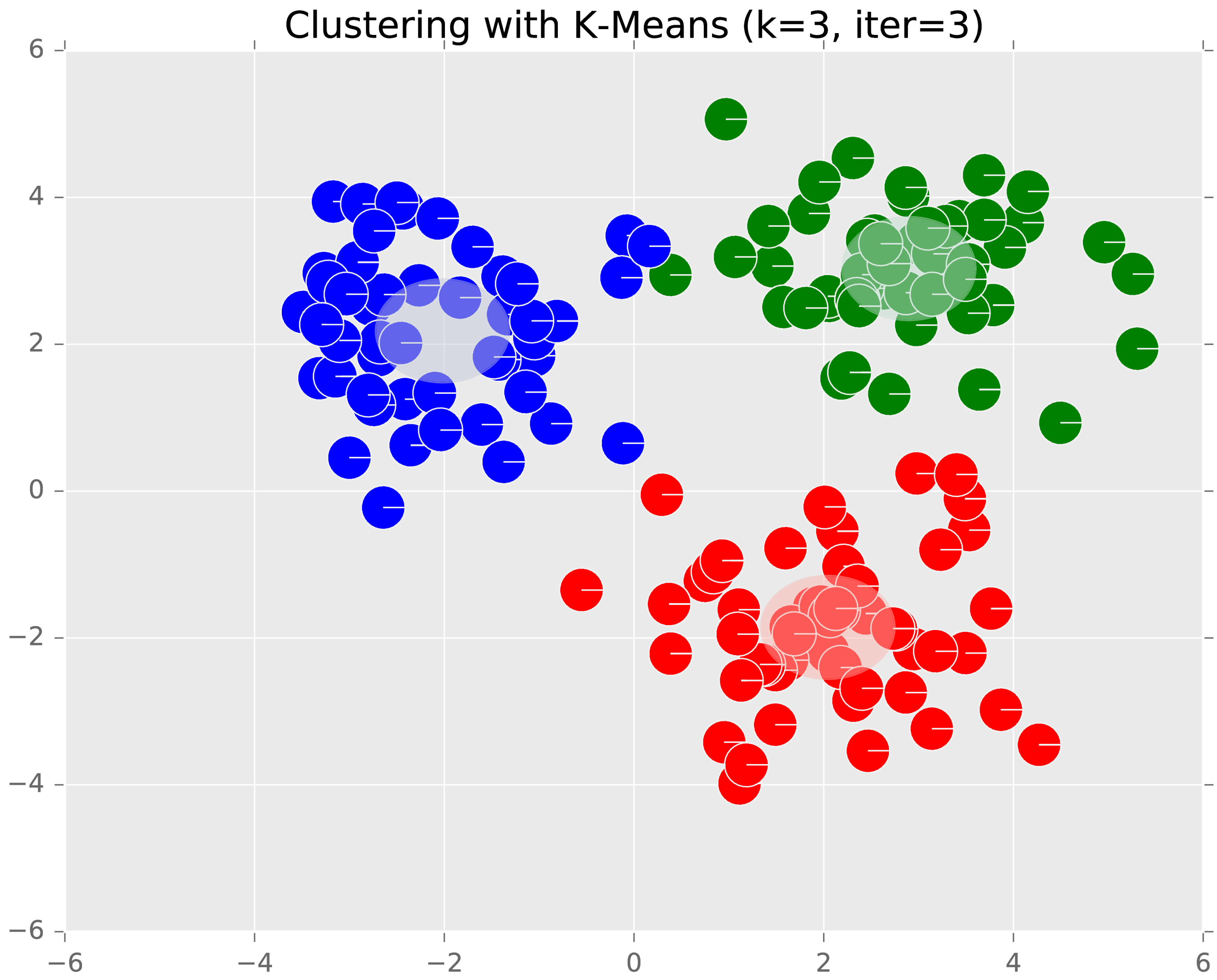


K-Means Algorithm

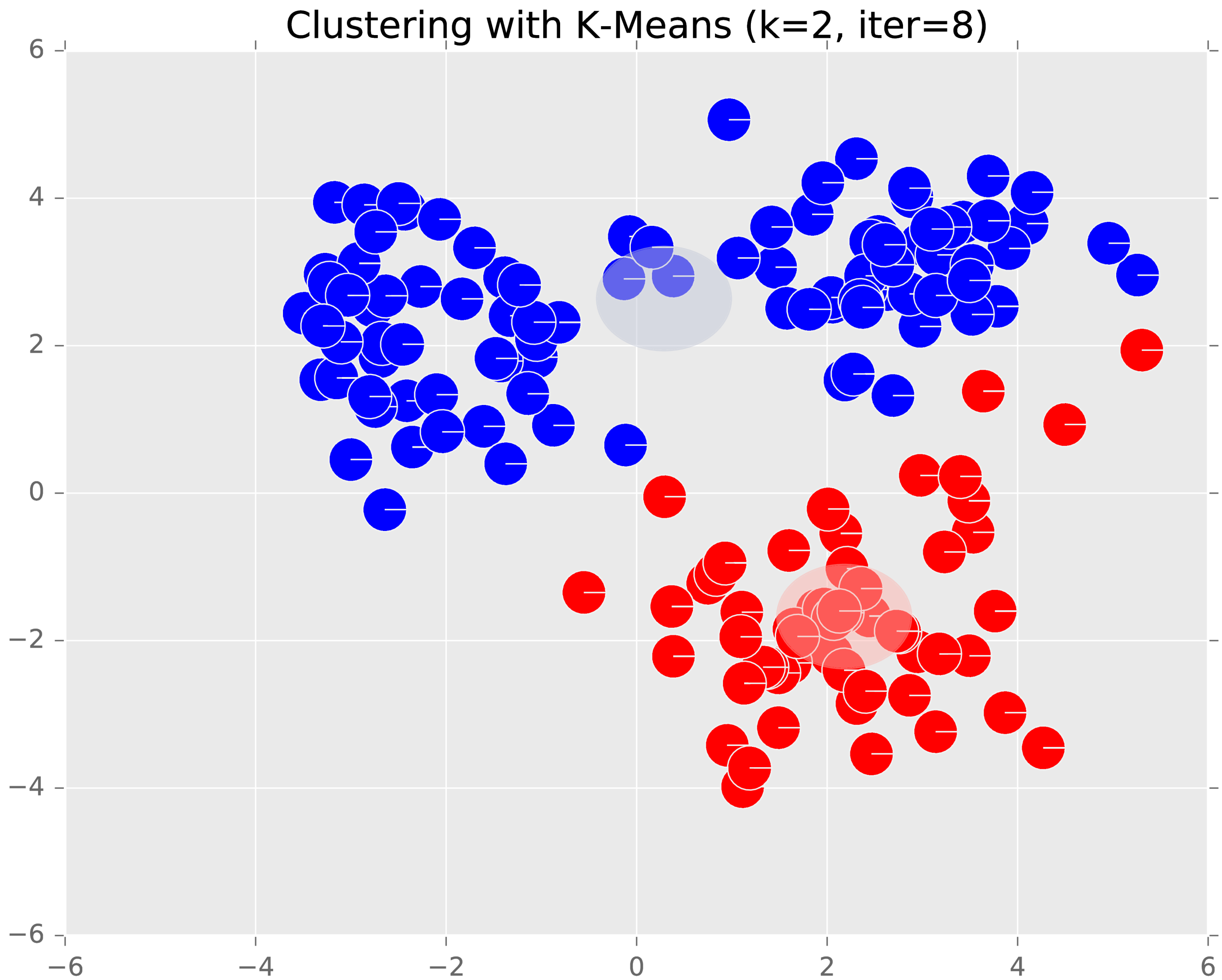
- **Given** unlabeled feature vectors
  $D = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$
- **Initialize** cluster centers $c = \{\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(K)}\}$
- **Repeat** until convergence:
  - for i in $\{1, \ldots, N\}$
    $z^{(i)} \leftarrow$ **index** j of cluster center **nearest** to $\mathbf{x}^{(i)}$
  - for j in $\{1, \ldots, K\}$
    $\mathbf{c}^{(j)} \leftarrow$ **mean** of **all** points assigned to cluster j


bedtime
3:00
12 00
9.00
20
30
40
age

Example: K-Means



Example: K-Means



Poll

Sample Questions

2.2 Lloyd's algorithm

Circle the image which depicts the cluster center positions after 1 iteration of Lloyd's algorithm.

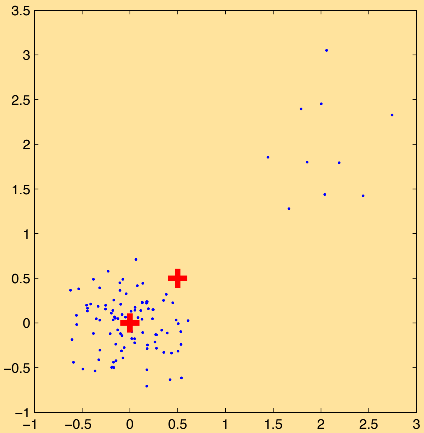
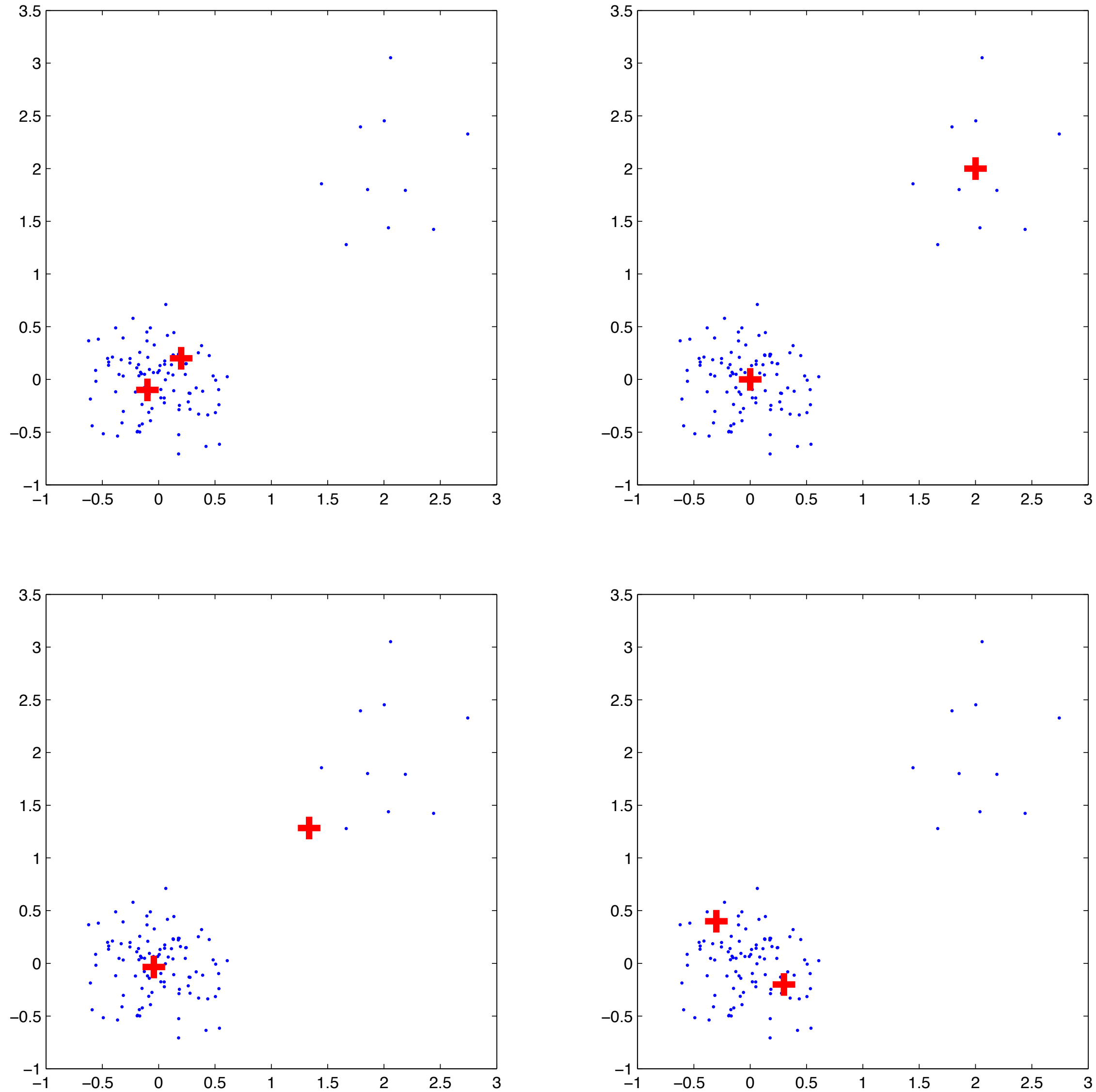


Figure 2: Initial data and cluster centers



Recommender Systems

NETFLIX

Netflix Prize

COMPLETED

Home | Rules | Leaderboard | Update

Leaderboard

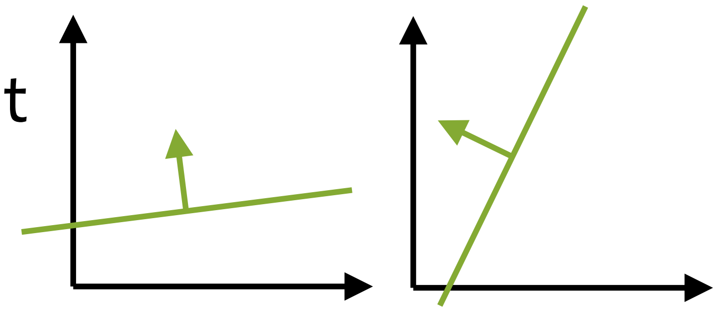
Showing Test Score. Click here to show quiz score

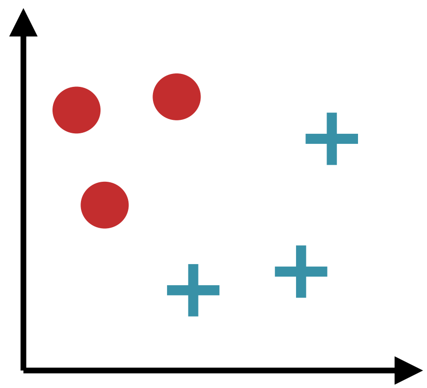
Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace_	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11

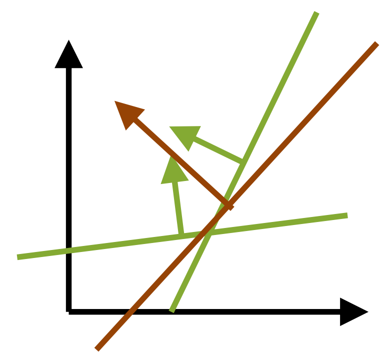
Weighted Majority Algorithm

(Littlestone & Warmuth, 1994)

- **Given:** pool *A* of binary classifiers (that you know nothing about)
- **Data:** stream of examples (i.e. online learning setting)
- **Goal:** design a new learner that uses the predictions of the pool to make new predictions
- **Algorithm:**
  - Initially weight all classifiers equally
  - Receive a training example and predict the (weighted) majority vote of the classifiers in the pool
  - Down-weight classifiers that contribute to a mistake by a factor of β







Weighted Majority Algorithm

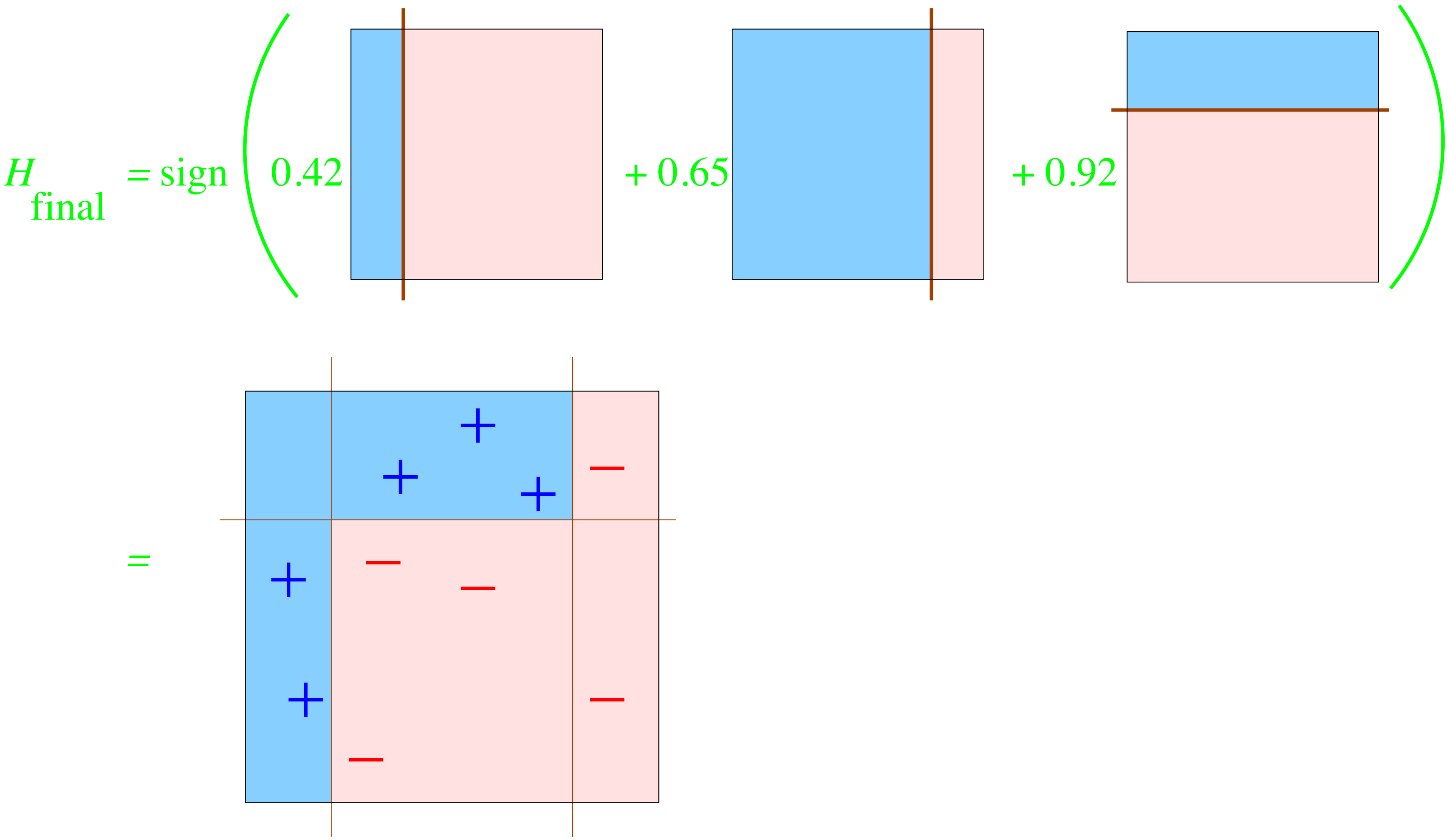
Theorems (Littlestone & Warmuth, 1994)

For the general case where WM is applied to a pool $\mathcal{A}$ of algorithms we show the following upper bounds on the number of mistakes made in a given sequence of trials:

1. $O(\log|\mathcal{A}|+m)$, if one algorithm of $\mathcal{A}$ makes at most m mistakes.
2. $O(\log\frac{|\mathcal{A}|}{k}+m)$, if each of a subpool of k algorithms of $\mathcal{A}$ makes at most m mistakes.
3. $O(\log\frac{|\mathcal{A}|}{k}+\frac{m}{k})$, if the total number of mistakes of a subpool of k algorithms of $\mathcal{A}$ is at most m.

These are "mistake bounds" of the variety we saw for the Perceptron algorithm

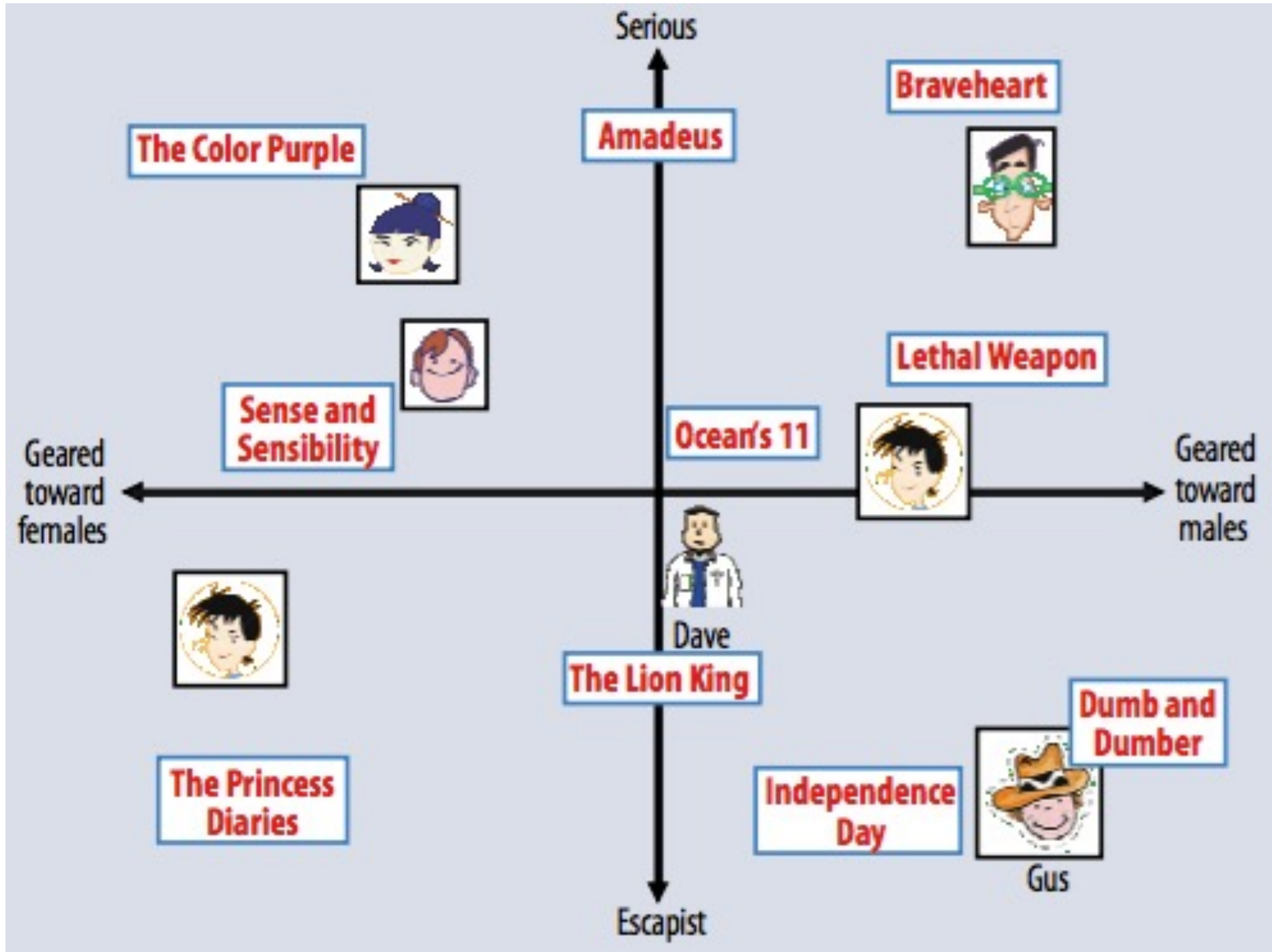
AdaBoost: Toy Example



Two Types of Collaborative Filtering

2. Latent Factor Methods

- Assume that both movies and users live in some **low-dimensional space** describing their properties
- **Recommend** a movie based on its **proximity** to the user in the latent space
- **Example Algorithm:** Matrix Factorization



Figures from Koren et al. (2009)

Example: MF for Netflix Problem

R

		NERO	JULIUS CAESAR	CLEOPATRA	SLEEPLESS IN SEATTLE	PRETTY WOMAN	CASABLANCA
HISTORY	1	1	1	1	0	0	0
	2	1	1	1	0	0	0
	3	1	1	1	0	0	0
BOTH	4	1	1	1	1	1	1
ROMANCE	5	-1	-1	-1	1	1	1
	6	-1	-1	1	1	1	1
	7	-1	-1	-1	1	1	1

$\approx$

U

	HISTORY	ROMANCE
1	1	0
2	1	0
3	1	0
4	1	1
5	-1	1
6	-1	1
7	-1	1

X

V^T

	NERO	JULIUS CAESAR	CLEOPATRA	SLEEPLESS IN SEATTLE	PRETTY WOMAN	CASABLANCA
HISTORY	1	1	1	0	0	0
ROMANCE	0	0	1	1	1	1

(a) Example of rank-2 matrix factorization

E

		NERO	JULIUS CAESAR	CLEOPATRA	SLEEPLESS IN SEATTLE	PRETTY WOMAN	CASABLANCA
HISTORY	1	0	0	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
BOTH	4	0	0	-1	0	0	0
ROMANCE	5	0	0	-1	0	0	0
	6	0	0	1	0	0	0
	7	0	0	-1	0	0	0

(b) Residual matrix

Figures from Aggarwal (2016)

Recommending Movies

Poll

Question:

Suppose you want to build a system that combines elements of collaborative filtering with content filtering, which of the following pieces of information about user behavior could be used to improve such a system?

Select all that apply

A. # of times a user watched a given movie
B. Total # of movies a user has watched
C. How often a user turns on subtitles
D. # of times a user paused a given movie
E. How many accounts a user is associated with
F. # of DVDs a user can rent at a time
G. None of the above

Classification and Regression: The Big Picture

Recipe for Machine Learning

1. Given data $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$
2. (a) Choose a decision function $h_{\boldsymbol{\theta}}(\mathbf{x}) = \cdots$ (parameterized by $\boldsymbol{\theta}$)

   (b) Choose an objective function $J_{\mathcal{D}}(\boldsymbol{\theta}) = \cdots$ (relies on data)
3. Learn by choosing parameters that optimize the objective $J_{\mathcal{D}}(\boldsymbol{\theta})$

$$\hat{\boldsymbol{\theta}} \approx \operatorname*{argmin}_{\boldsymbol{\theta}} J_{\mathcal{D}}(\boldsymbol{\theta})$$

4. Predict on new test example $\mathbf{x}_{\text{new}}$ using $h_{\boldsymbol{\theta}}(\cdot)$

$$\hat{y} = h_{\boldsymbol{\theta}}(\mathbf{x}_{\text{new}})$$

Decision Functions

- Perceptron: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \text{sign}(\boldsymbol{\theta}^T \mathbf{x})$
- Linear Regression: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{x}$
- Discriminative Models: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \operatorname*{argmax}_y p_{\boldsymbol{\theta}}(y \mid \mathbf{x})$
  - Logistic Regression: $p_{\boldsymbol{\theta}}(y = 1 \mid \mathbf{x}) = \sigma(\boldsymbol{\theta}^T \mathbf{x})$
  - Neural Net (classification): $p_{\boldsymbol{\theta}}(y = 1 \mid \mathbf{x}) = \sigma((\mathbf{W}^{(2)})^T \sigma((\mathbf{W}^{(1)})^T \mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})$
- Generative Models: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \operatorname*{argmax}_y p_{\boldsymbol{\theta}}(\mathbf{x}, y)$
  - Naive Bayes: $p_{\boldsymbol{\theta}}(\mathbf{x}, y) = p_{\boldsymbol{\theta}}(y) \prod_{m=1}^{M} p_{\boldsymbol{\theta}}(x_m \mid y)$

Optimization Method

- Gradient Descent: $\boldsymbol{\theta} \rightarrow \boldsymbol{\theta} - \gamma \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$
- SGD: $\boldsymbol{\theta} \rightarrow \boldsymbol{\theta} - \gamma \nabla_{\boldsymbol{\theta}} J^{(i)}(\boldsymbol{\theta})$
  for $i \sim \text{Uniform}(1, \ldots, N)$
  where $J(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} J^{(i)}(\boldsymbol{\theta})$
- mini-batch SGD
- closed form
  1. compute partial derivatives
  2. set equal to zero and solve

Objective Function

- MLE: $J(\boldsymbol{\theta}) = -\sum_{i=1}^{N} \log p(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$
- MCLE: $J(\boldsymbol{\theta}) = -\sum_{i=1}^{N} \log p(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)})$
- L2 Regularized: $J'(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \lambda ||\boldsymbol{\theta}||_2^2$
  (same as Gaussian prior $p(\boldsymbol{\theta})$ over parameters)
- L1 Regularized: $J'(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \lambda ||\boldsymbol{\theta}||_1$
  (same as Laplace prior $p(\boldsymbol{\theta})$ over parameters)

Learning Paradigms

Paradigm	**Data**
Supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$ $\quad \mathbf{x} \sim p^*(\cdot)$ and $y = c^*(\cdot)$
$\hookrightarrow$ Regression	$y^{(i)} \in \mathbb{R}$
$\hookrightarrow$ Classification	$y^{(i)} \in \{1, \ldots, K\}$
$\hookrightarrow$ Binary classification	$y^{(i)} \in \{+1, -1\}$
$\hookrightarrow$ Structured Prediction	$\mathbf{y}^{(i)}$ is a vector
Unsupervised	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ $\quad \mathbf{x} \sim p^*(\cdot)$
Semi-supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N_1} \cup \{\mathbf{x}^{(j)}\}_{j=1}^{N_2}$
Online	$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), (\mathbf{x}^{(3)}, y^{(3)}), \ldots\}$
Active Learning	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ and can query $y^{(i)} = c^*(\cdot)$ at a cost
Imitation Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}), (s^{(2)}, a^{(2)}), \ldots\}$
Reinforcement Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}, r^{(1)}), (s^{(2)}, a^{(2)}, r^{(2)}), \ldots\}$

ML Big Picture

Learning Paradigms:

What data is available and when? What form of prediction?

- supervised learning
- unsupervised learning
- semi-supervised learning
- reinforcement learning
- active learning
- imitation learning
- domain adaptation
- online learning
- density estimation
- recommender systems
- feature learning
- manifold learning
- dimensionality reduction
- ensemble learning
- distant supervision
- hyperparameter optimization

Problem Formulation:

What is the structure of our output prediction?

boolean	Binary Classification
categorical	Multiclass Classification
ordinal	Ordinal Classification
real	Regression
ordering	Ranking
multiple discrete	Structured Prediction
multiple continuous	(e.g. dynamical systems)
both discrete & cont.	(e.g. mixed graphical models)

Application Areas

Key challenges?

NLP, Speech, Computer Vision, Robotics, Medicine, Search

Theoretical Foundations:

What principles guide learning?

- ❑ probabilistic
- ❑ information theoretic
- ❑ evolutionary search
- ❑ ML as optimization

Facets of Building ML Systems:

How to build systems that are robust, efficient, adaptive, effective?

1. *Data prep*
2. *Model selection*
3. *Training (optimization / search)*
4. *Hyperparameter tuning on validation data*
5. *(Blind) Assessment on test data*

Big Ideas in ML:

Which are the ideas driving development of the field?

- *inductive bias*
- *generalization / overfitting*
- *bias-variance decomposition*
- *generative vs. discriminative*
- *deep nets, graphical models*
- *PAC learning*
- *distant rewards*

Course Level Objectives

You should be able to…

1. Implement and analyze existing learning algorithms, including well-studied methods for classification, regression, structured prediction, clustering, and representation learning
2. Integrate multiple facets of practical machine learning in a single system: data preprocessing, learning, regularization and model selection
3. Describe the the formal properties of models and algorithms for learning and explain the practical implications of those results
4. Compare and contrast different paradigms for learning (supervised, unsupervised, etc.)
5. Design experiments to evaluate and compare different machine learning techniques on real-world problems
6. Employ probability, statistics, calculus, linear algebra, and optimization in order to develop new predictive models or learning methods
7. Given a description of a ML technique, analyze it to identify (1) the expressive power of the formalism; (2) the inductive bias implicit in the algorithm; (3) the size and complexity of the search space; (4) the computational properties of the algorithm: (5) any guarantees (or lack thereof) regarding termination, convergence, correctness, accuracy or generalization power.

Course Staff

Education Associate


Brynn Edmunds

Instructor


Matt Gormley

Team A (HW2, HW6)


Sami Kale


Abuzar Khan

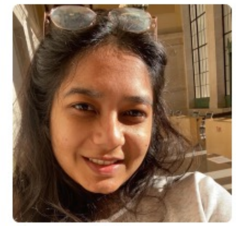

Bhargav Hadya


Aditya Bansal


Erin Gao

Team B (HW3, HW7)


Monica Geng


Asmita Hajra


Dishani Lahiri


Yash Gupta


Siva Subramanian

Team C (HW4, HW8)


Alex Xie


Pranay Gundam


Tanvi Karandikar


Emaan Ahmed


Pranit Chawla

Team D (HW5, HW9)


Abhishek Vijayakumar


Tara Lakdawala


Hang Shu


(Ads) Advaith Sridhar


Poorvi Hebbar