Machine Learning at Scale

James G. Shanahan[2]

Assistants: Jason Anastasopoulos[2] Liang Dai[1],

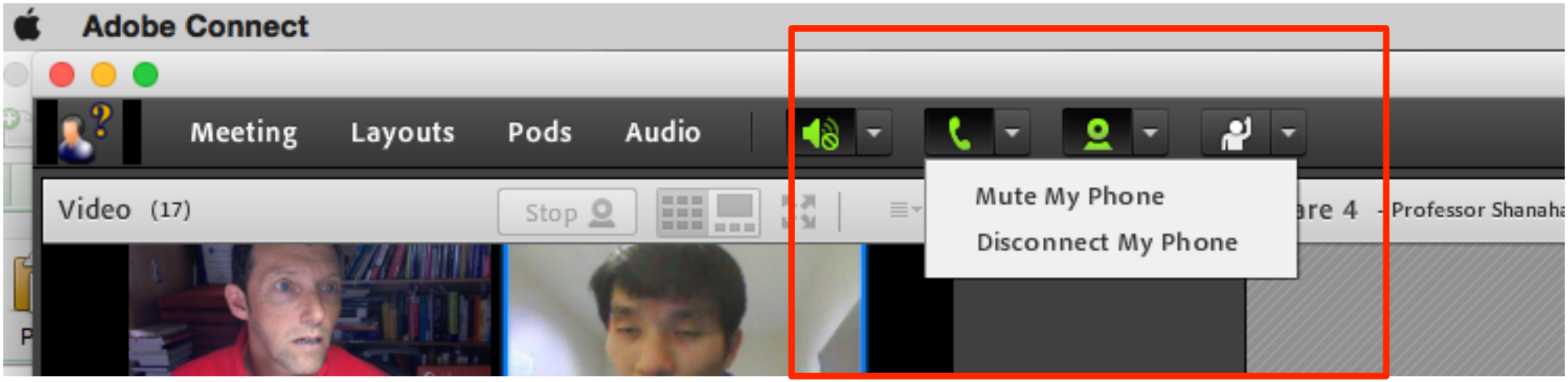
[1]NativeX, [2]iSchool UC Berkeley, CA, [3]UC Santa Cruz

EMAIL: James_DOT_Shanahan_AT_gmail_DOT_com

Live Session #1

January 12, 2016

Mute your mikes via Adobe Connect



Live Sessions Spring 2016

- Monday UCB-MIDS 1 W261 James Shanahan 7:00 PM - 8:30 PM EST
- **Tuesday UCB-MIDS 4 W261 Jason/James Shanahan 7:00 PM - 8:30 PM EST**
- **Wednesday UCB-MIDS 2 W261 James Shanahan 7:00 PM - 8:30 PM EST**
- **Wednesday UCB-MIDS 3 W261 James Shanahan 9:30 PM - 11:00 PM EST**

Live Session Outline

- **Welcome & Class Introductions**
  - Please mute you microphones
  - Start RECORDING (bonus points for reminding me!)
  - Class, homework, project Logistics + Office hours
  - Self-introductions (Bios + WWK01: Q1)
- **Class Intro**
- **Q&A (WK01)**
- **Probability theory introduction**
- **Naive Bayes**
  - Basic derivation
  - Various Naïve Bayes Flavours (Live Session #2)
- **Wrapup**
  - Finish RECORDING (bonus points for reminding me!)
  - Click End Meeting

Starting a Live Session

1. Start & Connect to the Audio

2. Start Your Webcam &
   Enable Webcam for Participants

3. Start Recording

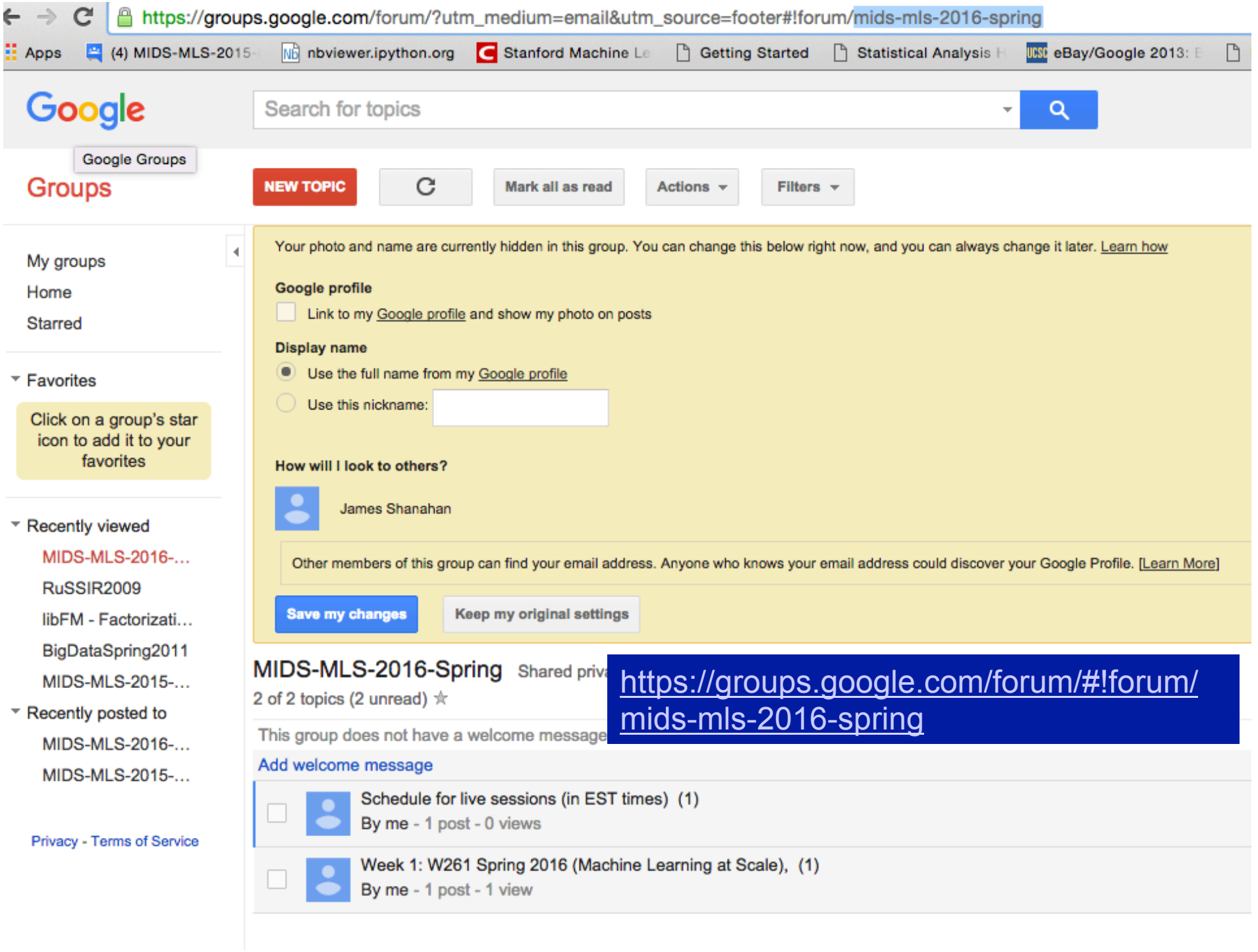
4. Allow Participants Into the Room

5. Merge Speaker Audio (if necessary)

This is also available on Page 3 of the "Live Session Essentials" document that you downloaded during Training Session 1.

Live sessions

- 80 minutes: directed discussion by instructors
- 10 minutes free talk


https://groups.google.com/forum/?utm_medium=email&utm_source=footer#!forum/mids-mls-2016-spring
Apps
(4) MIDS-MLS-2015-
nbviewer.ipython.org
Stanford Machine Le
Getting Started
Statistical Analysis H
eBay/Google 2013: E
Google
Search for topics
Google Groups
Groups
NEW TOPIC
Mark all as read
Actions
Filters
My groups
Home
Starred
Favorites
Click on a group's star icon to add it to your favorites
Recently viewed
MIDS-MLS-2016-...
RuSSIR2009
libFM - Factorizati...
BigDataSpring2011
MIDS-MLS-2015-...
Recently posted to
MIDS-MLS-2016-...
MIDS-MLS-2015-...
Privacy - Terms of Service
Your photo and name are currently hidden in this group. You can change this below right now, and you can always change it later. Learn how
Google profile
Link to my Google profile and show my photo on posts
Display name
Use the full name from my Google profile
Use this nickname:
How will I look to others?
James Shanahan
Other members of this group can find your email address. Anyone who knows your email address could discover your Google Profile. [Learn More]
Save my changes
Keep my original settings
MIDS-MLS-2016-Spring
Shared priva
2 of 2 topics (2 unread)
https://groups.google.com/forum/#!forum/mids-mls-2016-spring
This group does not have a welcome message
Add welcome message
Schedule for live sessions (in EST times) (1)
By me - 1 post - 0 views
Week 1: W261 Spring 2016 (Machine Learning at Scale), (1)
By me - 1 post - 1 view

Important dates

- Monday, January 11, 2016 -- Classes start Monday,
- January 18, 2016 -- Academic and administrative holiday
- **Friday, February 5, 2016 -- Last day to add or drop a class**
- Monday, February 15, 2016 -- Academic and administrative holiday
- Tuesday, March 22, 2016 – Friday, March 25, 2016 -- Academic holiday (spring break) *Note: classes will be held on Mon., Mar 21.*
- Friday, April 29, 2016 -- Last day of semester (Friday week 15)
- Wednesday, May 18, 2016 -- Grades due

13 Lectures, 1 Midterm, End of term exam, plus weekly homework and projects

Semester Week	Week in 2016	Monday	Notes	
1	Week 3	1/11/16		
2	Week 4	1/18/16		
3	Week 5	1/25/16		
4	Week 6	2/1/16		
5	Week 7	2/8/16		
6	Week 8	2/15/16		
7	Week 9	2/22/16		
8	Week 10	2/29/16	Mid Term Exam	
9	Week 11	3/7/16		
10	Week 12	3/14/16		
Spring Break	Week 13	3/21/16	Spring Break	
11	Week 14	3/28/16		
12	Week 15	4/4/16		
13	Week 16	4/11/16		
14	Week 17	4/18/16		
15	Week 18	4/25/16	End of term	4/29/15
	Week 19	5/2/16		
		5/18/2016 (Wednesday)		Grades Due

Mid term and end of term exams

- Will consist of Multiple choice questions
- Exam period will be your normal live sessions
- Exams are open book but no communication is allowed between students (exams will be proctored via an online proctoring system).

Homework and Projects

- **Each week you will have homework**
- **Homework will consist of lecture content based questions and project-based questions**
- **Projects will have phases (all structured) and where each phase will make up part of homework**
  - Phase 1 could make up HW 7
  - Phase 2 could make up HW 8
  - Phase 3 could make up HW 9

Logistics/Performance Evaluation

High ROI class: opens up new worlds;

13 Lecture weeks with 2 exam type weeks

- Facetime/Live session 90 minutes each week on Wednesdays
- Week 8 (mid term exam) and Week 15 (final project)
  - 2-3 hours

Performance Evaluation:

Homework+Projects 40% (iterate/agile): Projects will have phases (all structured)

Midterm Exam 20% (Week 8 of the semester)

Class participation 20%

- fact-to-face and in the online forums; please answer each others questions; collaborate; communicate

End term exam 20% (Week 15 of the Semester)

jupyter nbviewer

JUPYTER FAQ

Homework submissions will be in terms of iPython Notebooks

DATASCI W261: Machine Learning at Scale

Name,
Email address
Time of Submission
W261-1 Fall 2015
Week 9: Homework
November 3, 2015

HW9.0: Short answer questions

> What is PageRank and what is it used for in the context of web search?

PageRank is a graph-based algorithm that counts the number of links, each weighted its quality, that goes to a particular node, which in turn determines a PageRank score. In the context of web search, each node is a webpage and each link is the backlink for a particular page. The PageRank score determines the importance of the particular webpage. During websearch, a search engine typically brings out the pages with the highest PageRank scores in descending order.

> What modifications have to be made to the webgraph in order to leverage the machinery of Markov Chains to compute the steady stade distibuton?

The main modification is the handling of the dangling nodes, which are nodes with no out-links. This violates the irreducibility requirement, which is important for convergence of PageRank. In order to overcome this "dangling nodes" problem, teleportation is used where there is a chance (the damping factor) that any node will be "teleported" to any other node with equal probability.

> OPTIONAL: In topic-specific pagerank, how can we insure that the irreducible property is satified? (HINT: see HW9.4)

In order to guarantee satisfaction of irreducible property, we need to ensure nodes that are not reachable by any nonzero nodes in the topic-specific factors are removed. In practice, this is in general not a big issue and can typically be ignored.

Homework Submissions

- **Name, date stamp of submission, HW,**
- **Notebook submissions on ISVC**
  - Upload Notebook and its PDF
  - PDF (dropbox/Github)
  - If using Dropbox please provide a NBViewer link (http://nbviewer.ipython.org/) and the raw dropbox link
- **Submit on time!**

All Homeworks due....

- **All Homeworks due the following Tuesday at 8AM West coast time**

- ***Week-N* Homework will be due by Tuesday of *Week N+1* at 8AM West Coast Time.**

  - E.g., Week 1 homework is due by 8:00 AM (west coast time), Tuesday of week 2
  - E.g., Week 2 homework is due by 8:00 AM (west coast time), Tuesday of week 3
  - Etc…

Weekly Office Hours: on demand

- Propose to have on Mondays, at 5:30PM West Coast Time

- You can only attend if you submit a question(s) 24 hours before the start of office hours to the Google Group

Live Session Outline

- **Welcome & Class Introductions**
  - Please mute you microphones
  - Start RECORDING (bonus points for reminding me!)
  - Class, homework, project Logistics + Office hours
  - Self-introductions (Bios + WWK01: Q1)
- **Class Intro**
- **Q&A (WK01)**
- **Probability theory introduction**
- **Naive Bayes**
  - Basic derivation
  - Various Naïve Bayes Flavours (Live Session #2)
- **Wrapup**
  - Finish RECORDING (bonus points for reminding me!)
  - Click End Meeting

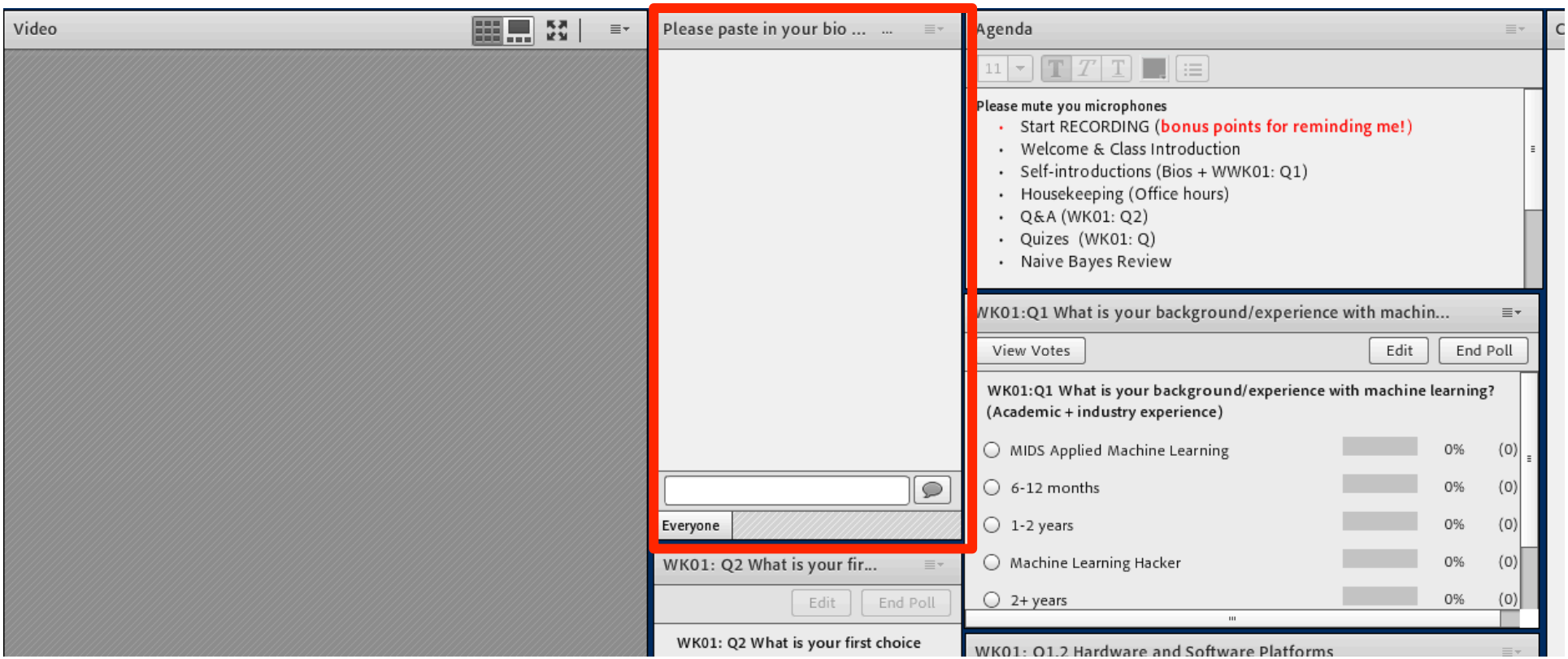
Self Introductions

- **Paste the following into the chat pod**
  - Your Location
  - MIDS
    - When did you start MIDS
    - and planned finish date
  - Background (paste 100 word bio into chat)
  - What you want to get out of this class
- **Get a local mug/glass (to remind us where you are from)**

Bio: Dr. James G. Shanahan

- **25 years experience in data science (AKA data mining).**
- **Currently:**
  - Church and Duncan Group, Inc. (2007), a boutique consultancy in large-scale data science
- **Previously**
  - SVP Data Science and chief scientist at NativeX, a mobile ad network.
  - (co)founded several companies, including Church and Duncan Group, Inc. (2007), a boutique consultancy in large-scale data science; RTBFast (2012), a real-time bidding engine infrastructure play; and Document Souls (1999), an anticipatory information system.
  - In addition, he has held appointments at Xerox Research, Mitsubishi Research, Clairvoyance Corp (a spinoff research lab from Carnegie Mellon University); and Turn Inc. (a technology leader in online advertising).
- **University of California at Berkeley and at Santa Cruz since 2009**
  - where he teaches graduate courses on big data analytics, distributed systems, machine learning, and stochastic optimization.
  - He also advises several high-tech start-ups and is executive vice president of science and technology at Irish Innovation Center (IIC).
- **He has published six books, more than 50 research publications, and over 20 patents in the areas of machine learning and information processing.**
- **PhD in engineering mathematics from the University of Bristol, UK, and holds a bachelor of science degree from the University of Limerick, Ireland. He is an EU Marie Curie fellow. In 2011 he was selected as a member of the Silicon Valley 50 (Top 50 Irish Americans in Technology).**

Self-Introductions:
Drop your bios in this chat



Self Introductions

- **Paste the following into the chat pod**
  - Your Location
  - MIDS
    - When did you start MIDS
    - and planned finish date
  - Background (paste 100 word bio into chat)
  - What you want to get out of this class
- **Get a local mug/glass (to remind us where you are from)**

Live Sessions Spring 2016

- Monday UCB-MIDS 1 W261 James Shanahan 7:00 PM - 8:30 PM EST
- **Tuesday UCB-MIDS 4 W261 Jason/James Shanahan 7:00 PM - 8:30 PM EST**
- **Wednesday UCB-MIDS 2 W261 James Shanahan 7:00 PM - 8:30 PM EST**
- **Wednesday UCB-MIDS 3 W261 James Shanahan 9:30 PM - 11:00 PM EST**

GROUP 4
Tuesday UCB-MIDS
4PM-5:30PM

Bay Area

Ohio

DC

Bay Area

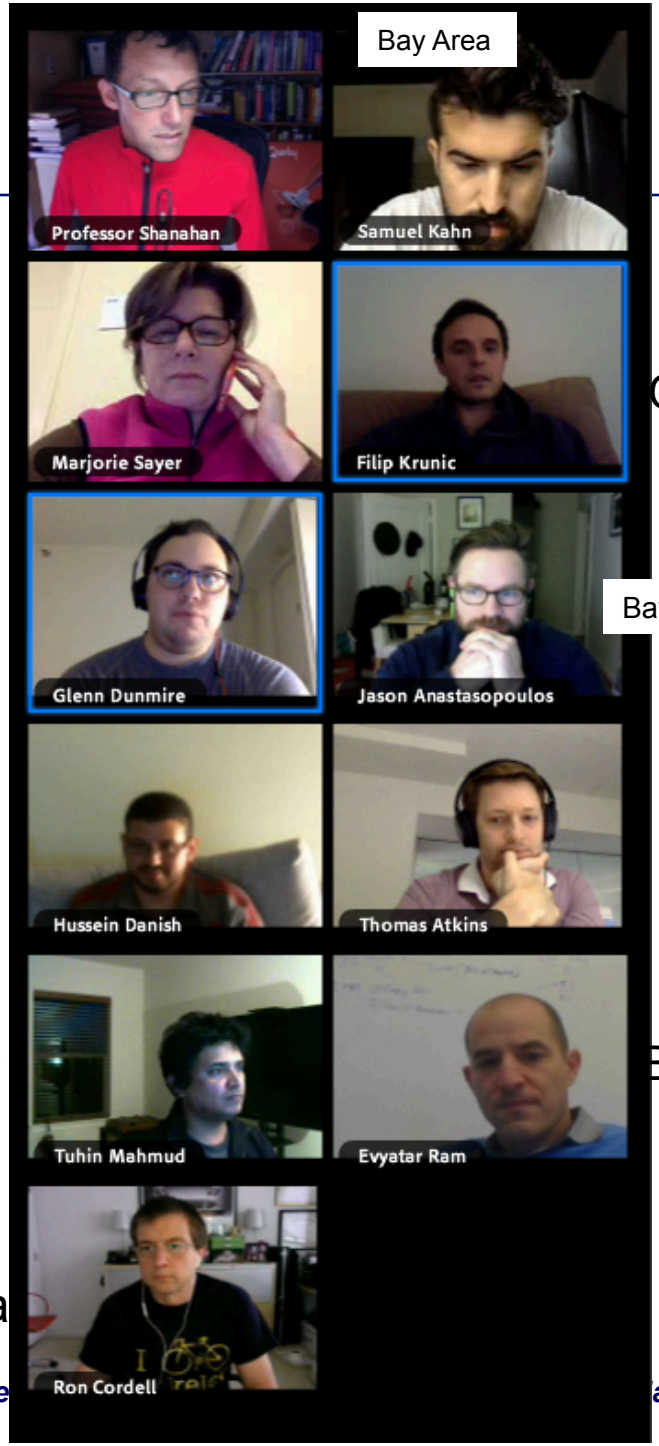
Montreal

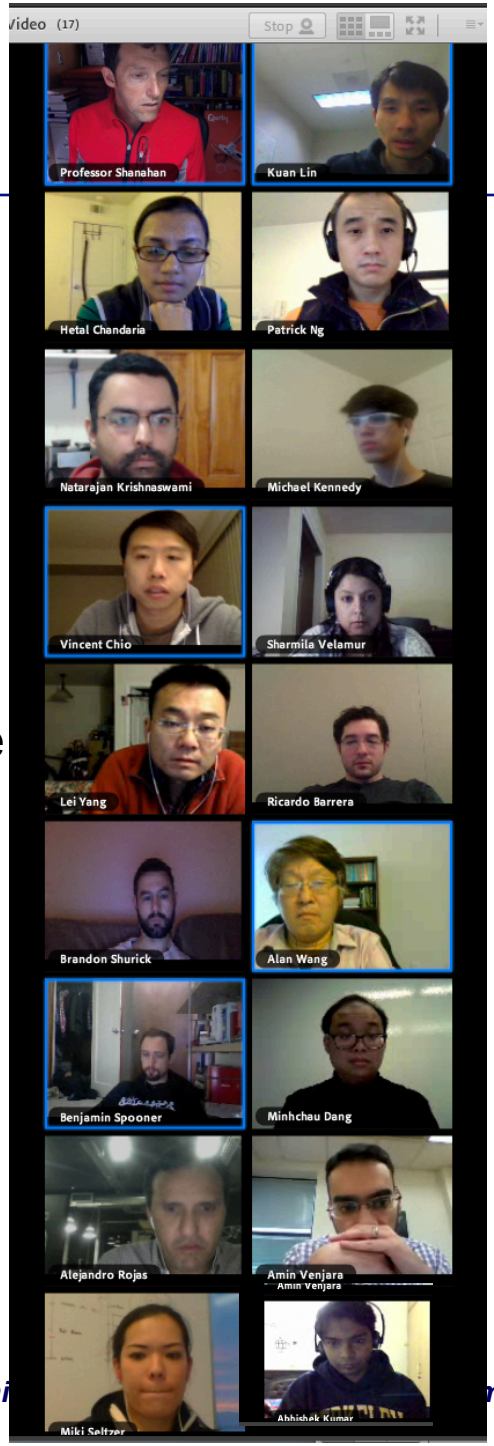
Australia (Sydney)

Austin TX

Bay Area

Bay Area





SoCal

Group

eBay

Hong Kong

NYC

UC Davis, Ca

GROUP 2
Wednesday UCB-MIDS
4PM-5:30PM

Toronto

Idaho

NY state

Seattle

Amsterdam

Socal: Irvine, CA

SoCal

SoCa

Santiago, Chile

New Jersey

SoCal

Delhi, India

SoCal

Cupertino

DC

Louisiana

Seattle

Group 3:
Wednesday 6:30-8:00

Bay Area

Silicon Valley

San Jose, CA

Bay Area

Seattle

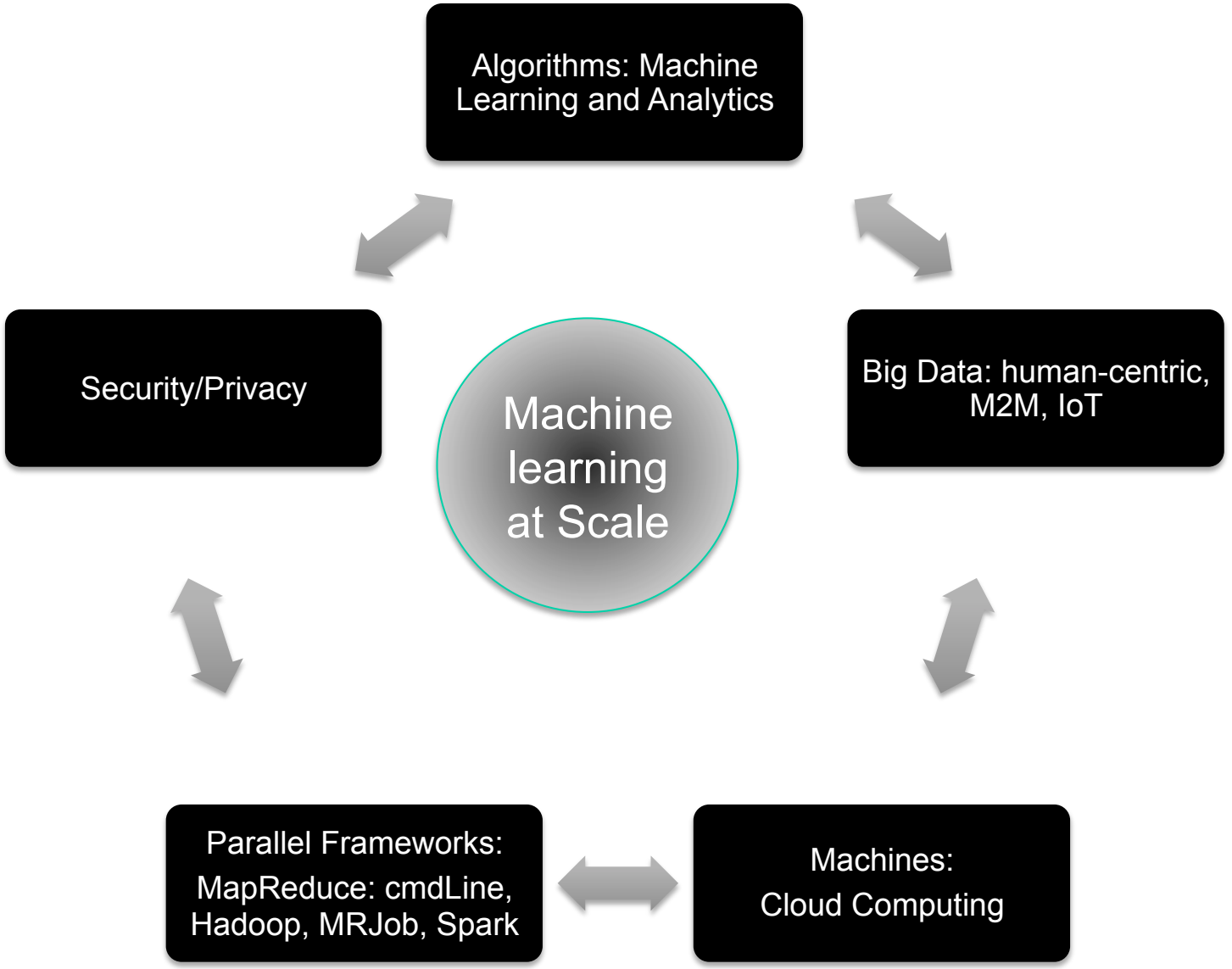
NYC

San Jose

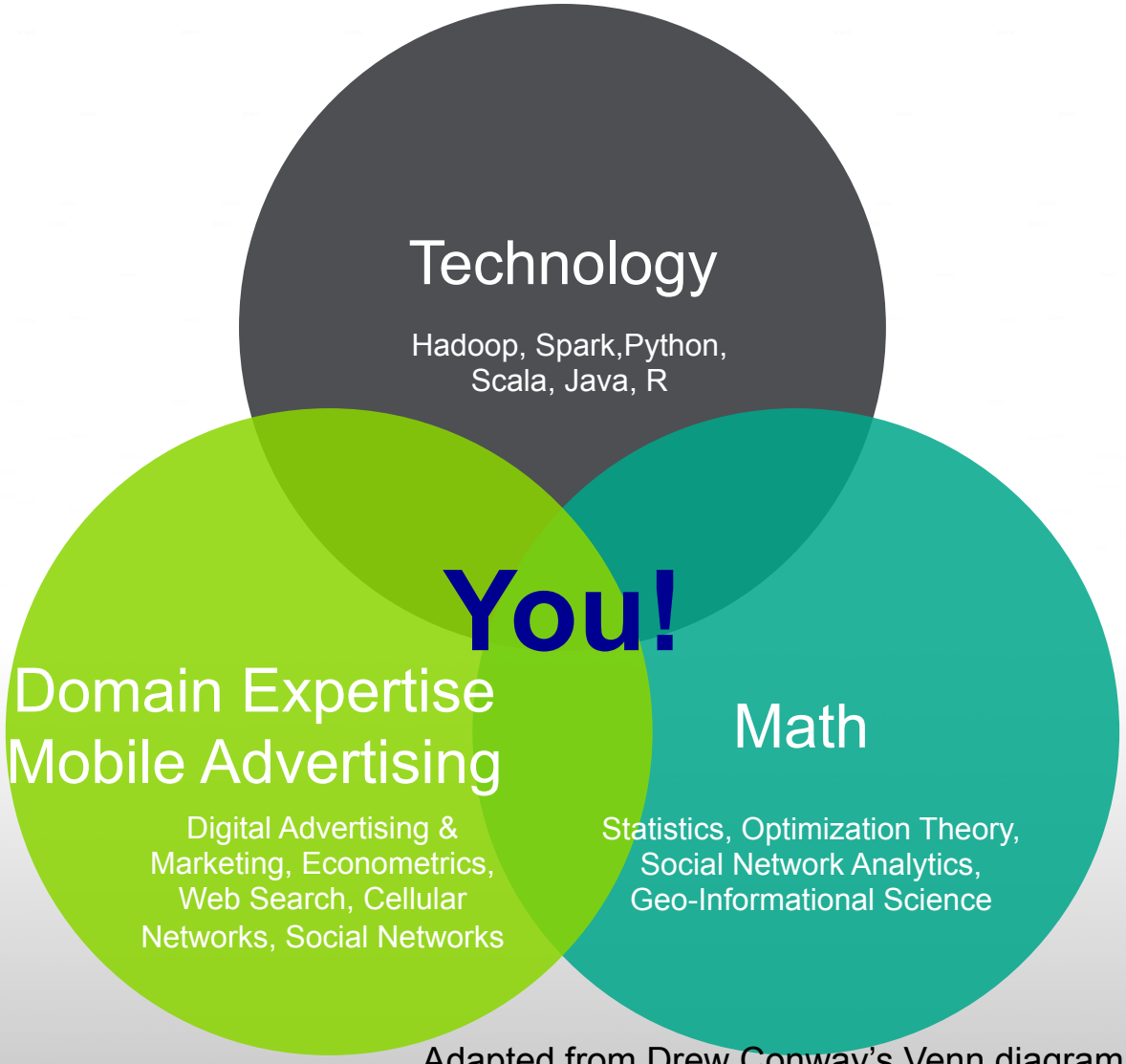
Live Session Outline

- **Welcome & Class Introductions**
  - Please mute you microphones
  - Start RECORDING (bonus points for reminding me!)
  - Class, homework, project Logistics + Office hours
  - Self-introductions (Bios + WWK01: Q1)
- **Class Intro**
- **Q&A (WK01)**
- **Probability theory introduction**
- **Naive Bayes**
  - Basic derivation
  - Various Naïve Bayes Flavours (Live Session #2)
- **Wrapup**
  - Finish RECORDING (bonus points for reminding me!)
  - Click End Meeting

Machine learning at Scale

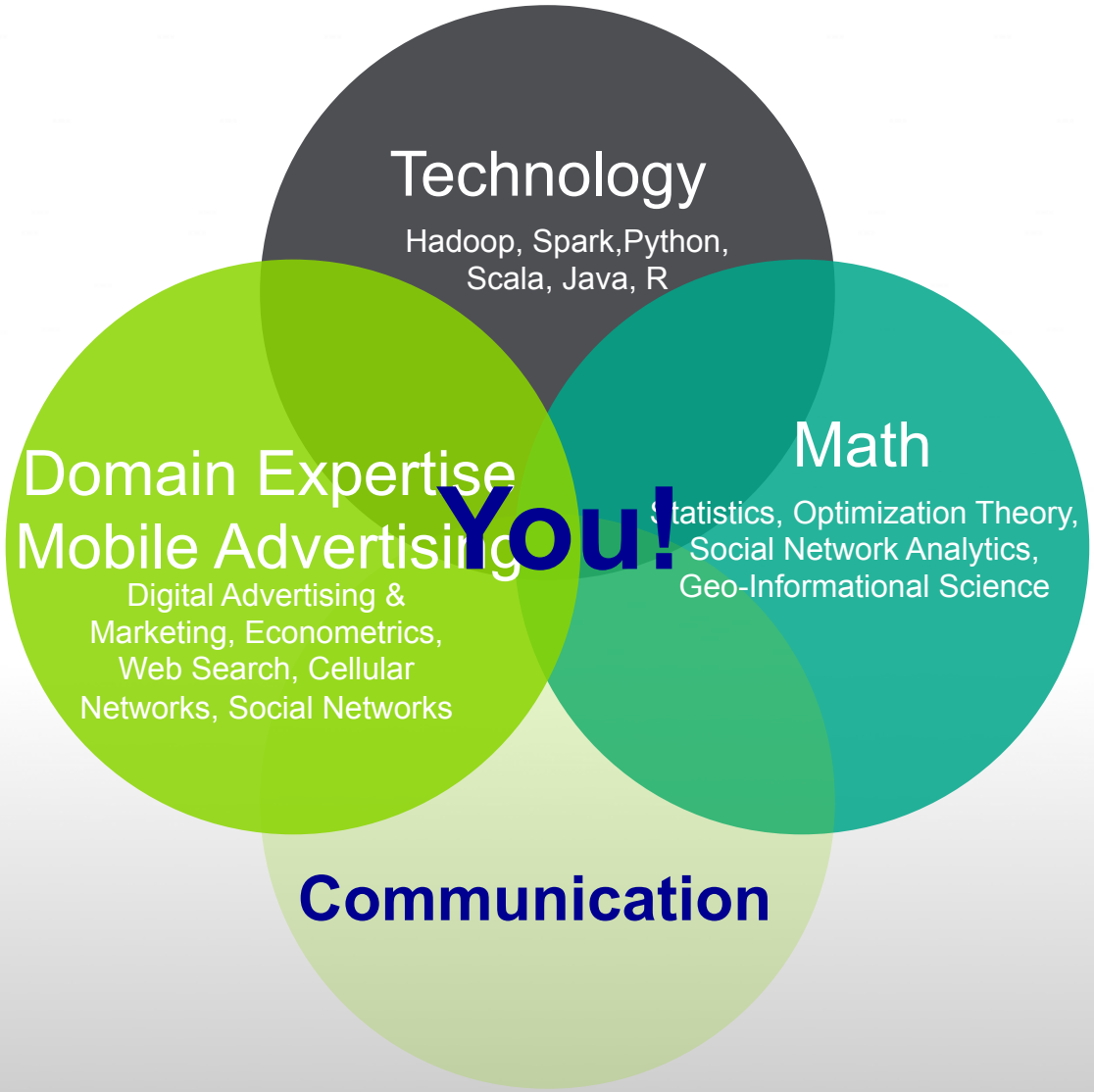


Data Science

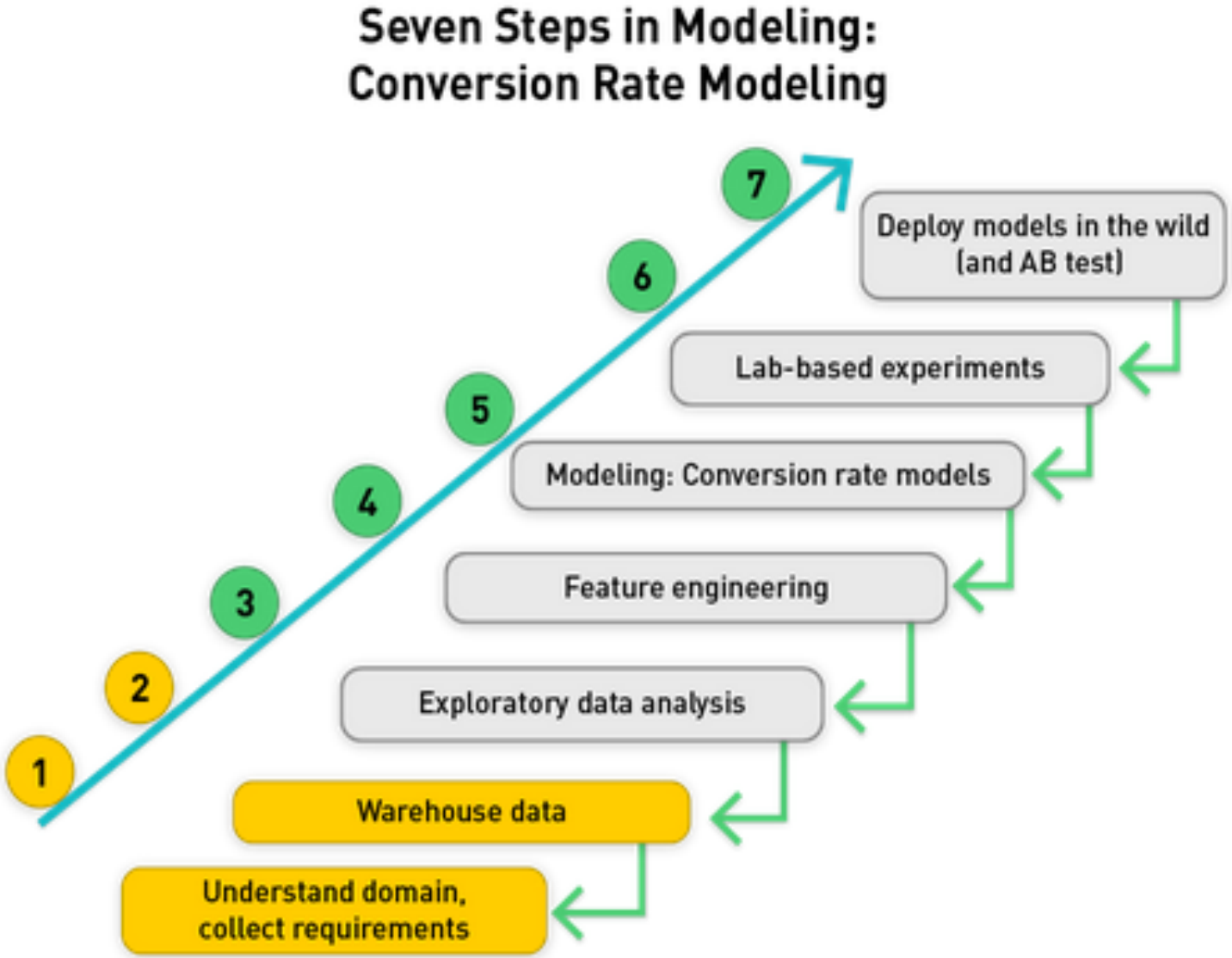


Adapted from Drew Conway's Venn diagram of data science

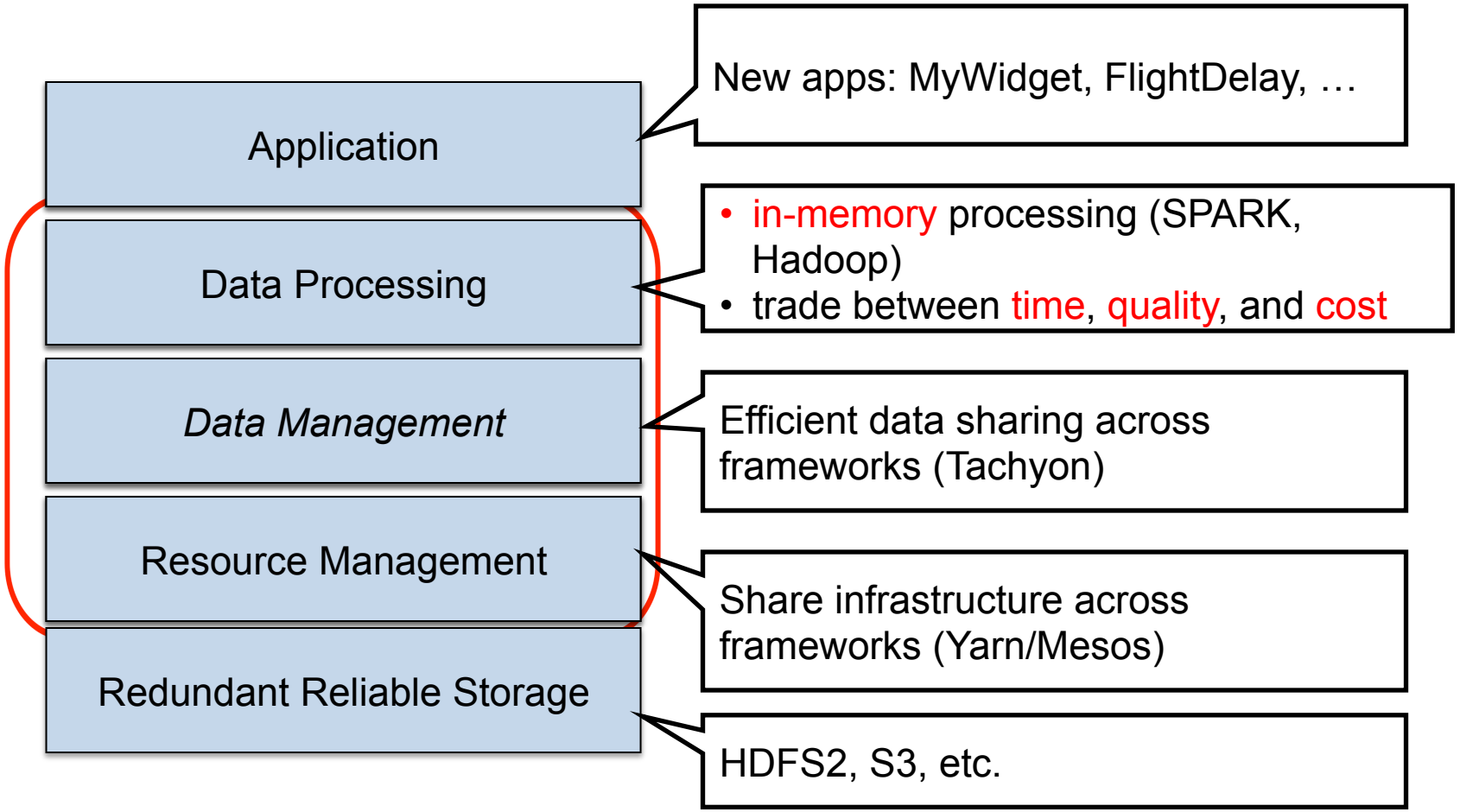
Data Scientist



Typical Abstract Data Science Pipeline

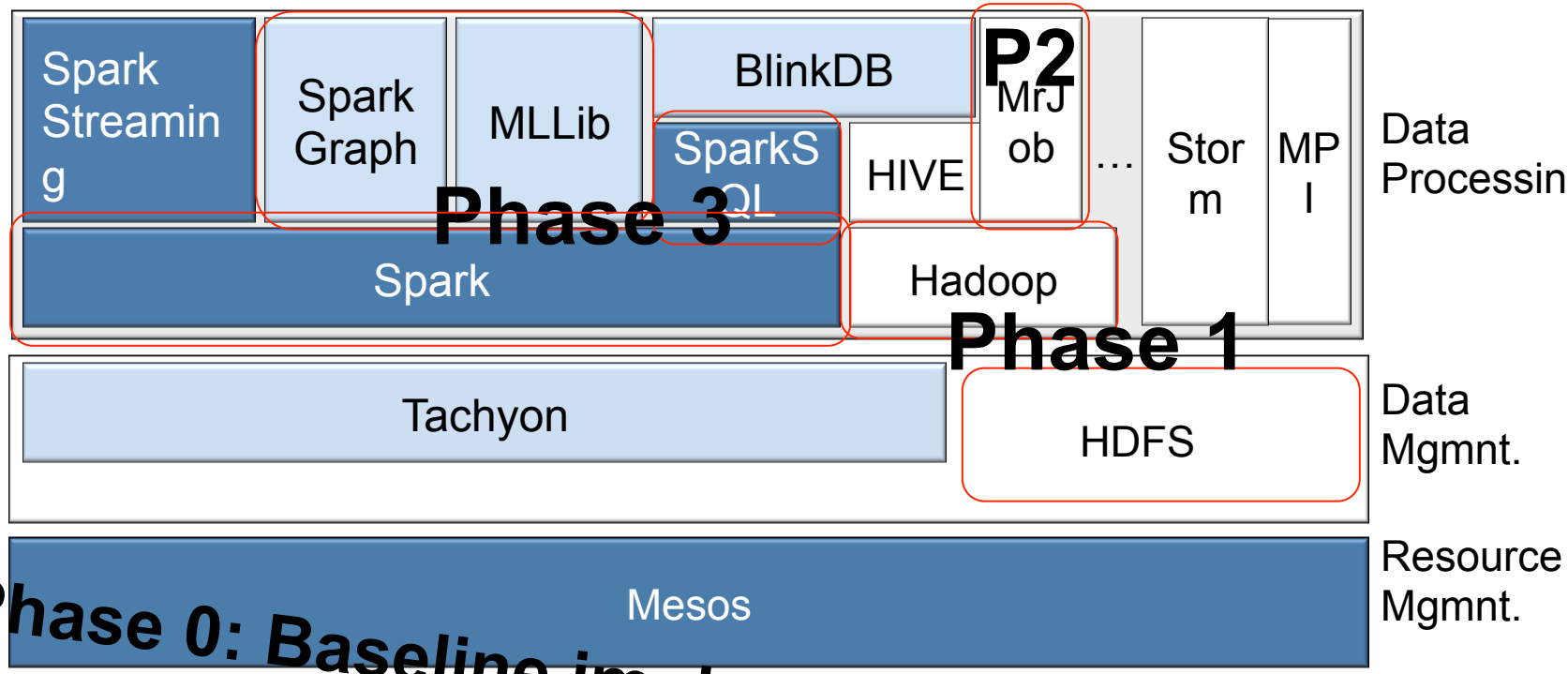


Berkeley Data Analytics Stack (BDAS)



Machine Learning at Scale: Class Phases

- **Focus on distributed computation using functional programming over very large datasets (highlighted in RED)**
- **Develop scalable ML algorithms**
- **Phase 0: Command line; Phase 1: Hadoop/HDFS; Phase 2: MrJob; Phase 3: Spark**

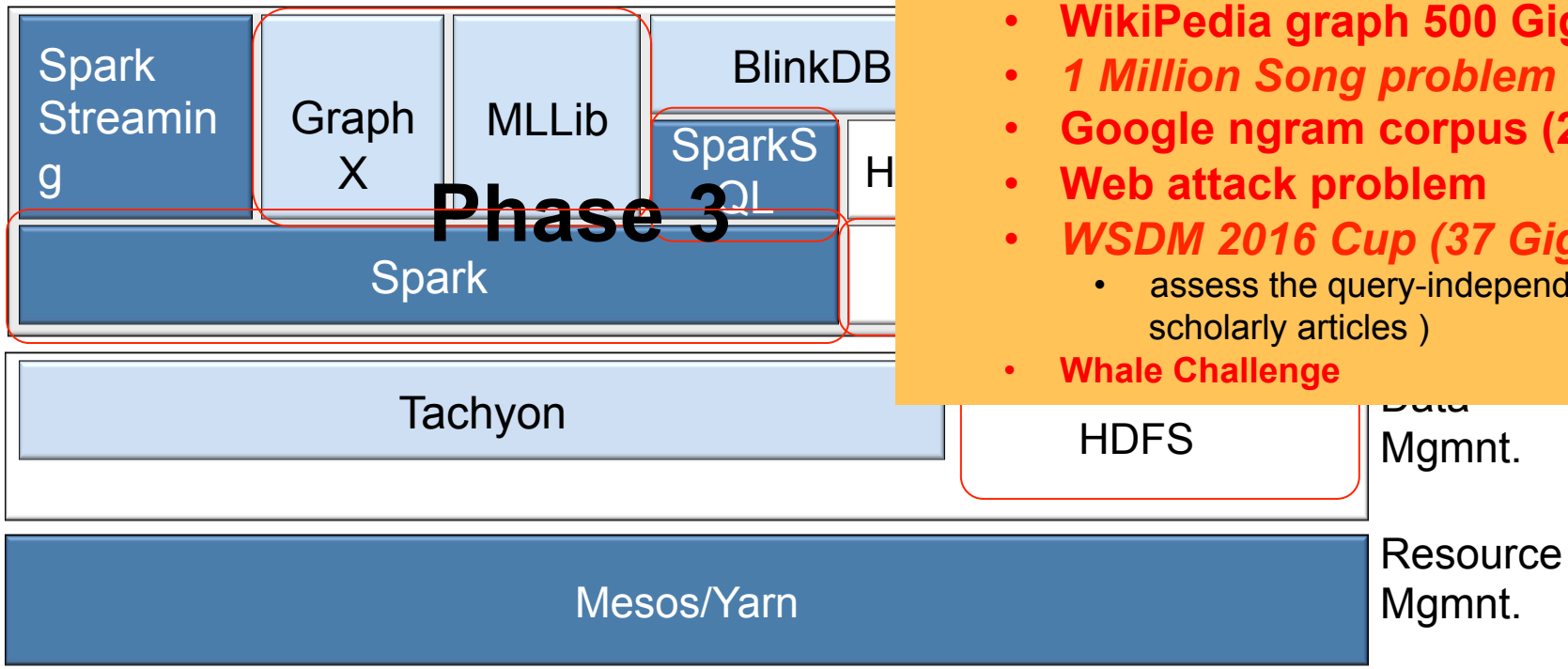


Machine Learning at Scale: Class Phases

- **Distributed computation using functional programming-like paradigm PLUS scalable ML algorithms PLUS Applications**
- **Phase 0: Poorman MapReduce (Unit 1)**
- **Phase 1: Hadoop/HDFS (Units 2 – 3);**
- **Phase 2: MrJob; (Units 4 – 9)**
- **Phase 3: Spark (unit 10 – 14)**

+

- **ML, Graphs,**
- **Case Studies**
  - **$100 Billion problem CTR pre**
  - **WikiPedia graph 500 Gig**
  - ***1 Million Song problem***
  - **Google ngram corpus (2.2 TB**
  - **Web attack problem**
  - ***WSDM 2016 Cup (37 Gig)***
    - assess the query-independent imp scholarly articles)
  - **Whale Challenge**



ML + Systems + CaseStudies == ML at Scale

- **Parallel frameworks for big data**
  - Unix, Hadoop, MrJob, Spark
- **Supervised Machine Learning**
  - Convex optimization, gradient descent, linear regression, decision trees, ensembles of models, support vector machines
- **Unsupervised**
  - Expectation maximization, matrix multiplication, alternating least squares
- **Graphs**
  - random walks, PageRank, graph search algorithms such as BFS, shortest path
- **Hybrid algos**
  - Supervised ML + Random walks
- **Applications**
  - Digital advertising, social media, healthcare, ecommerce, entertainment

Plus
- Metrics
- Statistics

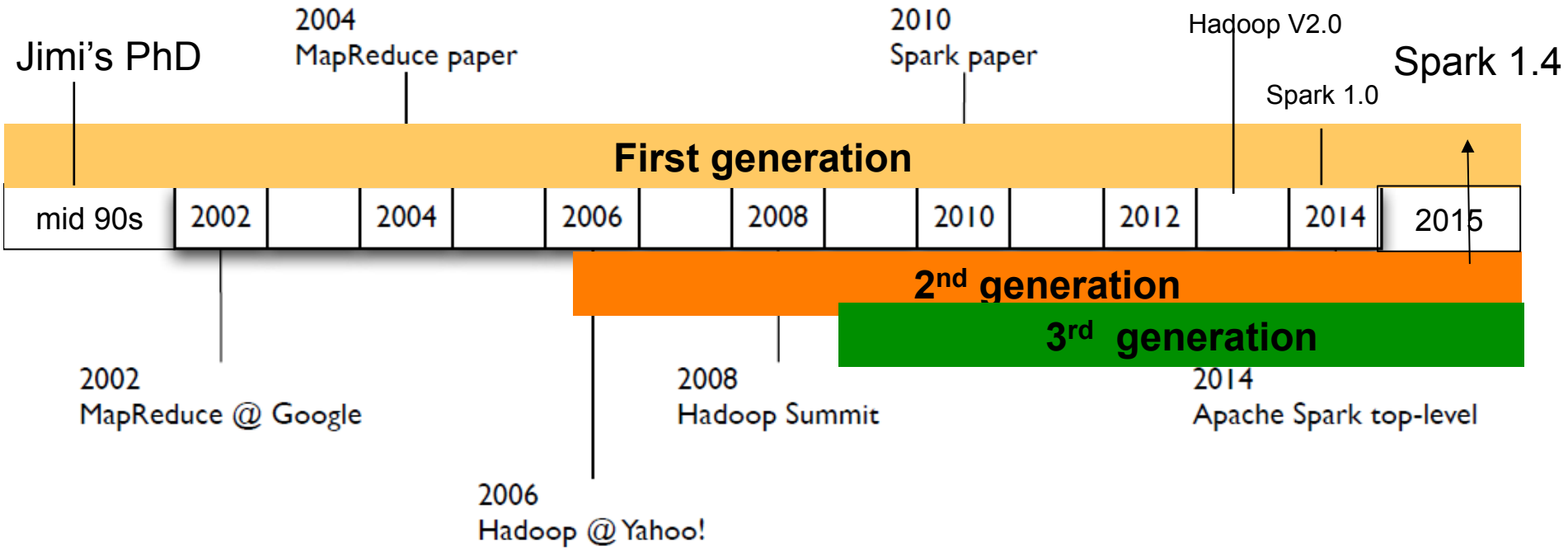
Datasets and Case studies

- **Google ngram corpus (2.2 TB)**
- **WikiPedia graph (500 Gig)**
- **$100 Billion problem CTR prediction (20-30 Gig)**
- **Web attack (severely sparse)**
- ***1 Million Song problem***
- ***WSDM 2016 Cup (37 Gig)***
  - *assess the query-independent importance of scholarly articles)*

Three generations of machine learning

- **First generation: dataset that fits in memory**
  - Single node learning summary statistics and some batch modeling (at small scale); SQL, R
  - Down sampling the data
- **Second generation: General purpose clusters and frameworks**
  - Distributed frameworks that allows us to divide and conquer problems
  - Learning using general purpose frameworks such as hadoop big data analysis offline, realtime decision making, homegrown specialist systems (Hadoop for analysis and modeling;), Hadoop, R
  - In-house purpose built systems; specialist sport
- **Third generation: Purpose-built libraries and frameworks**
  - Built for iterative algorithms that are common place in ML
  - huge scale realtime analysis and decision making systems
  - Specialized frameworks for large scale manipulation the type of data you are workign with.
  - For example, Machine learning libraries like MLLib in Spark, graph processing libraries like Apache Giraph or GraphX in Spark

Evolution of Map-Reduce frameworks for big data processing

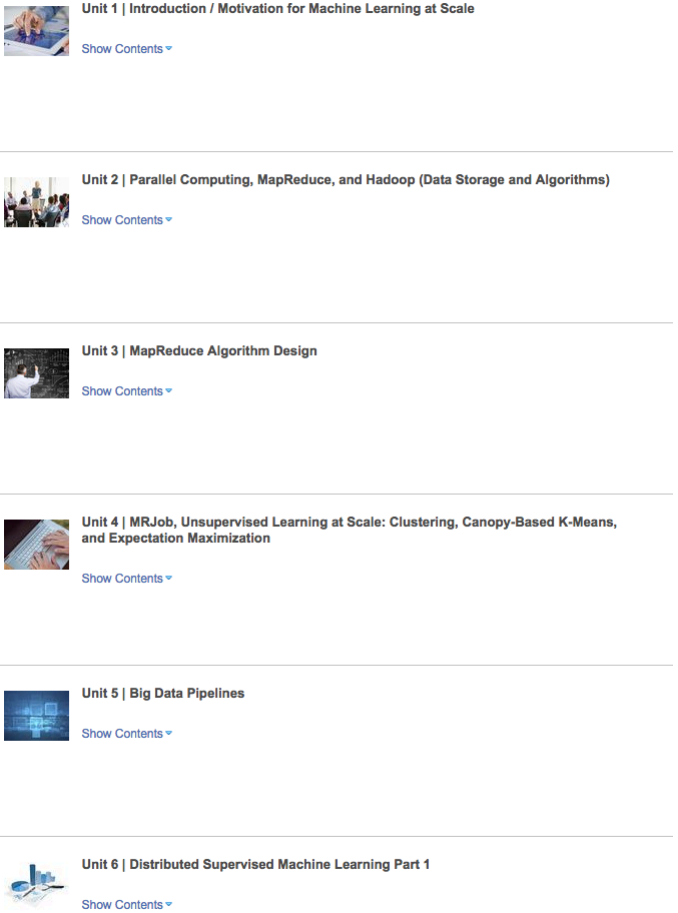


Scale, Real-world case studies, OpenSource

Syllabus

https://ucb-mids-cms-prod.2u.com/node/17978
https://learn.datascience.berkeley.edu/course/view.php?id=55&group=0&page=coursework

Syllabus
https://www.dropbox.com/s/al9y7ckynn5wnxg/DATASCI%20W261%20syllabus_Rev%20-%20Published%20-2015-08-19%20-live.pdf?dl=0



Unit 1 | Introduction and Motivation for Machine Learning at Scale

Show Contents

Unit 2 | Parallel Computing, MapReduce, and Hadoop (Data Storage and Algorithms)

Show Contents

Unit 3 | MapReduce Algorithm Design

Show Contents

Unit 4 | MRJob, Unsupervised Learning at Scale: Clustering, Canopy-Based K-Means, and Expectation Maximization

Show Contents

Unit 5 | Big Data Pipelines

Show Contents

Unit 6 | Distributed Supervised Machine Learning, Part 1

Show Contents

Unit 7 | Introduction to Graph Algorithms at Scale: Single Shortest Path Algorithm

Show Contents

Unit 8 | Midterm

Show Contents

Unit 9 | Large Scale Graph Processing: Random Walks, PageRank, and Personalized PageRank

Show Contents

Unit 10 | Spark: From Basics to Advanced

Show Contents

Unit 11 | Distributed Supervised Machine Learning, (in Spark) Part 2

Show Contents

Unit 12 | Decision Trees

Show Contents

Unit 13 | Predicting and Recognizing Links and Attributes in Social Networks

Show Contents

Unit 14 | Alternating Lease Squares and Various Optimizations in Spark/Working With Spark MLlib

Show Contents

Lecture: Deep Learning via Neural Networks at scale

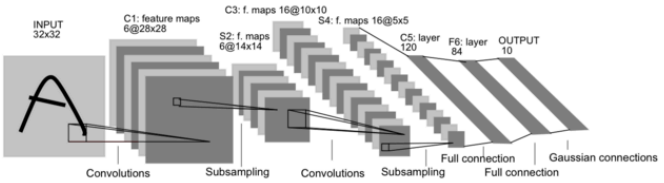
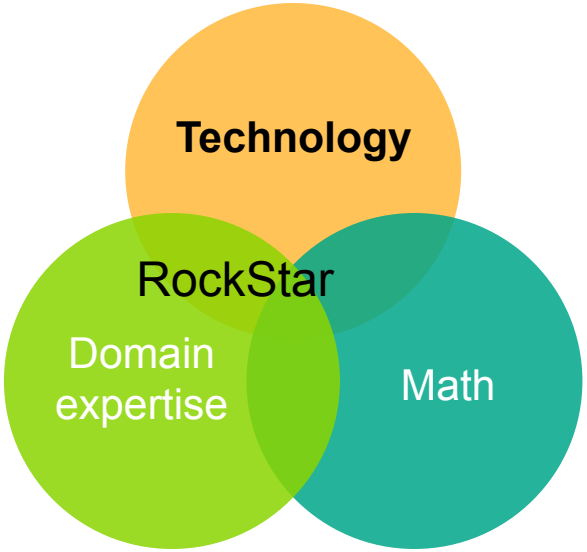
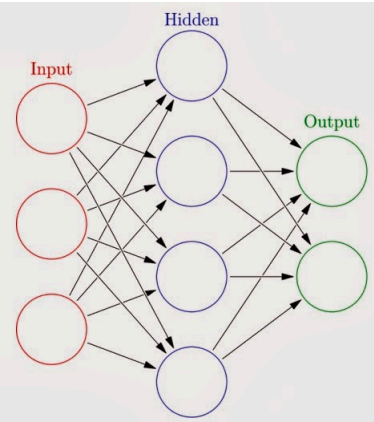


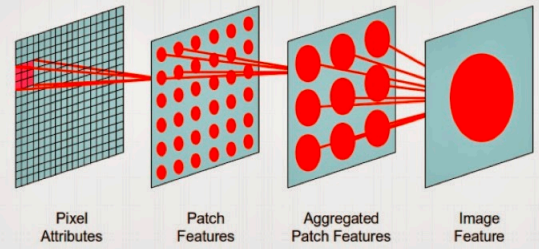
Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Deep Learning via Neural Networks at scale

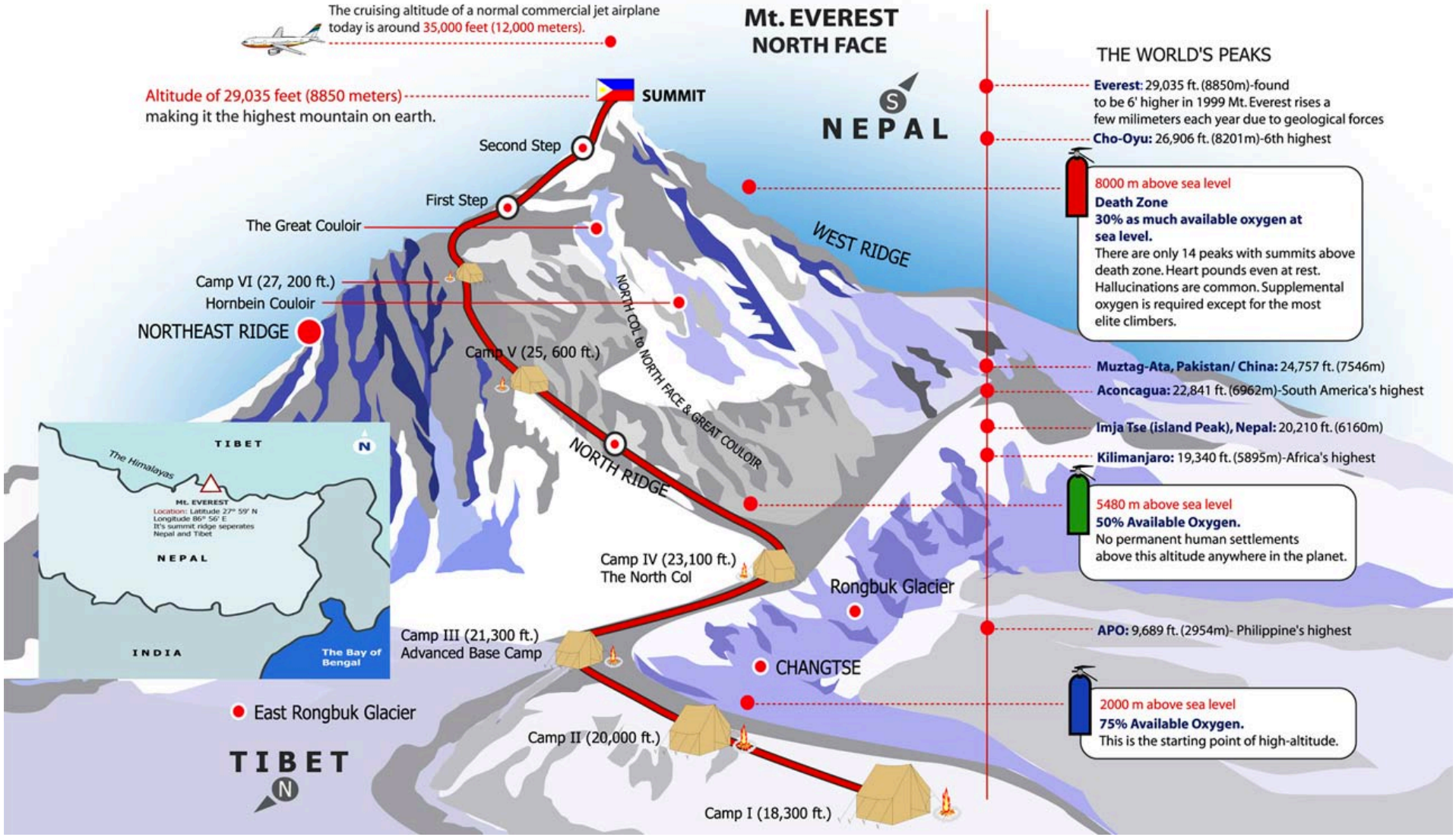
RockStars and Super Models Lecture





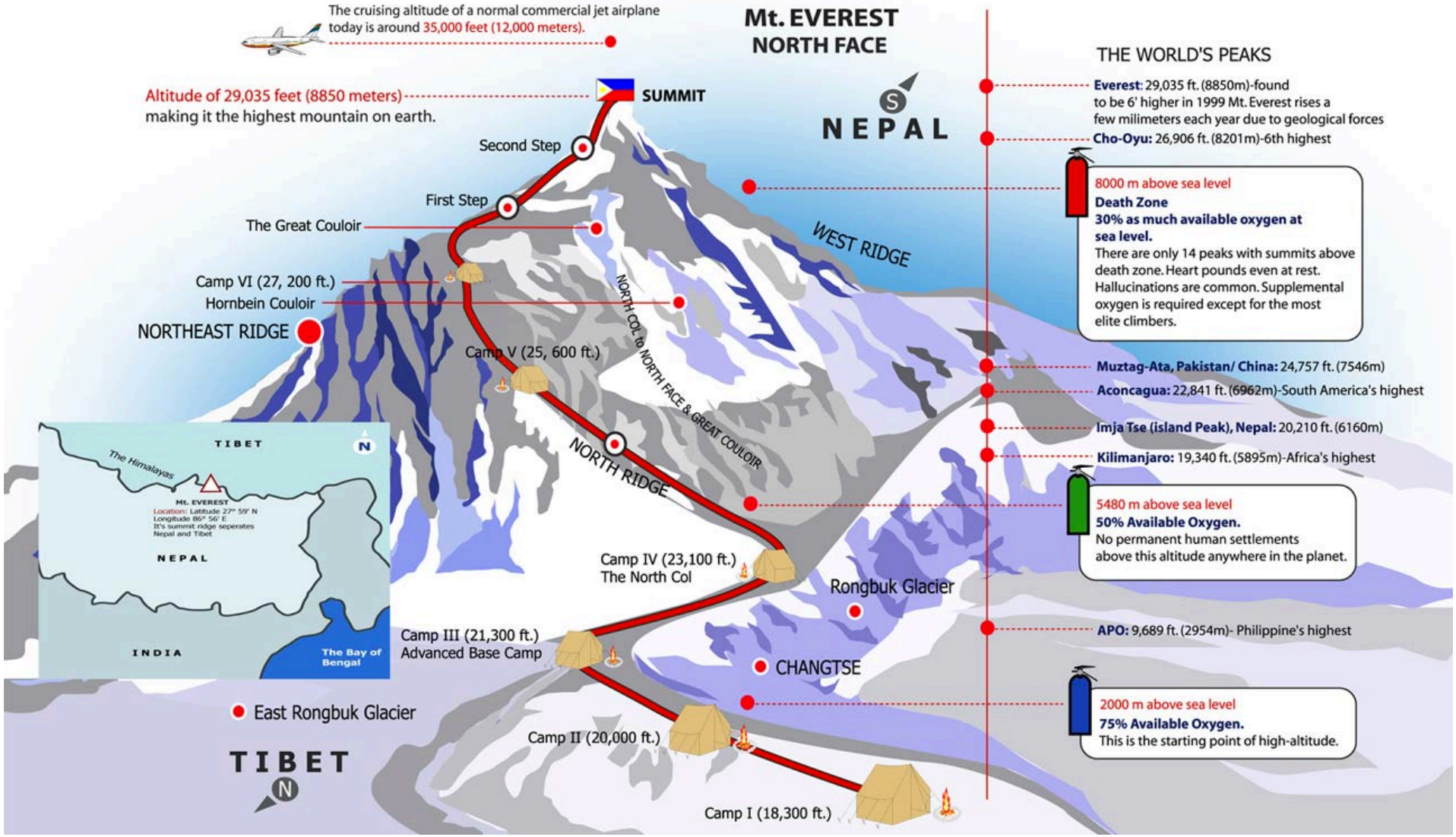


Week 9/10: Summited Mt Everest



After Week 10

……..you will be cruising at high octane levels



Machine Learning @ Scale: Type II Fun

- **Getting lost, getting cold, getting hungry, getting wet, getting scared, and coming out on top; that's the stuff you remember. That's Type II fun.**
- **Even if you've never heard about the fun scale, you will probably understand it pretty intuitively. On this highly scientific spectrum,**
  - Type I is the easy, fun-while-it's-happening stuff—mellow powder skiing, lazy cragging, afternoon hiking. You're bummed when it's over, but you'd be hard-pressed to remember more than a few specific examples.
  - Type II Tough going, character building and
  - Type III fun resides at the other end of the scale—miserable while it's happening, still miserable when it's over and just as miserable to think about later.
    - Anything that ends with you eating your own shoes, being evacuated by helicopter, or featuring prominently in a non-fiction bestseller likely classifies as Type III.

Fun: intellectual suntan

Fun?, Grueling, muscle memory

Total Disaster

- **- See more at: http://www.backcountry.com/explore/type-ii-fun#sthash.CZankeqe.dpuf**

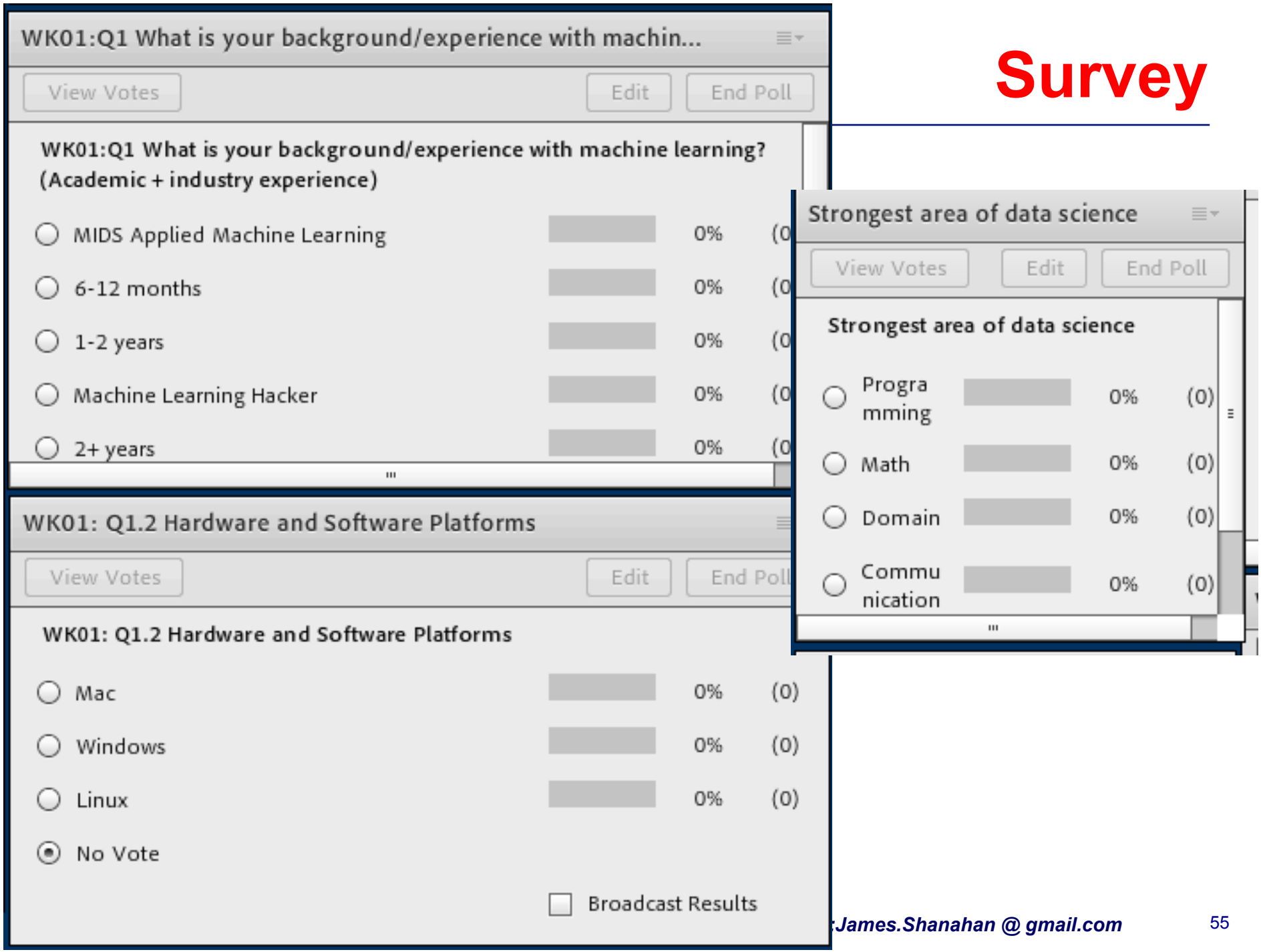
After this class you will …

- **Role**
  - Individual contributors: R&D, r&D, R&d, D
  - Managers/leaders
- **Focus**
  - Research
    - Continue to study and get a PhD on subject
    - Teach and shape future generations of data scientists
    - Theoretical and applied research
    - Applications
  - Development- infrastructure: Build infrastructure
  - Development
    - Architects of big data pipelines and large scale ML
    - Full stack people
    - Build Apps (on a fully supported framework)
    - Manage teams who this

This class will be demanding

- But it is a high ROI class
- Plan on spending 20 hours +/-10 hours per week on this class

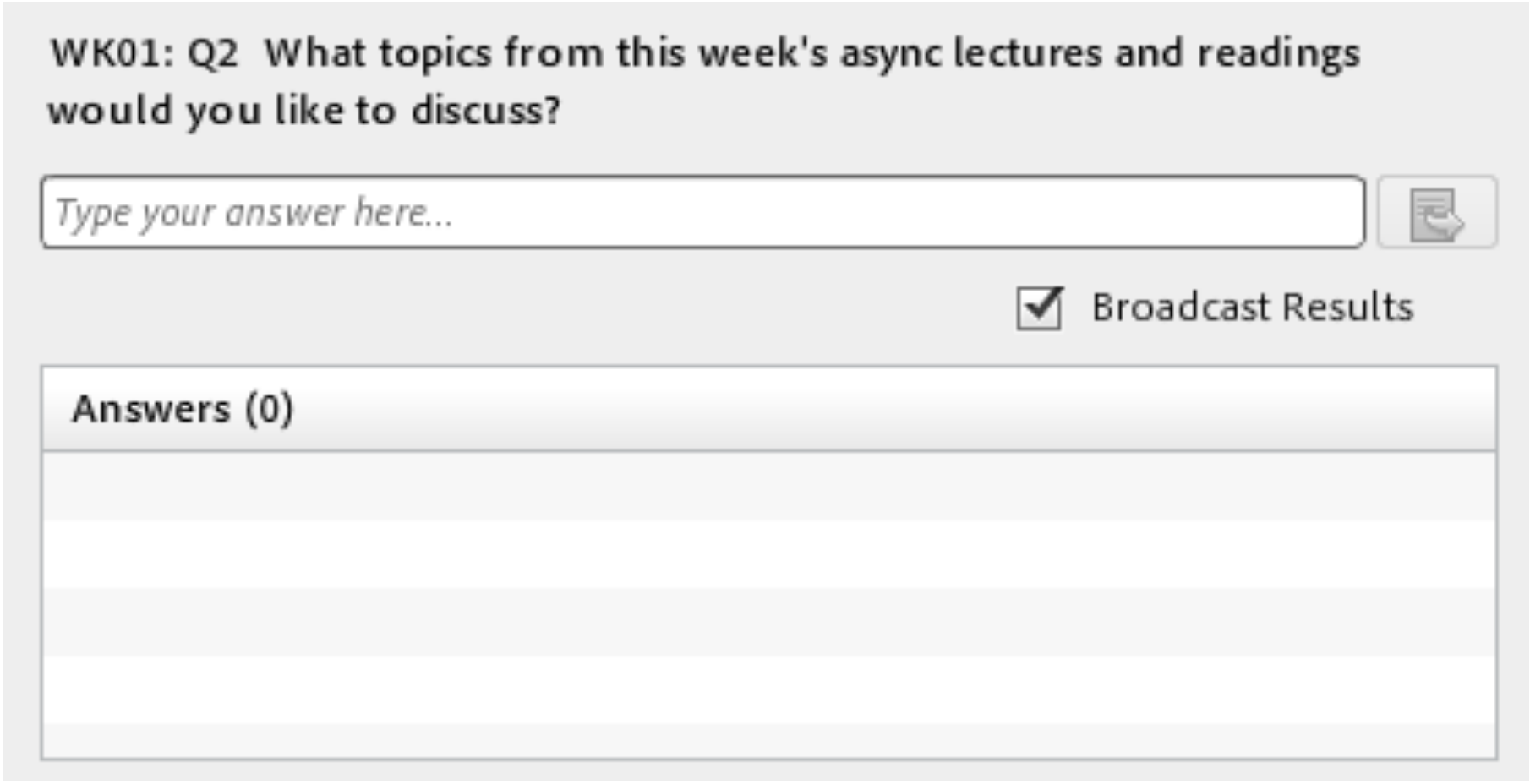
Survey

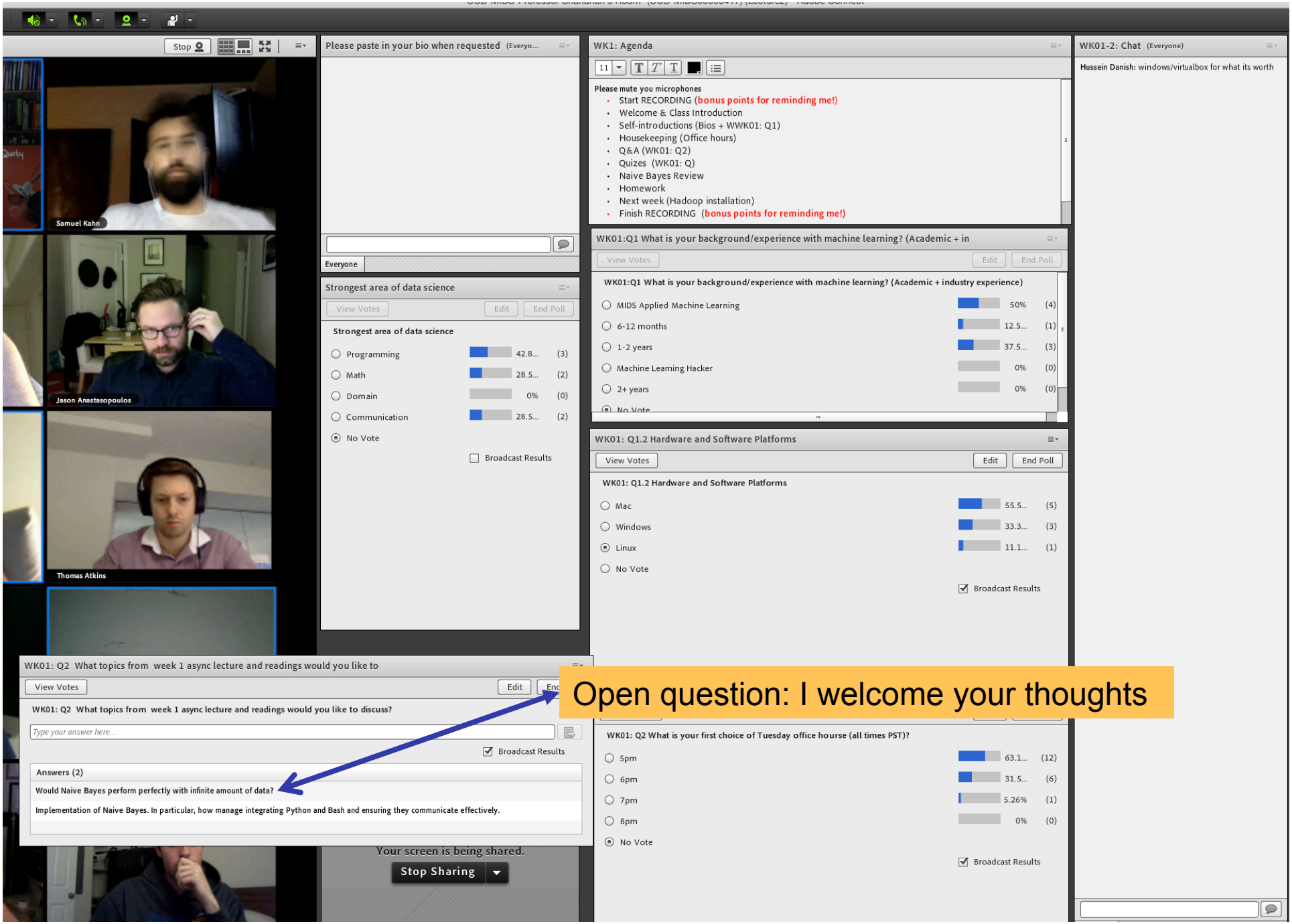

WK01:Q1 What is your background/experience with machin...
View Votes
Edit
End Poll
WK01:Q1 What is your background/experience with machine learning? (Academic + industry experience)
MIDS Applied Machine Learning 0% (0
6-12 months 0% (0
1-2 years 0% (0
Machine Learning Hacker 0% (0
2+ years 0% (0
Strongest area of data science
View Votes
Edit
End Poll
Strongest area of data science
Progra mming 0% (0)
Math 0% (0)
Domain 0% (0)
Commu nication 0% (0)
WK01: Q1.2 Hardware and Software Platforms
View Votes
Edit
End Poll
WK01: Q1.2 Hardware and Software Platforms
Mac 0% (0)
Windows 0% (0)
Linux 0% (0)
No Vote
Broadcast Results

Live Session Outline

- **Welcome & Class Introductions**
  - Please mute you microphones
  - Start RECORDING (bonus points for reminding me!)
  - Class, homework, project Logistics + Office hours
  - Self-introductions (Bios + WWK01: Q1)
- **Class Intro**
- **Q&A (WK01)**
- **Probability theory introduction**
- **Naive Bayes**
  - Basic derivation
  - Various Naïve Bayes Flavours (Live Session #2)
- **Wrapup**
  - Finish RECORDING (bonus points for reminding me!)
  - Click End Meeting

Questions on Unit 1?




Samuel Kahn
Jason Anastasopoulos
Thomas Atkins
Please paste in your bio when requested
Everyone
WK1: Agenda
Please mute you microphones
Start RECORDING (bonus points for reminding me!)
Welcome & Class Introduction
Self-introductions (Bios + WWK01: Q1)
Housekeeping (Office hours)
Q&A (WK01: Q2)
Quizes (WK01: Q)
Naive Bayes Review
Homework
Next week (Hadoop installation)
Finish RECORDING (bonus points for reminding me!)
WK01-2: Chat (Everyone)
Hussein Danish: windows/virtualbox for what its worth
Strongest area of data science
View Votes
Edit
End Poll
Strongest area of data science
Programming 42.8... (3)
Math 28.5... (2)
Domain 0% (0)
Communication 28.5... (2)
No Vote
Broadcast Results
WK01:Q1 What is your background/experience with machine learning? (Academic + in
WK01:Q1 What is your background/experience with machine learning? (Academic + industry experience)
MIDS Applied Machine Learning 50% (4)
6-12 months 12.5... (1)
1-2 years 37.5... (3)
Machine Learning Hacker 0% (0)
2+ years 0% (0)
No Vote
WK01: Q1.2 Hardware and Software Platforms
WK01: Q1.2 Hardware and Software Platforms
Mac 55.5... (5)
Windows 33.3... (3)
Linux 11.1... (1)
No Vote
Broadcast Results
WK01: Q2 What topics from week 1 async lecture and readings would you like to
WK01: Q2 What topics from week 1 async lecture and readings would you like to discuss?
Type your answer here...
Broadcast Results
Answers (2)
Would Naive Bayes perform perfectly with infinite amount of data?
Implementation of Naive Bayes. In particular, how manage integrating Python and Bash and ensuring they communicate effectively.
Open question: I welcome your thoughts
WK01: Q2 What is your first choice of Tuesday office hourse (all times PST)?
5pm 63.1... (12)
6pm 31.5... (6)
7pm 5.26% (1)
8pm 0% (0)
No Vote
Broadcast Results
Your screen is being shared.
Stop Sharing

Unit 1: Class Summary

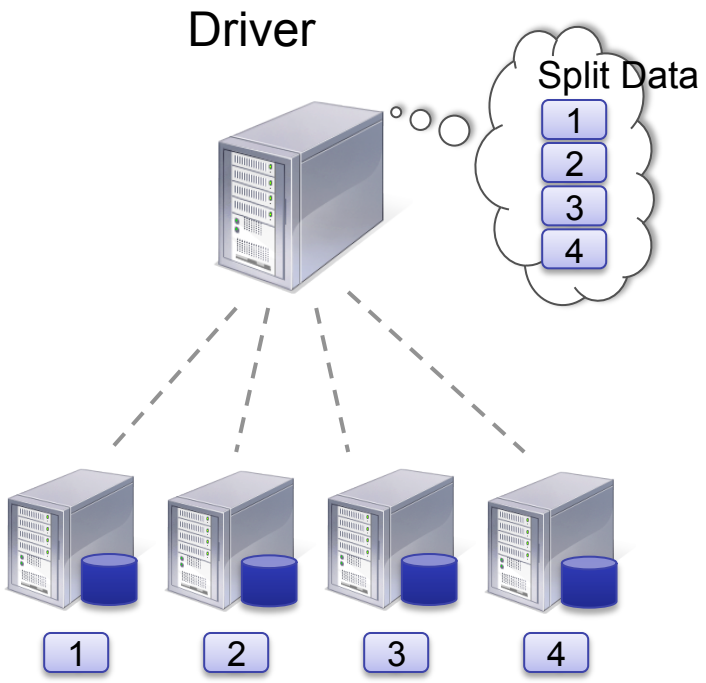
- **Big Data (what, why, where, who, how) [15]**
  - Big data definitions (5)
  - Sources of big data
    - Society (5)
    - Internet of things (5
- **Data Scientist [5]**
- **Data modeling pipeline(7 steps) [5]**
- **Large scale machine learning [10] (50 +**
  - More data or more scientists
- **Bias-variance as a means of understanding more data [15]**
  - Bias-variance background
  - BLT: Is more data better? [2]
  - Is more data better? [3] [45]
- **Large scale problem solving thru Divide and conquer [35]**
  - Poorman's map-reduce using command line (15)
  - BLT: Poorman's map-reduce using command line (10)
  - Solution Poorman's map-reduce using command line
  - SOLID foundation and approeciation for other Map-Reduce frameworks such as Hadoop and Spark

Command Line: Divide and Conquer

- **Partitions the data**
- **The framework processes the objects within a partition in sequence, and can processes multiple partitions in parallel**

Challenges?

Partition and distribute data



Slave Processors or Computers
Executors

Issues to be addressed

- How to break large problem into smaller problems? Decomposition for parallel processing
- How to assign tasks to workers distributed around the cluster?
- How do the workers get the data?
- How to synchronize among the workers?
- How to communicate with works?
- How to share partial results among workers?
- How to do all these in the presence of errors and hardware failures?
- MR is supported by a distributed file system that addresses many of these aspects.

Divide and conquer does not come for free: there are obligations in terms of communication, and synchronization

Machine Learning Challenges

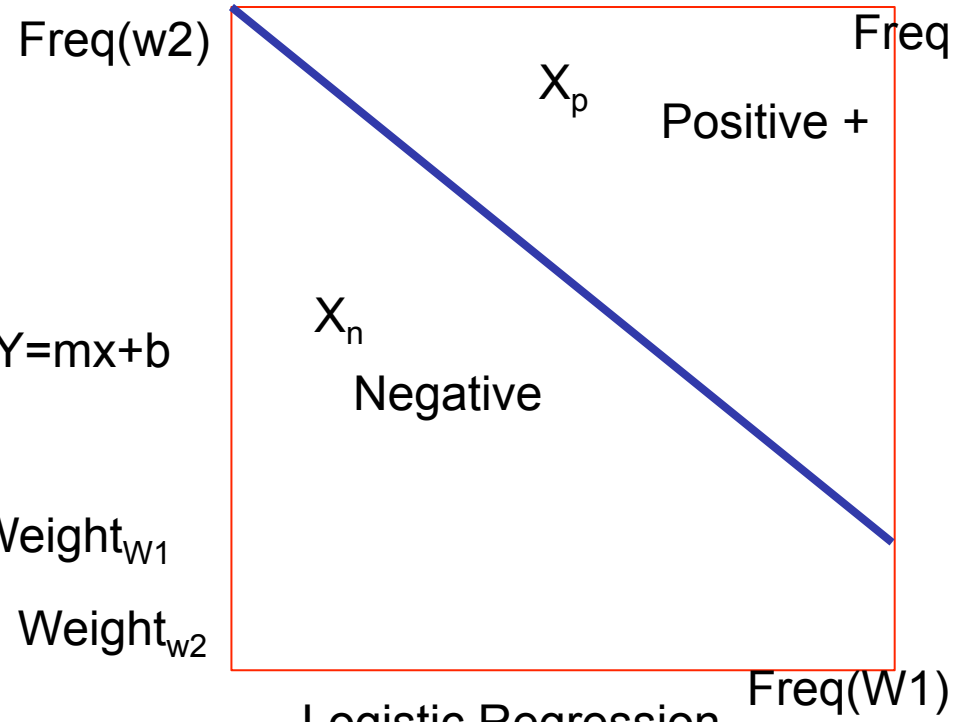
- **More data versus more data science**
- **Implementing ML to work on big data**
  - Naïve Bayes (today)

ML Objective: P(Data|Model)

Discriminative versus generative ML

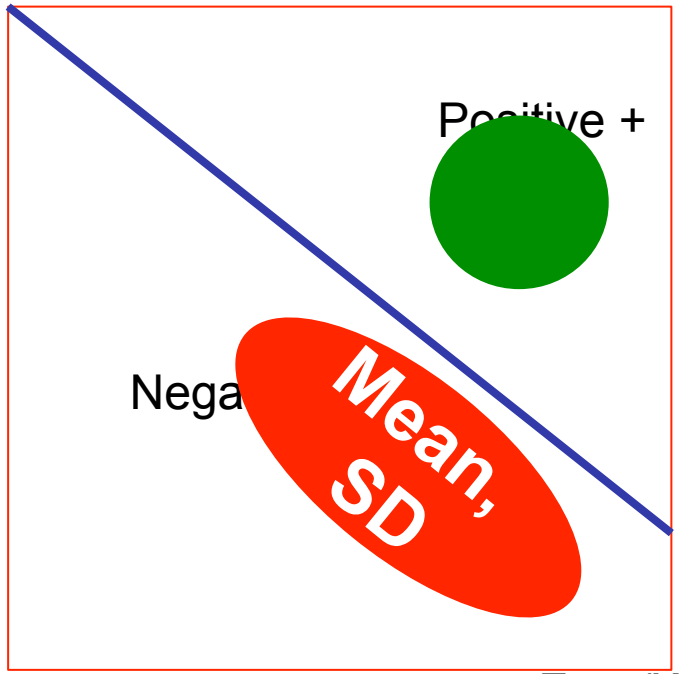
Target: Business versus nonBusiness: Inputs: W1, W2

Discriminative Model (LR, SVMs)



Logistic Regression

Generative Model



Best estimate for means, SD, var

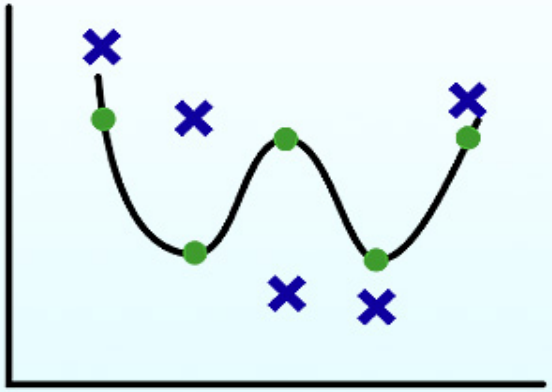
Naïve Bayes

Example	W1	w2	Class
1	1	4	negative
2	5	5	positive

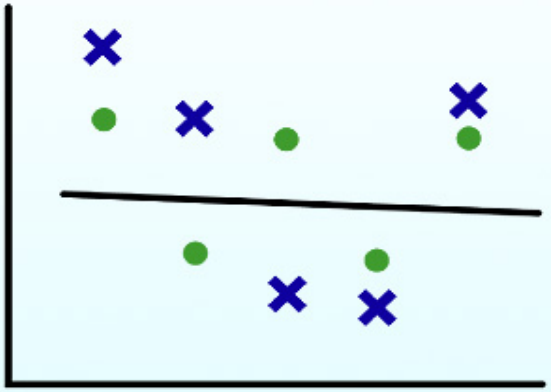
Naïve Bayes Classifier for Te

$P(Y=y_k)P(X_1,X$

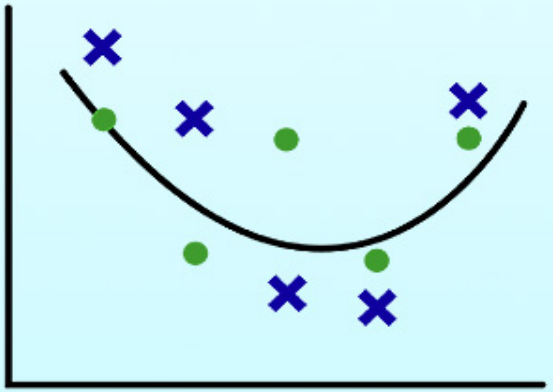
Bias-Variance Tradeoff in Model Selection in Simple Problem



(a) High variance/low bias.
4th-order polynomial ($p = 5$).

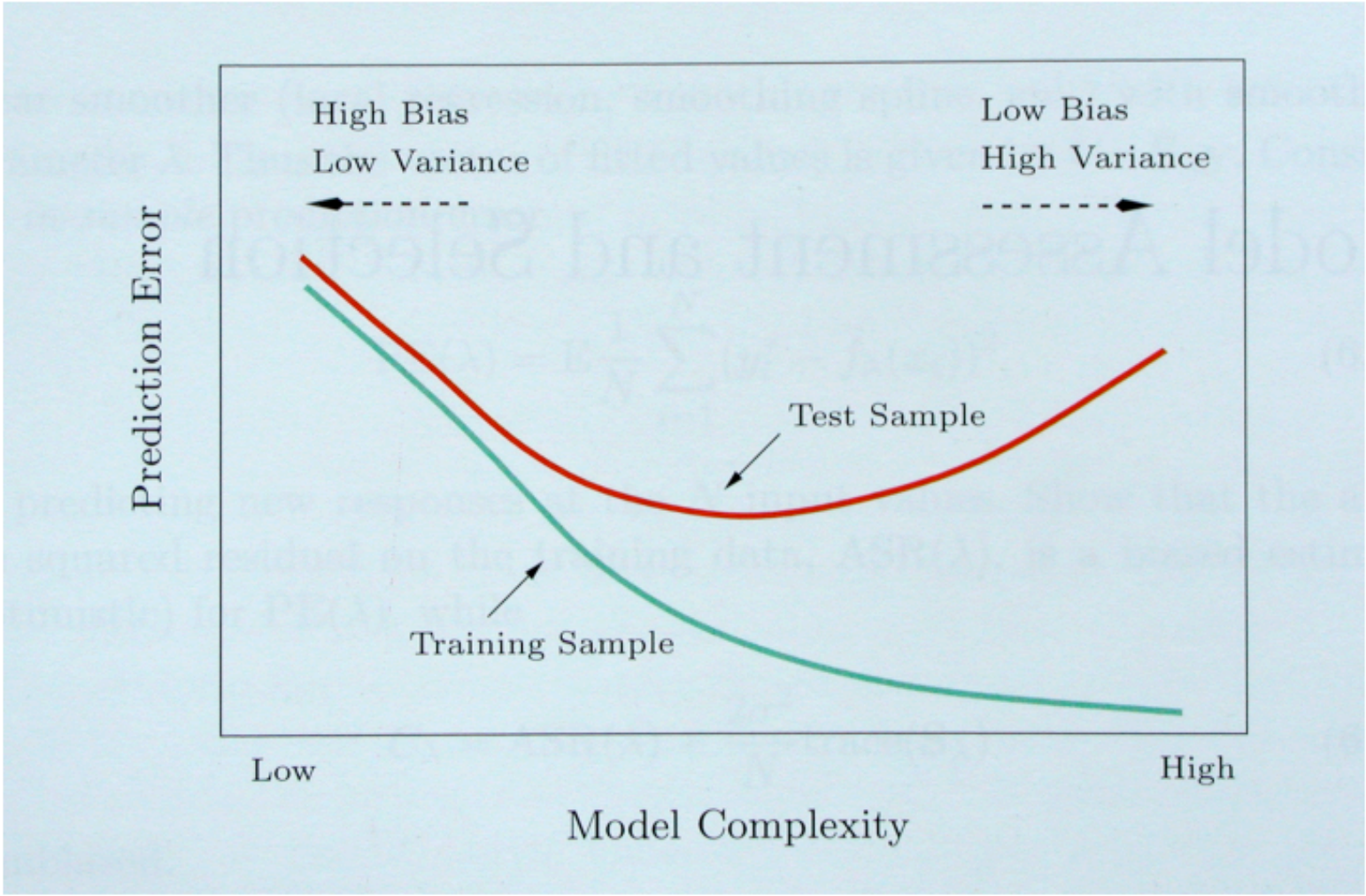


(b) Low variance/high bias.
1st-order polynomial ($p = 2$).



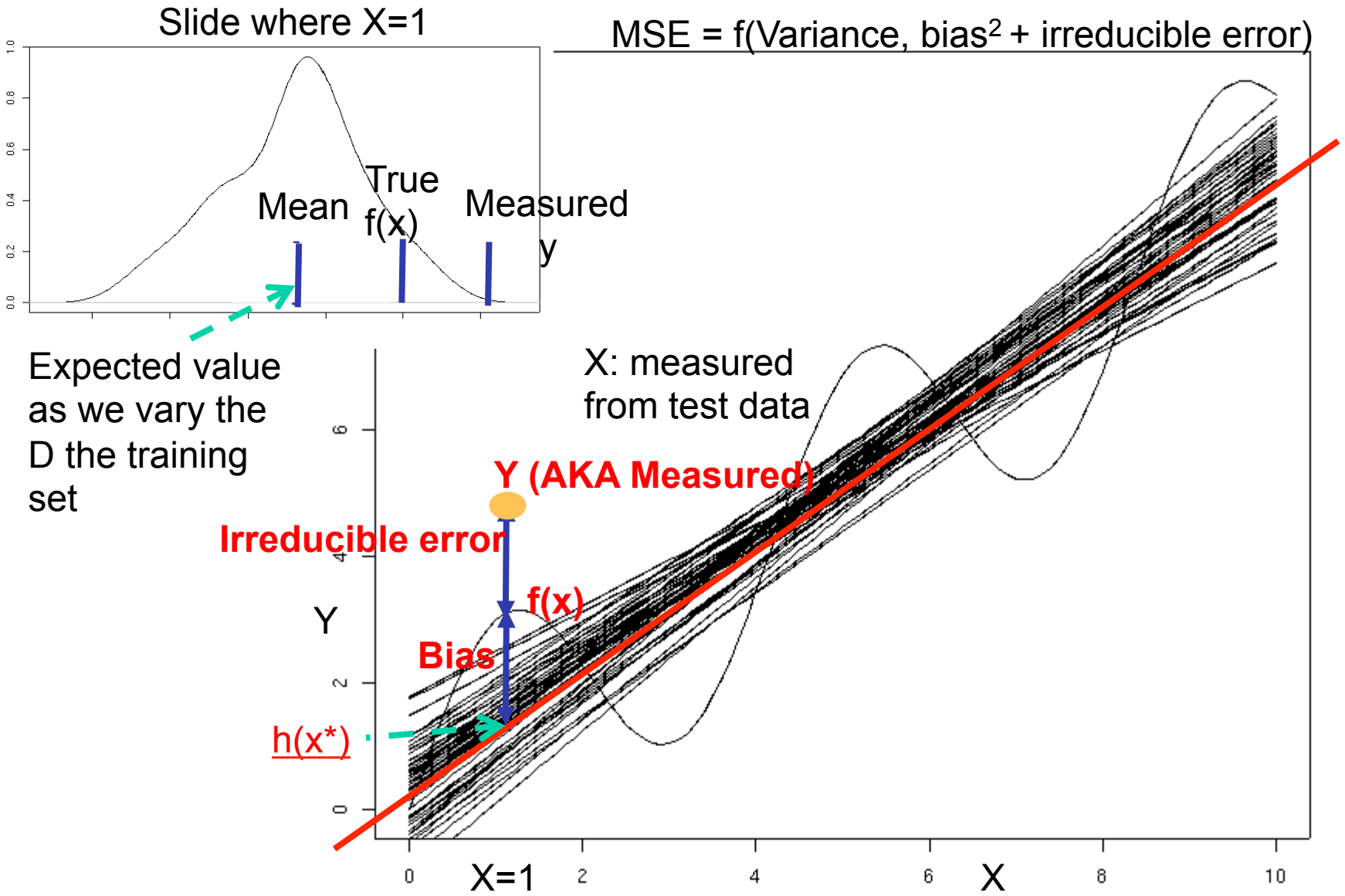
(c) Balanced variance & bias.
Minimum MSE.
2nd-order polynomial ($p = 3$).

Bias/Variance Tradeoff



Hastie, Tibshirani, Friedman "Elements of Statistical Learning" 2001

50 linear models (using different samples)



Bias-Variance Analysis

- Given a new data point **x**, what is the **expected prediction error?**
- Assume that the data points are drawn i.i.d. from *a unique underlying probability distribution P*
- The goal of the analysis is to compute, for an arbitrary new point **x**,

$$E_P\left[(y - h(\mathbf{x}))^2\right]$$

where y is the value of **x** that could be present in a data set, and the expectation is over *all training sets* drawn according to P.

- We will decompose this expectation into three components:

  bias, variance and noise

Bias-variance decomposition (2)

- Putting everything together, we have:

$$
\begin{aligned}
E_P\left[(y-h(\mathbf{x}))^2\right] &= E_P\left[(h(\mathbf{x})-\overline{h}(\mathbf{x}))^2\right] + \\
&\quad \overline{h}(\mathbf{x})^2 - 2f(\mathbf{x})\overline{h}(\mathbf{x}) + f(\mathbf{x})^2 + \\
&\quad E_P\left[(y-f(\mathbf{x}))^2\right] \\
&= E_P\left[(h(\mathbf{x})-\overline{h}(\mathbf{x}))^2\right] + \qquad \text{(variance)} \\
&\quad (h(\mathbf{x})-f(\mathbf{x}))^2 + \qquad \text{(bias)}^2 \\
&\quad E_P\left[(y-f(\mathbf{x}))^2\right] \qquad \text{(noise)} \\
&= \text{Var}[h(\mathbf{x})] + \text{Bias}[h(\mathbf{x})]^2 + E_P[\varepsilon^2] \\
&= \text{Var}[h(\mathbf{x})] + \text{Bias}[h(\mathbf{x})]^2 + \sigma^2
\end{aligned}
$$

- Expected prediction error = Variance + Bias^2 + Noise^2

Estimating Bias and Variance (continued)

- For each data point **x**, we will now have the observed corresponding value y and several predictions $y_1, \ldots, y_K$.
- Compute the average prediction h.
- Estimate **bias** as $(h - y)$
- Estimate **variance** as $\Sigma_k (y_k - h)^2/(K - 1)$
- Assume noise is 0

http://www-scf.usc.edu/~csci567/17-18-bias-variance.pdf

Bias, Variance, and Noise

- **Using a test data set with 20 data points**
- **For each data point x* in the test data set compute variance over the variance predictions (50 models give 50 predictions for each data point x*).**
- **For each data point x* calculate**
  - **Variance**: $E[(h(x^*) - \underline{h(x^*)})^2]$ ## $\Sigma(h(x^*) - \underline{h(x^*)})^2/50$
    - Describes how much h(x*) varies from one training set S to another
  - **Bias**: $[\underline{h(x^*)} - f(x^*)]$
    - Describes the <u>average</u> error of h(x*).
  - **Noise**: $E[(y^* - f(x^*))^2] = E[\varepsilon^2] = \sigma^2$
    - Describes how much y* varies from f(x*)

Bias-Variance written more formally for a single test point x*, using say 20 models

Using a test data set with 20 data points sum $(h(x^*) - y^*)^2$ over each of the 20 points and take the average

$$E_D[(h(x^*) - y^*)^2] =$$

Expected MSE wrt different models that are learned from different datasets D. E.g., 20 models yield 20 predictions $h_D(x^*)$

$$= E_D[(h_D(x^*) - \underline{h(x^*)})^2] + \textit{Variance} +$$

$$(\underline{h(x^*)} - f(x^*))^2 + \quad \textit{Bias}^2 +$$

$$E[(y^* - f(x^*))^2] \quad \textit{Noise}^2$$

$$= Var(h(x^*)) + Bias(h(x^*))^2 + E[\varepsilon^2]$$

$$= Var(h(x^*)) + Bias(h(x^*))^2 + \sigma^2$$

Expected prediction error = Variance + Bias^2 + Noise^2

Estimating Bias and Variance (continued)

- For each data point **x**, we will now have the observed corresponding value y and several predictions $y_1, \ldots, y_K$.
- Compute the average prediction h.
- Estimate **bias** as $(h - y)$
- Estimate **variance** as $\Sigma_k (y_k - h)^2/(K - 1)$
- Assume noise is 0

$h_D(x^*)$ model prediction (assume 20 training datesets)
h(x*) Average model prediction
f(x*) TRUE (Actual function value)
Y* Observed target data (noisy)

http://www-scf.usc.edu/~csci567/17-18-bias-variance.pdf
Excellent Slides from Sofus

Bias-variance trade-off

- Consider fitting a logistic regression LTU to a data set vs. fitting a large neural net.
- Which one do you expect to have higher bias? Higher variance?
- Typically, *bias* comes from not having good hypotheses in the considered class
- *Variance* results from the hypothesis class containing too many hypotheses
- Hence, we are faced with a *trade-off:* choose a more expressive class of hypotheses, which will generate higher variance, or a less expressive class, which will generate higher bias.

Source of bias

- Inability to represent certain decision boundaries
  - E.g., linear threshold units, naïve Bayes, decision trees
- Incorrect assumptions
  - E.g, failure of independence assumption in naïve Bayes
- Classifiers that are "too global" (or, sometimes, too smooth)
  - E.g., a single linear separator, a small decision tree.

If the bias is high, the model is *underfitting* the data.

Source of variance

- Statistical sources
  - Classifiers that are "too local" and can easily fit the data
    - E.g., nearest neighbor, large decision trees
- Computational sources
  - Making decision based on small subsets of the data
    - E.g., decision tree splits near the leaves
  - Randomization in the learning algorithm
    - E.g., neural nets with random initial weights
  - Learning algorithms that make sharp decisions can be unstable (e.g. the decision boundary can change if one training example changes)

If the variance is high, the model is *overfitting* the data

Measuring Bias and Variance

- In practice (unlike in theory), we have only ONE training set S.
- We can simulate multiple training sets by bootstrap replicates
  - $S' = \{\mathbf{x} \mid \mathbf{x}$ is drawn at random with replacement from $S\}$ and $|S'| = |S|$.

Bias-Variance Tradeoff

Bias–variance decomposition of squared error [edit]

Suppose that we have a training set consisting of a set of points $x_1, \ldots, x_n$ and real values y_i associated with each point x_i. We assume that there is a functional, but noisy relation $y_i = f(x_i) + \epsilon$, where the noise, ϵ, has zero mean and variance σ^2.

We want to find a function $\hat{f}(x)$, that approximates the true function $y = f(x)$ as well as possible, by means of some learning algorithm. We make "as well as possible" precise by measuring the mean squared error between y and $\hat{f}(x)$: we want $(y - \hat{f}(x))^2$ to be minimal, both for $x_1, \ldots, x_n$ *and for points outside of our sample*. Of course, we cannot hope to do so perfectly, since the y_i contain noise ϵ; this means we must be prepared to accept an *irreducible error* in any function we come up with.

Finding an $\hat{f}$ that generalizes to points outside of the training set can be done with any of the countless algorithms used for supervised learning. It turns out that whichever function $\hat{f}$ we select, we can decompose its expected error on an unseen sample x as follows:[3]:34[4]:223

$$\mathrm{E}\left[(y - \hat{f}(x))^2\right] = \mathrm{Bias}\left[\hat{f}(x)\right]^2 + \mathrm{Var}\left[\hat{f}(x)\right] + \sigma^2$$

Where:

$$\mathrm{Bias}\left[\hat{f}(x)\right] = \mathrm{E}\left[\hat{f}(x)\right] - f(x)$$

and

$$\mathrm{Var}\left[\hat{f}(x)\right] = \mathrm{E}\left[\left(\hat{f}(x) - \mathrm{E}[\hat{f}(x)]\right)^2\right]$$

The expectation ranges over different choices of the training set $x_1, \ldots, x_n, y_1, \ldots, y_n$, all sampled from the same distribution. The three terms represent:

- the square of the *bias* of the learning method, which can be thought of the error caused by the simplifying assumptions built into the method. E.g., when approximating a non-linear function $f(x)$ using a learning method for linear models, there will be error in the estimates $\hat{f}(x)$ due to this assumption;
- the *variance* of the learning method, or, intuitively, how much the learning method $\hat{f}(x)$ will move around its mean;
- the irreducible error σ^2. Since all three terms are non-negative, this forms a lower bound on the expected error on unseen samples.[3]:34

The more complex the model $\hat{f}(x)$ is, the more data points it will capture, and the lower the bias will be. However, complexity will make the model "move" more to capture the data points, and hence its variance will be larger.

using radial basis functions (blue). Several trials are shown in each graph. For each trial, a few noisy data points are provided as training set (top). For a wide spread (image 2) the bias is high: the RBFs cannot fully approximate the function (especially the central dip), but the variance between different trials is low. As spread decreases (image 3 and 4) the bias decreases: the blue curves more closely approximate the red. However, depending on the noise in different trials the variance between trials increases. In the lowermost image the approximated values for x=0 varies wildly depending on where the data points were located.

https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

https://theclevermachine.wordpress.com/2013/04/21/model-selection-underfitting-overfitting-and-the-bias-variance-tradeoff/

Bias-Variance
Order 1, 2,3 polynomial

xGrid = linspace(-1,1,100);

```
% FIT MODELS TO K INDEPENDENT DATASETS
K = 50;
for iS = 1:K
    ySim = f(x) + noiseSTD*randn(size(x));
    for jD = 1:numel(degree)
        % FIT THE MODEL USING polyfit.m
        thetaTmp = polyfit(x,ySim,degree(jD));
        % EVALUATE THE MODEL FIT USING polyval.m
        simFit{jD}(iS,:) = polyval(thetaTmp,xGrid);
    end
end

% DISPLAY ALL THE MODEL FITS
h = [];
for iD = 1:numel(degree)
    figure(iD+1)
    hold on
    % PLOT THE FUNCTION FIT TO EACH DATASET
    for iS = 1:K
        h(1) = plot(xGrid,simFit{iD}(iS,:),'color',brighten(cols(iD,:),.6));
    end
    % PLOT THE AVERAGE FUNCTION ACROSS ALL FITS
    h(2) = plot(xGrid,mean(simFit{iD}),'color',cols(iD,:),'Linewidth',5);
    % PLOT THE UNDERLYING FUNCTION f(x)
    h(3) = plot(xGrid,f(xGrid),'color','k','Linewidth',3);
    % CALCULATE THE SQUARED ERROR AT EACH POINT, AVERAGED ACROSS ALL DATASETS
    squaredError = (mean(simFit{iD})-f(xGrid)).^2;
    % PLOT THE SQUARED ERROR
    h(4) = plot(xGrid,squaredError,'k--','Linewidth',3);
    uistack(h(2),'top')
    hold off
    axis square
    xlim([-1 1])
    ylim([-1 1])
    legend(h,{sprintf('Individual g_{%d}(x)',degree(iD)),'Mean of All Fits','f(x)','Squared Error'},'Location','WestOu
    title(sprintf('Model Order=%d',degree(iD)))
end
```

For polynomial =iD

iS Dataset Sample

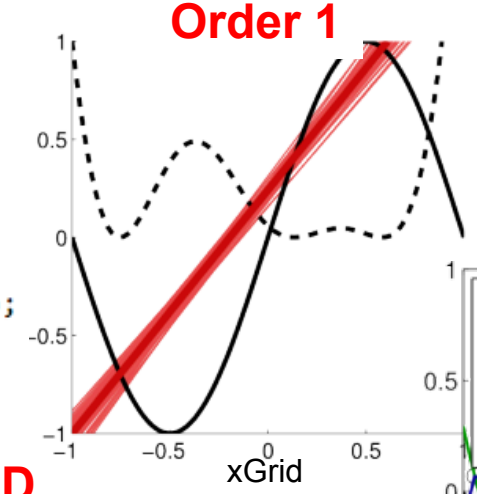
Predictions for sample set iS using model iD

Avg predictions of xGrid

BIAS

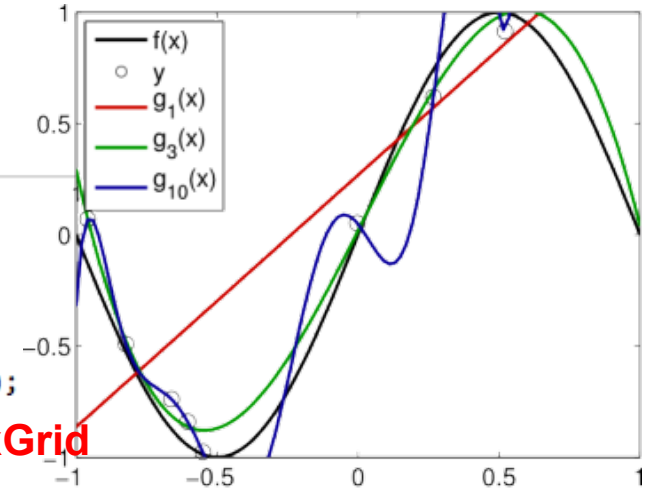
True error

Order 1



f(x) = sin(pi X x)

Order 1, 2, 3 with data

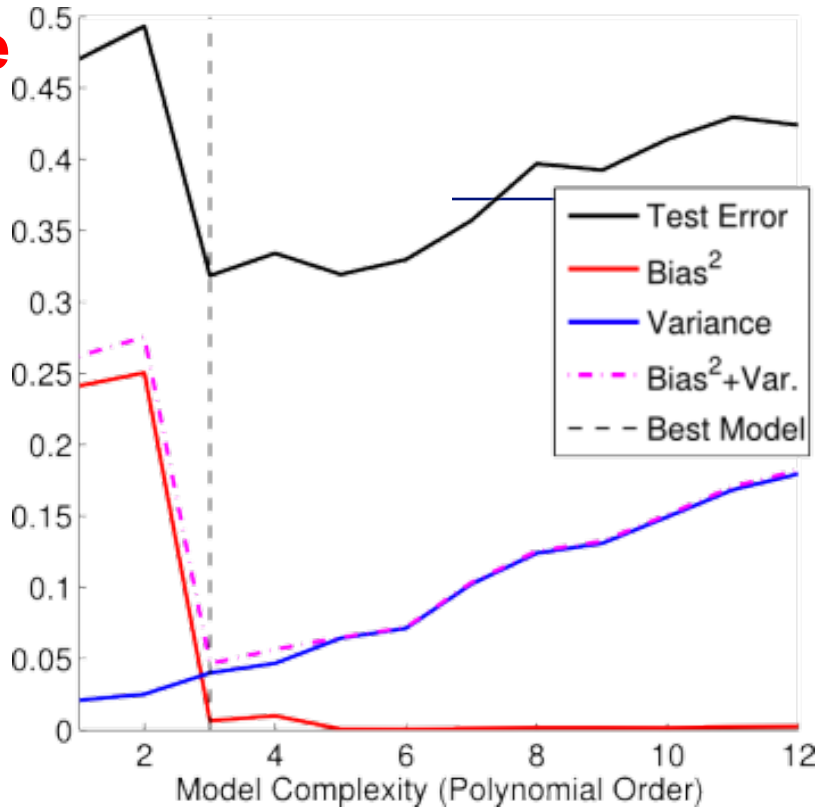


F(x) = true functi
ySim = noisy dat
simFit matrix of pr

Our original goal was to approximate f(x), not the data points per se.

Bias-Variance Order [1-12] polynomials

```
% # INITIALIZE SOME VARIABLES
xGrid = linspace(-1,1,N);
meanPrediction = zeros(K,N);
thetaHat = {};
x = linspace(-1,1,N);
x = x(randperm(N));
for iS = 1:K % LOOP OVER DATASETS
    % CREATE OBSERVED DATA, y
    y = f(x) + noiseSTD*randn(size(x));

    % CREATE TRAINING SET
    xTrain = x(1:nTrain);
    yTrain = y(1:nTrain);

    % CREATE TESTING SET
    xTest = x(nTrain+1:end);
    yTest = y(nTrain+1:end);

    % FIT MODELS
    for jD = 1:nPolyMax

        % MODEL PARAMETER ESTIMATES
        thetaHat{jD}(iS,:) = polyfit(xTrain,yTrain,jD);

        % PREDICTIONS
        yHatTrain{jD}(iS,:) = polyval([thetaHat{jD}(iS,:)],xTrain); TRAINING SET
        yHatTest{jD}(iS,:) = polyval([thetaHat{jD}(iS,:)],xTest);% TESTING SET

        % MEAN SQUARED ERROR
        trainErrors{jD}(iS) = mean((yHatTrain{jD}(iS,:) - yTrain).^2); % TRAINING
        testErrors{jD}(iS) = mean((yHatTest{jD}(iS,:) - yTest).^2); % TESTING
    end
end

% CALCULATE AVERAGE PREDICTION ERROR, BIAS, AND VARIANCE
for iD = 1:nPolyMax
    trainError(iD) = mean(trainErrors{iD});
    testError(iD) = mean(testErrors{iD});
    biasSquared(iD) = mean((mean(yHatTest{iD})-f(xTest)).^2);
    variance(iD) = mean(var(yHatTest{iD},1));
end
[~,bestModel] = min(testError);
```



Test Error = Variance(x_i)+ Bias(x_i)+irreducibleError

Avg(Bias (x_i))
Avg(Variance (x_i))

Best least squares fit for monomials of degree 1 to n.

- p = polyfit(x,y,n) returns the coefficients for a polynomial p(x) of degree n that is a best fit (in a least-squares sense) for the data in y. The coefficients in p are in descending powers, and the length of p is n+1

polyfit

Polynomial curve

Syntax

```
p = polyfit(x,y,n)
[p,S] = polyfit(x,y,n)
[p,S,mu] = polyfit(x,y,n)
```

http://www.mathworks.com/help/matlab/ref/polyfit.html

Description

`p = polyfit(x,y,n)` returns the coefficients for a polynomial `p(x)` of degree n that is a best fit (in a least-squares sense) for the data in y. The coefficients in p are in descending powers, and the length of p is n+1

$$p(x) = p_1 x^n + p_2 x^{n-1} + \ldots + p_n x + p_{n+1}.$$

`[p,S] = polyfit(x,y,n)` also returns a structure S that can be used as an input to `polyval` to obtain error estimates.

`[p,S,mu] = polyfit(x,y,n)` also returns `mu`, which is a two-element vector with centering and scaling values. `mu(1)` is `mean(x)`, and `mu(2)` is `std(x)`. Using these values, `polyfit` centers x at zero and scales it to have unit standard deviation

$$\hat{x} = \frac{x - \bar{x}}{\sigma_x}.$$

This centering and scaling transformation improves the numerical properties of both the polynomial and the fitting algorithm.

Examples

▼ Fit Polynomial to Trigonometric Function

Generate 10 points equally spaced along a sine curve in the interval `[0,4*pi]`.

```
x = linspace(0,4*pi,10);
y = sin(x);
```

Use `polyfit` to fit a 7th-degree polynomial to the po

```
p = polyfit(x,y,7);
```

Evaluate the polynomial on a finer grid and plot the re

```
x1 = linspace(0,4*pi);
y1 = polyval(p,x1);
figure
plot(x,y,'o')
hold on
plot(x1,y1)
hold off
```

```
> x1=0.1; x=c(x1^6, x1^5, x1^4,x1^3, x1^2, x1^1, 1)
> t(x) %*% ((c(0.0084,  -0.0983,    0.4217,   -0.7435,    0.1471,    1.1064,
0.00044117)))
          [,1]
[1,] 0.1118499
>
```

Fit a polynomial of degree 6

Determine the coefficients of the approximating polynomial of degree 6.

```
p = polyfit(x,y,6)
```

```
p =

    0.0084   -0.0983    0.4217   -0.7435    0.1471    1.1064    0.0004
```

To see how good the fit is, evaluate the polynomial at the data points and generate a table showing the data, fit, and error.

```
f = polyval(p,x);
T = table(x,y,f,y-f,'VariableNames',{'X','Y','Fit','FitError'})
```

```
T =

     X        Y         Fit         FitError
    ___    _______    _________    __________

      0          0    0.00044117    -0.00044117
    0.1    0.11246       0.11185     0.00060836
    0.2     0.2227       0.22231     0.00039189
    0.3    0.32863       0.32872    -9.7429e-05
    0.4    0.42839        0.4288    -0.0004066
    0.5     0.5205       0.52093    -0.0004256
    0.6    0.60386       0.60408    -0.0002282
    0.7     0.6778       0.67775     4.6383e-0
    0.8     0.7421       0.74183     0.0002699
    0.9    0.79691       0.79654     0.0003651
      1     0.8427       0.84238      0.000316
    1.1    0.88021       0.88005     0.0001594
    1.2    0.91031       0.91035    -3.9919e-0
    1.3    0.93401       0.93422      -0.00021
    1.4    0.95229       0.95258    -0.0002993
    1.5    0.96611       0.96639    -0.0002809
    1.6    0.97635       0.97652    -0.00016704
    1.7    0.98379       0.98379     8.3306e-07
    1.8    0.98909       0.98893     0.00016278
    1.9    0.99279       0.99253     0.00025791
      2    0.99532       0.99508     0.00024347
    2.1    0.99702       0.99691      0.0001131
    2.2    0.99814       0.99823    -8.8548e-05
    2.3    0.99886       0.99911    -0.00025673
    2.4    0.99931       0.99954    -0.00022451
    2.5    0.99959       0.99936     0.00023151
```

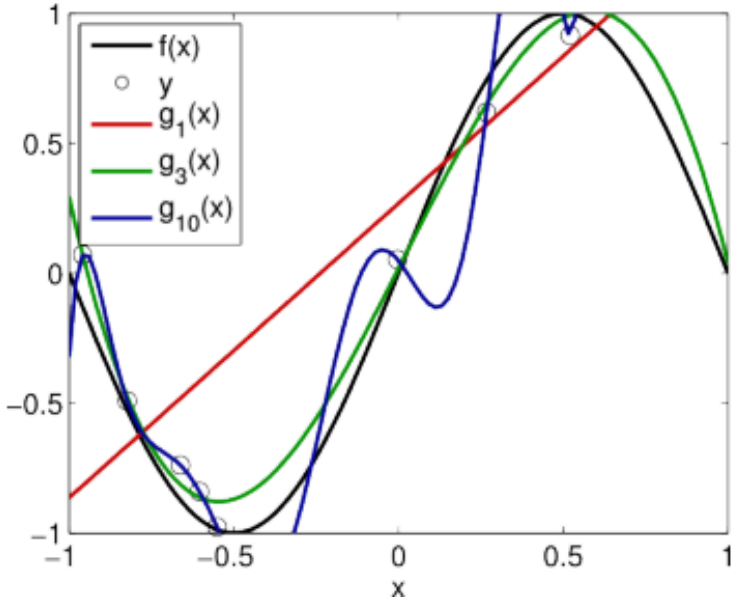
http://www.mathworks.com/help/matlab/ref/polyfit.html

```
> x1=0.1; x=c(x1^6, x1^5, x1^4,x1^3, x1^2, x1^1, 1)
> t(x) %*% ((c(0.0084,  -0.0983,    0.4217,   -0.7435,    0.1471,    1.1
0.00044117)))
         [,1]
[1,] 0.1118499
>
```

Fit a polynomial of degree 1, 3, 10

Below we estimate the parameters of three polynomial model functions of increasing complexity (using Matlab's `polyfit.m`) to the sampled data displayed above. Specifically, we estimate the functions $g_1(x)$, $g_3(x)$ and $g_{10}(x)$.

```
% FIT POLYNOMIAL MODELS & DISPLAY
% (ASSUMING PREVIOUS PLOT ABOVE STILL AVAILABLE)
degree = [1,3,10];
theta = {};
cols = [.8 .05 0.05; 0.05 .6 0.05; 0.05 0.05 .6];
for iD = 1:numel(degree)
    figure(1)
    theta{iD} = polyfit(x,y,degree(iD));
    fit{iD} = polyval(theta{iD},xGrid);
    h(end+1) = plot(xGrid,fit{iD},'color',cols(iD,:),'Linewidth',2);
    xlim([-1 1])
    ylim([-1 1])
end
legend(h,'f(x)','y','g_1(x)','g_3(x)','g_{10}(x)','Location','Northwest')
```



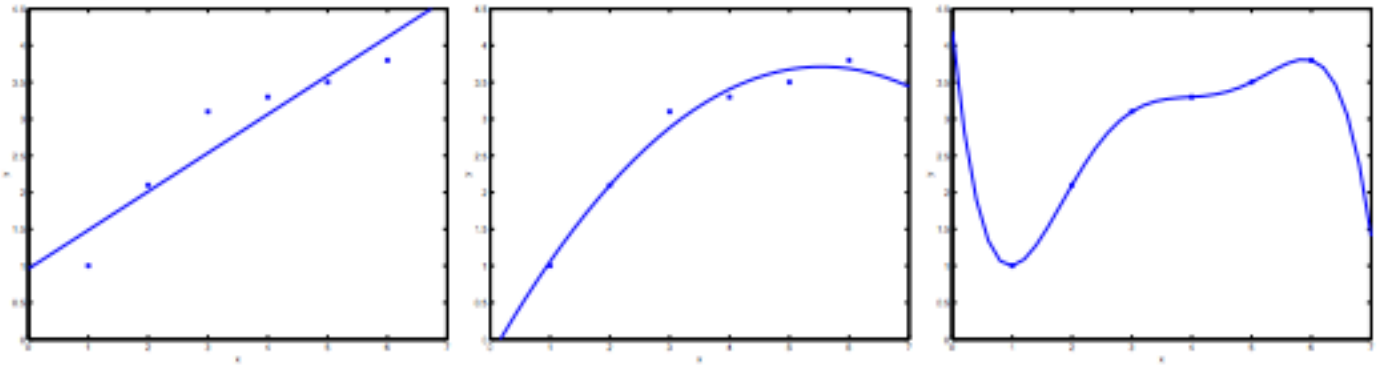
```
% FIT MODELS TO K INDEPENDENT DATASETS
K = 50;
for iS = 1:K
    ySim = f(x) + noiseSTD*randn(size(x));
    for jD = 1:numel(degree)
        % FIT THE MODEL USING polyfit.m
        thetaTmp = polyfit(x,ySim,degree(jD));
        % EVALUATE THE MODEL FIT USING polyval.m
        simFit{jD}(iS,:) = polyval(thetaTmp,xGrid);
    end
end

% DISPLAY ALL THE MODEL FITS
h = [];
for iD = 1:numel(degree)
    figure(iD+1)
    hold on
    % PLOT THE FUNCTION FIT TO EACH DATASET
    for iS = 1:K
        h(1) = plot(xGrid,simFit{iD}(iS,:),'color',brighten(cols(iD,:),.6));
    end
    % PLOT THE AVERAGE FUNCTION ACROSS ALL FITS
    h(2) = plot(xGrid,mean(simFit{iD}),'color',cols(iD,:),'Linewidth',5);
    % PLOT THE UNDERLYING FUNCTION f(x)
    h(3) = plot(xGrid,f(xGrid),'color','k','Linewidth',3);
    % CALCULATE THE SQUARED ERROR AT EACH POINT, AVERAGED ACROSS ALL DATASETS
    squaredError = (mean(simFit{iD})-f(xGrid)).^2;
    % PLOT THE SQUARED ERROR
    h(4) = plot(xGrid,squaredError,'k--','Linewidth',3);
    uistack(h(2),'top')
    hold off
    axis square
    xlim([-1 1])
    ylim([-1 1])
    legend(h,{sprintf('Individual g_{%d}(x)',degree(iD)),'Mean of All Fits','f(x)','Squared Error'},'Location','West
    title(sprintf('Model Order=%d',degree(iD)))
end
```

Review and Conclusions

- **For regression problems (squared error loss), the expected error rate can be decomposed into**
  - $\text{Bias}(x^*)^2 + \text{Variance}(x^*) + \text{Noise}(x^*)$
- **For classification problems (0/1 loss), the expected error rate depends on whether bias is present:**
  - if $B(x^*) = 1$: $B(x^*) - [V(x^*) + N(x^*) - 2\, V(x^*)\, N(x^*)]$
  - if $B(x^*) = 0$: $B(x^*) + [V(x^*) + N(x^*) - 2\, V(x^*)\, N(x^*)]$
  - or $B(x^*) + Vu(x^*) - Vb(x^*)$ [ignoring noise]

1 Bias/variance tradeoff

When talking about linear regression, we discussed the problem of whether to fit a "simple" model such as the linear "$y = \theta_0 + \theta_1 x$," or a more "complex" model such as the polynomial "$y = \theta_0 + \theta_1 x + \cdots \theta_5 x^5$." We saw the following example:



Fitting a 5th order polynomial to the data (rightmost figure) did not result in a good model. Specifically, even though the 5th order polynomial did a very good job predicting y (say, prices of houses) from x (say, living area) for the examples in the training set, we do not expect the model shown to be a good one for predicting the prices of houses not in the training set. In other words, what's has been learned from the training set does not *generalize* well to other houses. The **generalization error** (which will be made formal shortly) of a hypothesis is its expected error on examples not necessarily in the training set.

Both the models in the leftmost and the rightmost figures above have large generalization error. However, the problems that the two models suffer from are very different. If the relationship between y and x is not linear,

Bias Variance Tradeoff

- **For more details:**
  - Lecture 1 from w261
  - http://www-scf.usc.edu/~csci567/17-18-bias-variance.pdf
  - http://www.eecs.berkeley.edu/~jegonzal/talks/linear_regression.pdf

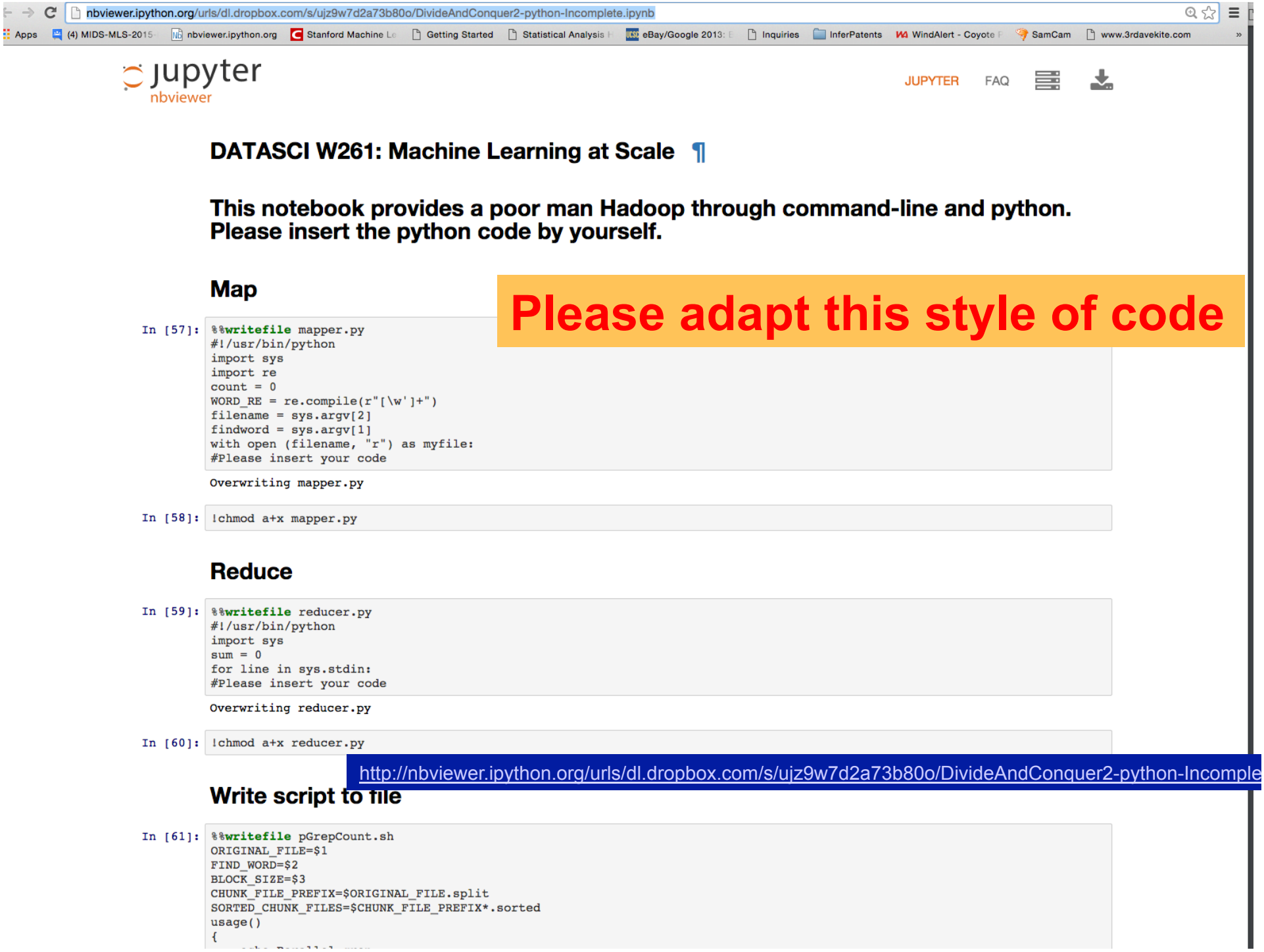
Live Session Outline

- **Welcome & Class Introductions**
  - Please mute you microphones
  - Start RECORDING (bonus points for reminding me!)
  - Self-introductions (Bios + WWK01: Q1)
  - Class updates and Housekeeping (Office hours)
- **This week**
  - Q&A (WK01: Q2)
  - Quizzes (WK01: Q)
  - Naive Bayes Review
  - Homework for this week
- **Next week (Hadoop installation)**
- **Wrapup**
  - Finish RECORDING (bonus points for reminding me!)
  - Click End Meeting

WK01: Q1.10.1 Suppose you are facing a supervised learning problem and have a very large data set (m=100,000,000). How can you tell if using all the data is likely to perform much better than using a small subset of the data say, (m=1,000)? As a data scientist, what would you do?

- ◯ There is no need to verify this; using a larger dataset always gives much better performance. 0% (0)
- ◯ Plot Jtrain(θ) as a function of the number of iterations of the optimization algorithm (such as gradient descent). Note: Where Jtrain(θ) denotes the error over the training data. 0% (0)
- ◯ Plot a learning curve (Jtrain(θ) and JCV(θ) plotted as a a function of m) for some range of values of m (say up to m=1,000) and verify that the algorithm has bias when m is small. Note: Where Jtrain(θ) and JCV(θ) denotes the error over the training data and cross validation set respectively. 0% (0)
- Plot a learning curve for a range of values of

- **Quiz 2: fill in the mapper and reducer code**
- **Any questions**


nbviewer.ipython.org/urls/dl.dropbox.com/s/ujz9w7d2a73b80o/DivideAndConquer2-python-Incomplete.ipynb
Apps
(4) MIDS-MLS-2015
nbviewer.ipython.org
Stanford Machine Le
Getting Started
Statistical Analysis H
eBay/Google 2013: E
Inquiries
InferPatents
WindAlert - Coyote F
SamCam
www.3rdavekite.com
jupyter
nbviewer
JUPYTER
FAQ
DATASCI W261: Machine Learning at Scale ¶
This notebook provides a poor man Hadoop through command-line and python.
Please insert the python code by yourself.
Map
Please adapt this style of code
In [57]:
%%writefile mapper.py
#!/usr/bin/python
import sys
import re
count = 0
WORD_RE = re.compile(r"[\w']+")
filename = sys.argv[2]
findword = sys.argv[1]
with open (filename, "r") as myfile:
#Please insert your code
Overwriting mapper.py
In [58]:
!chmod a+x mapper.py
Reduce
In [59]:
%%writefile reducer.py
#!/usr/bin/python
import sys
sum = 0
for line in sys.stdin:
#Please insert your code
Overwriting reducer.py
In [60]:
!chmod a+x reducer.py
http://nbviewer.ipython.org/urls/dl.dropbox.com/s/ujz9w7d2a73b80o/DivideAndConquer2-python-Incomple
Write script to file
In [61]:
%%writefile pGrepCount.sh
ORIGINAL_FILE=$1
FIND_WORD=$2
BLOCK_SIZE=$3
CHUNK_FILE_PREFIX=$ORIGINAL_FILE.split
SORTED_CHUNK_FILES=$CHUNK_FILE_PREFIX*.sorted
usage()
{

Live Session Outline

- **Welcome & Class Introductions**
  - Please mute you microphones
  - Start RECORDING (bonus points for reminding me!)
  - Class, homework, project Logistics + Office hours
  - Self-introductions (Bios + WWK01: Q1)
- **Class Intro**
- **Q&A (WK01)**
- **Probability theory introduction**
- **Naive Bayes**
  - Basic derivation
  - Various Naïve Bayes Flavours (Live Session #2)
- **Wrapup**
  - Finish RECORDING (bonus points for reminding me!)
  - Click End Meeting

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case)
- **Naïve Bayes (next live session)**
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
  - Continuos input variables (Cover later)
- **Case Study: Spam detector in Naïve Bayes**

Probability Basics

Probability theory 1/3

- **Probability theory is the branch of mathematics concerned with probability, the analysis of random phenomena.**
- **The central objects of probability theory are random variables, stochastic processes, and events:**
  - mathematical abstractions of non-deterministic events or measured quantities that may either be single occurrences or evolve over time in an apparently random fashion.

Probability theory 2/3

- **It is not possible to predict precisely results of random events.**
- **However, if a sequence of individual events, such as coin flipping or the roll of dice, is influenced by other factors, such as friction, it will exhibit certain patterns, which can be studied and predicted.**
- **Two representative mathematical results describing such patterns are the**
  - (1) law of large numbers (LLN) and the
  - (2) central limit theorem. (central tendencies)

  - In probability theory, the law of large numbers (LLN) is a theorem that describes the result of performing the same experiment a large number of times.
  - According to the law, the average of the results obtained from a large number of trials should be close to the expected value, and

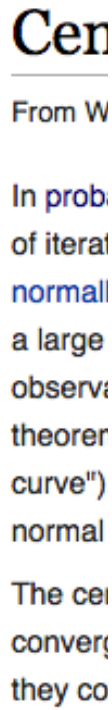
law of large numbers

- In probability theory, the law of large numbers (LLN) is a theorem that describes the result of performing the same experiment a large number of times.
- According to the law, the average of the results obtained from a large number of trials should be close to the expected value, and will tend to become closer as more trials are performed.

CLT

- In probability theory, the central limit theorem (CLT) states that, given certain conditions, the arithmetic mean of a sufficiently large number of iterates of independent random variables, each with a well-defined expected value and well-defined variance, will be approximately normally distributed, regardless of the underlying distribution.[1][2]

- To illustrate what this means, suppose that a sample is obtained containing a large number of observations, each observation being randomly generated in a way that does not depend on the values of the other observations, and that the arithmetic average of the observed values is computed.
- If this procedure is performed many times, the central limit theorem says that the computed values of the average will be distributed according to the normal distribution (commonly known as a "bell curve").
- A simple example of this is that if one flips a coin many times, the probability of getting a given number of heads should follow a norm curve, with mean equal to half the total number of flips.


Cen
From W
In prob
of iterat
normall
a large
observa
theorer
curve")
normal
The cer
converg
they co

Probability theory 3/3

- As a mathematical foundation for statistics, probability theory is essential to many human activities that involve quantitative analysis of large sets of data.
- Methods of probability theory also apply to descriptions of complex systems given only partial knowledge of their state, as in statistical mechanics.
- A great discovery of twentieth century physics was the probabilistic nature of physical phenomena at atomic scales, described in quantum mechanics.

Notation

- Proposition - statement or assertion about a state of the world
- Variable X is a set of mutually exclusive propositions x_i
- Variables – upper-case
- Propositions – lowercase
  - **Example (X=x, Y=y, Z=z)**
  - **Shortened: (x,y,z)**
- Sets of variables – bold
  - Example: (**X, Y, Z**)
- Latent/Hidden variable – states are inferred but never observed directly

From Axioms: deduce theorems and propositions

- One strategy in mathematics is to start with a few statements, then build up more mathematics from these statements.
- The beginning statements are known as axioms. An axiom is typically something that is mathematically self evident. From a relatively short list of axioms deductive logic is used to prove other statements, called theorems or propositions.
- The area of mathematics known as probability is no different.
- Underlying probability is a handful of axioms from which we can derive all sorts of results. But what are these probability axioms?
- Probability can be reduced to three axioms.
- It presupposes that we have a set of outcomes called the sample space S comprised of subsets called events E1, E2, . . ., En and a way of assigning a probability to any event E. The probability of the event E is denoted by P(E).

Axioms of Probabilities

http://statistics.about.com/od/Mathstat/a/what-is-the-power-set.htm

Axiom 1 : $0 \le P(A) \le 1$

Axiom 2: $P(Sure\ Proposition) = 1$

Axiom 3: $P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$

Marginal probability: $P(A) = P(A, B) + P(A, \neg B)$

$$P(A) = \sum_i P(A, B_i)$$

The first axiom of probability is that the probability of any event is a nonnegative real number. This means that the smallest that a probability can ever be is zero, and that it cannot be infinite.

The third axiom of probability deals with mutually exclusive events. If E1 and E2 are mutually exclusive, meaning that they have an empty intersection and we use U to denote the union, then P(E1 U E2) = P(E1) + P(E2).

Background

$$P(A) + P(\neg A) = 1$$

$$P(A \mid B)$$

Pr(A) → P(A|B)

- Belief in A under the assumption that B is known with absolute certainty
- A is conditioned on B

Probability Basics → Naïve Bayes Models

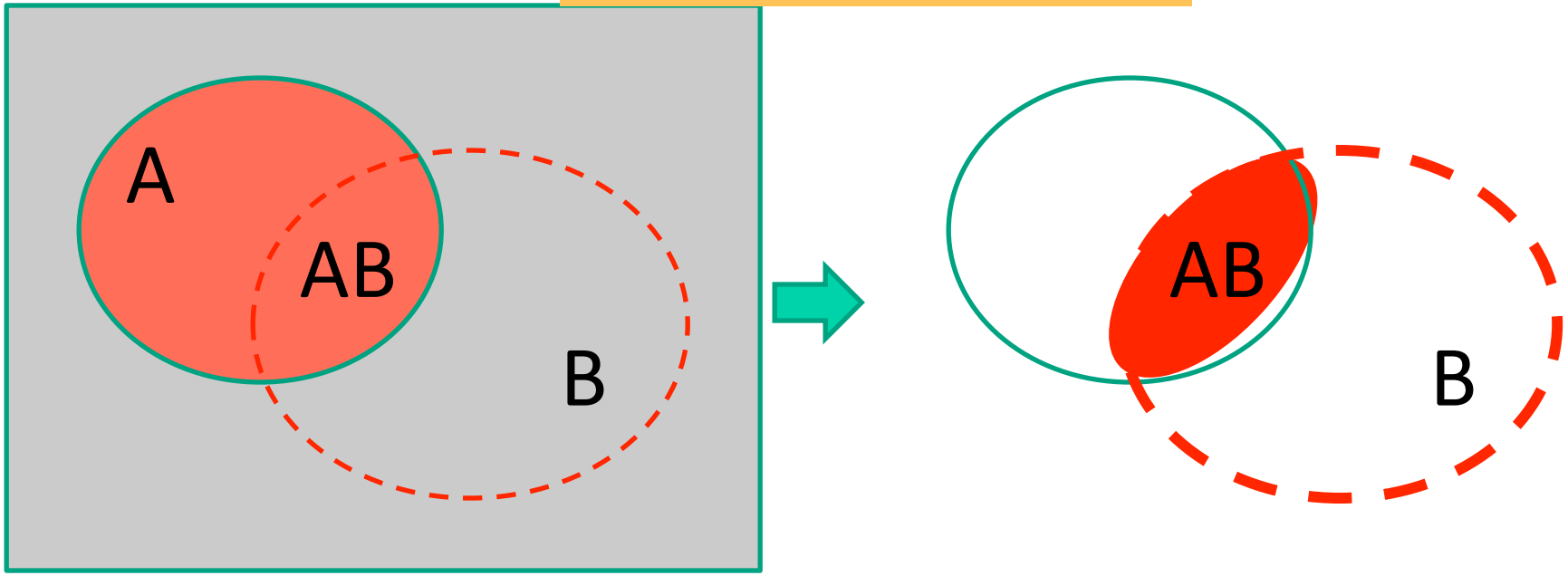
- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case)
- **Naïve Bayes (next live session)**
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
  - Continuos input variables (Cover later)
- **Case Study: Spam detector in Naïve Bayes**

Calculating Conditional Probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Conditional probability can be seen to be the probability with respect to a reduced sample space. We can illustrate the conditional probability with the Venn diagram.

Pr(A) → P(A|B)



Conditional Probabilities Examples

Example 1: If the probability that a research project will be well planned is 0.8 and the probability that is will be well planned and well executed is 0.72, what is the probability that a well planned research project will be well executed?

Answer: $0.72/0.8 = 0.9$. P[Exec|Plan]=P[Exec&Plan]/P[Plan]

Example 2: Roll a die twice. What is the probability that the total we get is 3? Given the information that the first number is 1, what is probability that the total we get is 3?

Answer: 2/36 and 1/6.

2 of 6x6 = 36 equilikely events: 1+2, 2+1

Since 1st number =1, 2nd number must =2 to get a sum of 3. There are 6 possible 2nd outcomes, 1 of which is to get a 2, hence 1/6

Conditional probability as an easier path to an answer

Sometimes the conditional probability can be determined easily, so we can actually use the conditional probability to calculate probability.

The multiplication rule: $P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$.

Example: There are 3 red balls and 2 blue balls in a box. Randomly take 2 balls from the box. What is the probability that both are red?

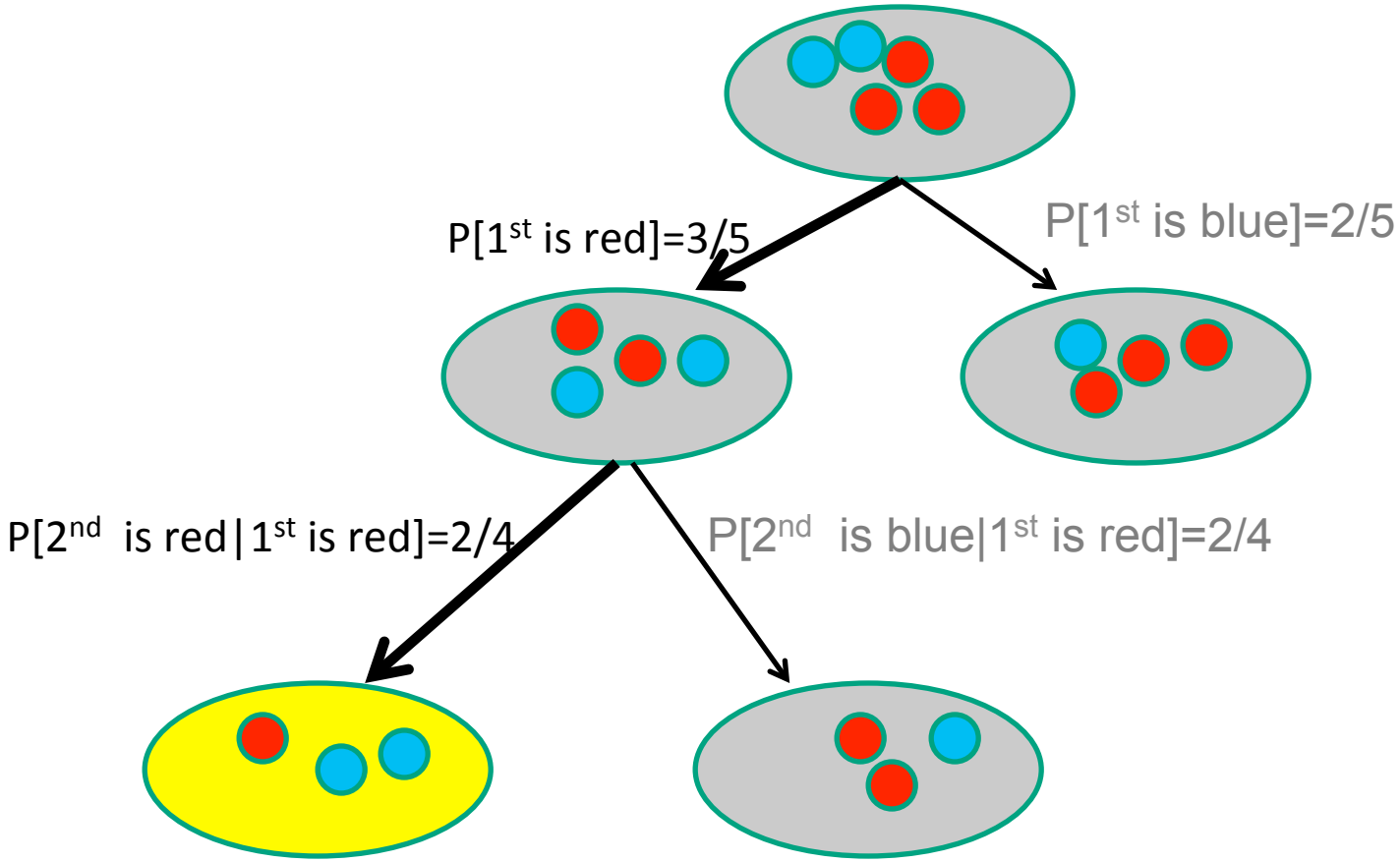
Answer: $P(R_1 \cap R_2) = P(R_1)P(R_2|R_1) = (3/5)(2/4) = 0.3$.

3 of the 5 are red

With 1 red ball gone, 2 of the remaining 4 balls are red

Alternative solution: P[2 red in 2 tries] = Comb(3,2)*Comb(2,0)/ Comb(5,2) = 0.3

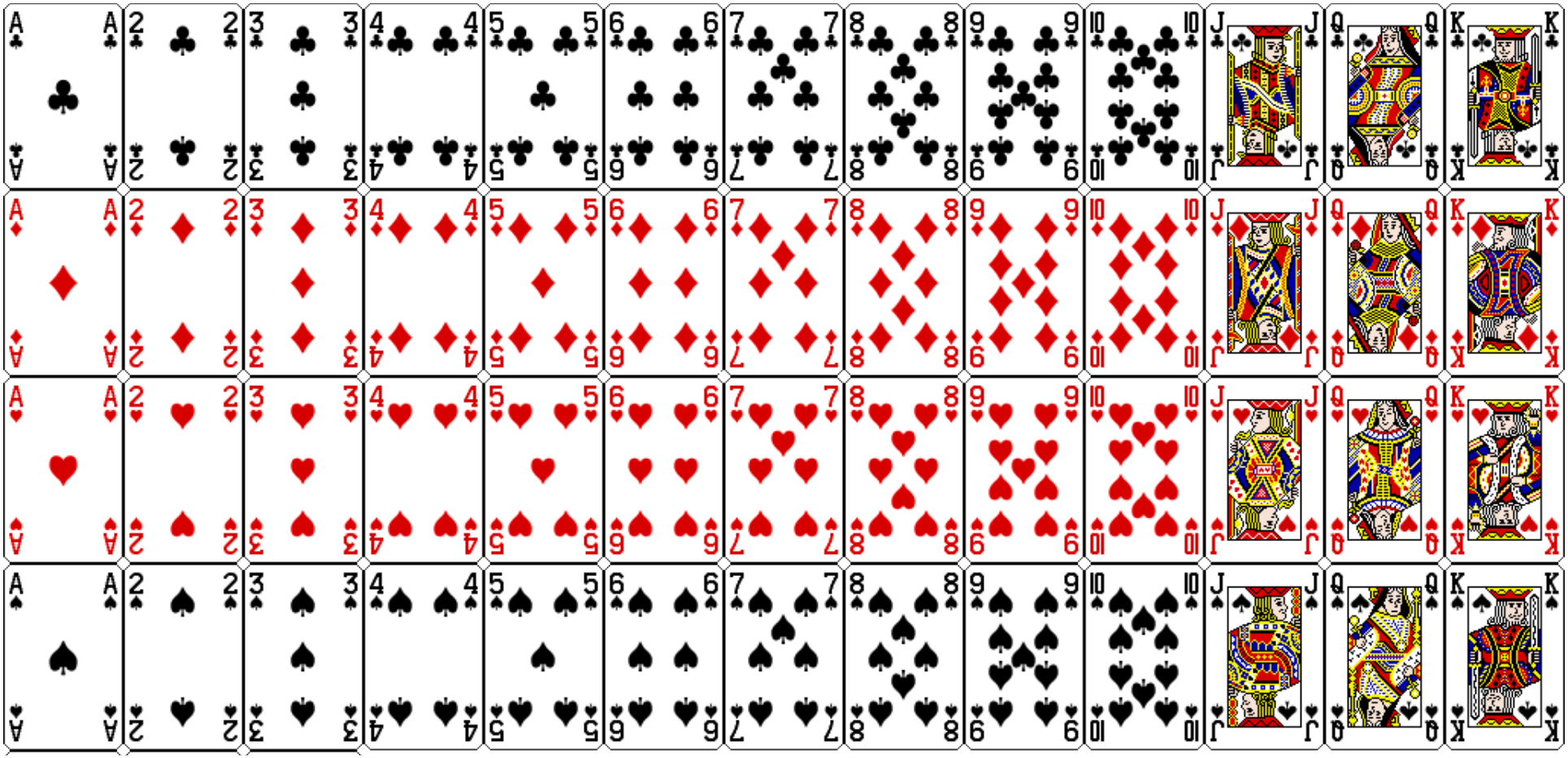
Solution via Tree Diagram



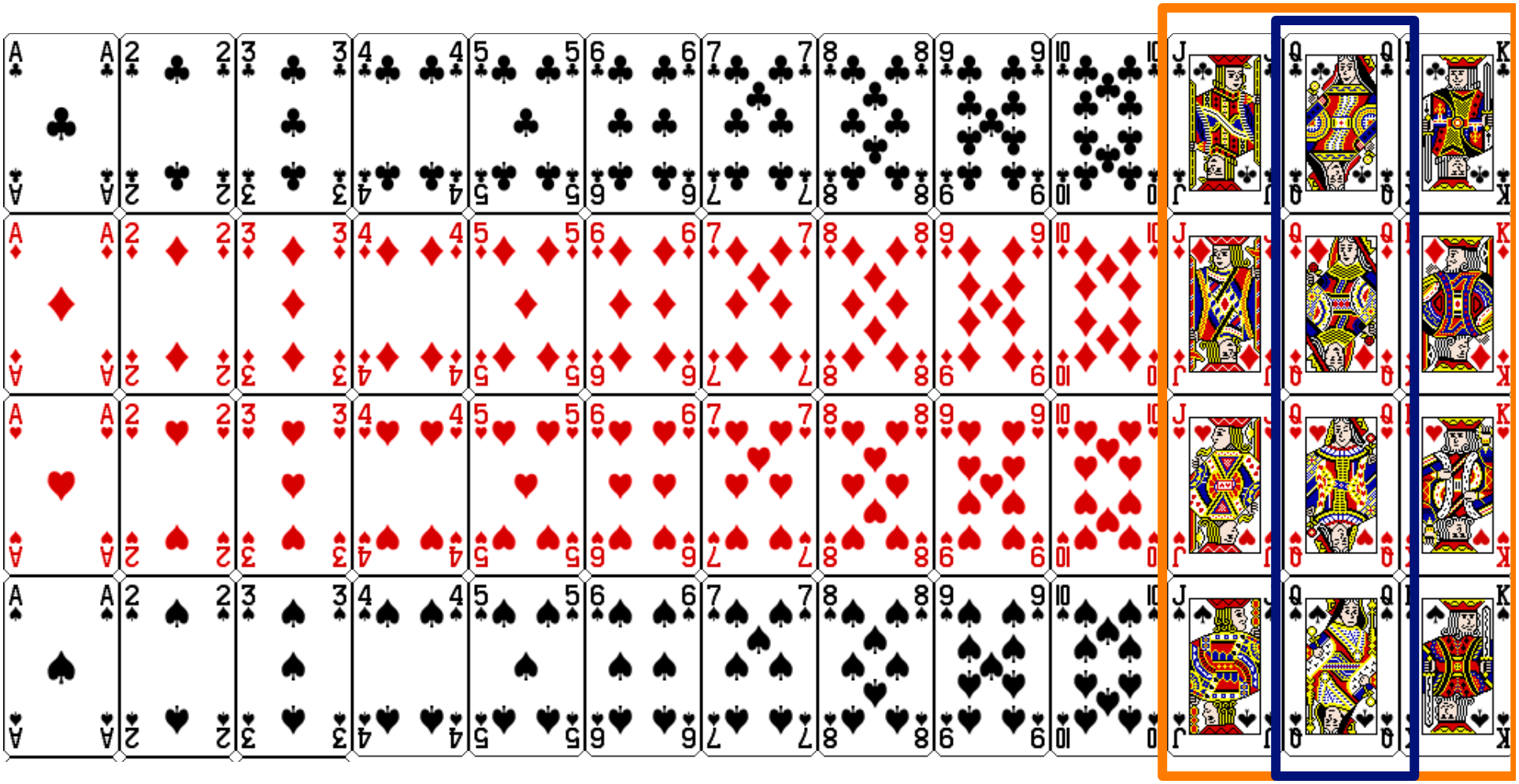
P[both are red] = p[2nd is red|1st is red] x P[1st is red] = 2/4 x 3/5 = 6/20 = 0.3

Conditional probability example

Royal Cards = {Jack, Queen, King}

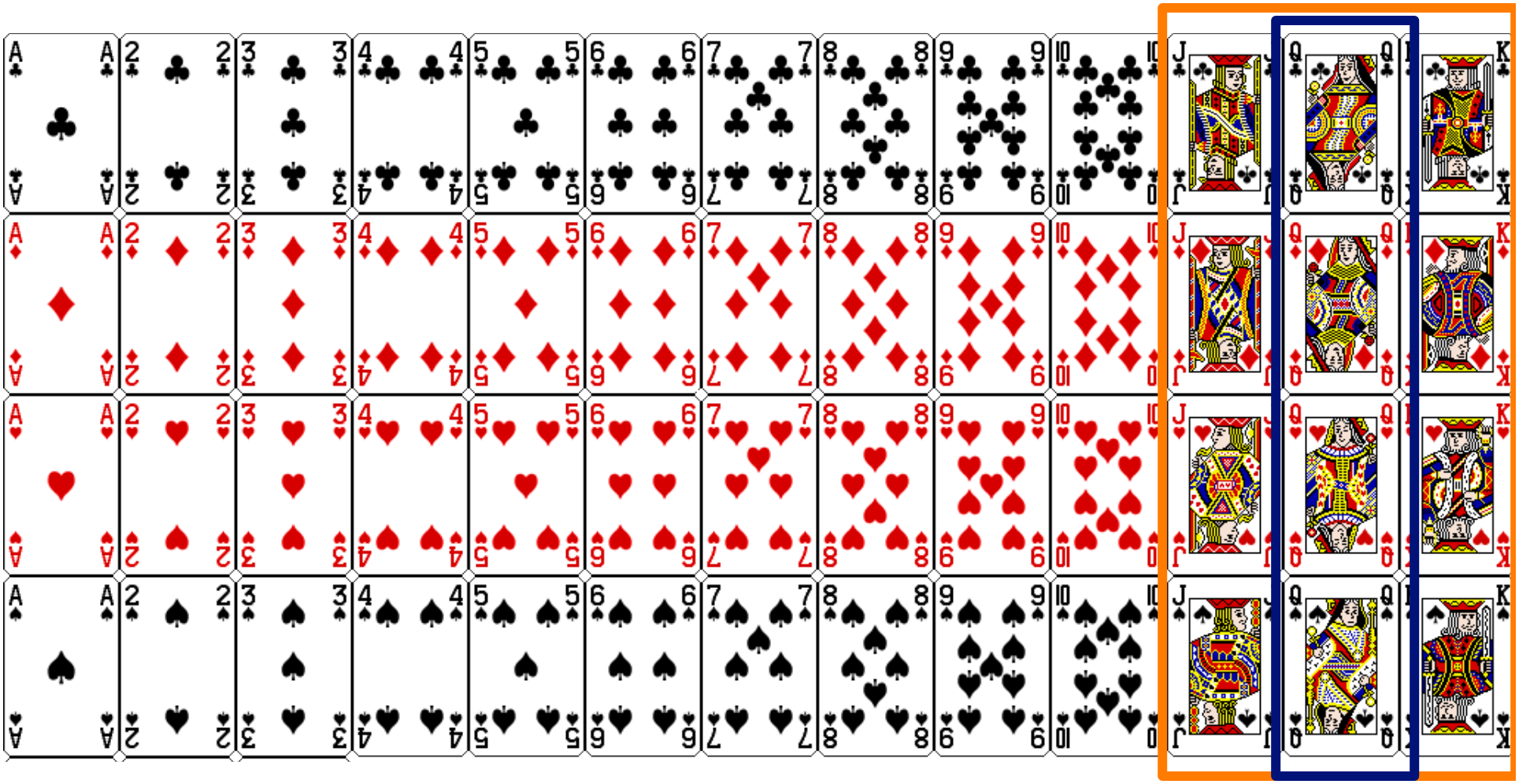


Conditional probability example



$$P(\text{Queen}|\text{RoyalCard}) = \frac{P(\text{Queen} \cap \text{RoyalCard})}{P(\text{RoyalCard})} = \frac{1/13}{3/13} = \frac{1}{3}$$

Pr(Queen|Club) is independent?



$$P(\text{Queen}|\text{RoyalCard}) = \frac{P(\text{Queen} \cap \text{RoyalCard})}{P(\text{RoyalCard})} = \frac{1/13}{3/13} = \frac{1}{3}$$

Conditional Probability Example

The initial intuition for conditional probability comes from considering probabilities that are ratios. In the case of ratios, $P(\mathsf{E}|\mathsf{F})$, as defined above, is the fraction of items in F that are also in E. We show this as follows. Let n be the number of items in the sample space, n_{F} be the number of items in F, and n_{EF} be the number of items in $\mathsf{E} \cap \mathsf{F}$. Then

$$\frac{P(\mathsf{E} \cap \mathsf{F})}{P(\mathsf{F})} = \frac{n_{\mathsf{EF}}/n}{n_{\mathsf{F}}/n} = \frac{n_{\mathsf{EF}}}{n_{\mathsf{F}}},$$

which is the fraction of items in F that are also in E. As far as meaning, $P(\mathsf{E}|\mathsf{F})$ means the probability of E occurring given that we know F has occurred.

Example 1.6 *Again consider drawing the top card from a deck of cards, let* Queen *be the set of the* 4 *queens,* RoyalCard *be the set of the* 12 *royal cards, and* Spade *be the set of the* 13 *spades. Then*

$$P(\mathsf{Queen}) = \frac{1}{13}$$

$$P(\mathsf{Queen}|\mathsf{RoyalCard}) = \frac{P(\mathsf{Queen} \cap \mathsf{RoyalCard})}{P(\mathsf{RoyalCard})} = \frac{1/13}{3/13} = \frac{1}{3}$$

$$P(\mathsf{Queen}|\mathsf{Spade}) = \frac{P(\mathsf{Queen} \cap \mathsf{Spade})}{P(\mathsf{Spade})} = \frac{1/52}{1/4} = \frac{1}{13}.$$

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case)
- **Naïve Bayes (next live session)**
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
  - Continuos input variables (Cover later)
- **Case Study: Spam detector in Naïve Bayes**

Product Rule is Fundamental

$$P(Queen \mid C\,\text{lub}) = \frac{P(Queen, C\,\text{lub})}{P(C\,\text{lub})} = \frac{\frac{1}{52}}{\frac{13}{52}} = \frac{1}{13}$$

$$P(A \mid B) = \frac{P(A, B)}{P(B)}$$

Chain Rule, Bayes Rule follow from the Product Rule

$P(A, B)$ is the belief in the joint event of A and B (joint probability)

$P(B)$ is the marginal probability of *B*

PRODUCT RULE :

$$P(A, B) = P(A \mid B)P(B)$$

Product Rule is Fundamental!

$$P(Queen \mid C\,\text{lub}) = \frac{P(Queen, C\,\text{lub})}{P(C\,\text{lub})} = \frac{\frac{1}{52}}{\frac{13}{52}} = \frac{1}{13}$$

$$P(A \mid B) = \frac{P(A,B)}{P(B)}$$

Chain Rule, Bayes Rule follow from the Product Rule

PRODUCT RULE :

$$P(A,B) = P(A \mid B)P(B)$$

Part 2!!
(Often left to your imaginations)

$$P(A \mid B) = \frac{P(A,B)}{P(B)} \; and \; P(B \mid A) = \frac{P(A,B)}{P(A)}$$

$$P(A,B) = P(A \mid B)P(B) = P(B \mid A)P(A)$$

Marginalize via Conditional Probability

Simpler calculation: get marginal from joint; decompose joint using product rule

FROM $P(A) = \sum_i P(A, B_i)$ Marginalize A from joint A,B

AND PRODUCT RULE $P(A, B) = P(A \mid B)P(B)$

$$P(A) = \sum_i P(A \mid B_i)P(B_i)$$

the belief in any event *A* is a weighted sum over the beliefs in all the distinct ways that *A* might be realized.

Chain Rule follows from the Product Rule

PRODUCT RULE: $P(A,B) = P(A \mid B)P(B)$

1. Chain Rule = Generalization of PRODUCT RULE.
2. It permits the calculation of any member of the joint distribution of a set of random variables using only conditional probabilities

$$P(E_1, E_2 \ldots E_n) = P(E_n \mid E_{n-1}, \ldots, E_2, E_1) P(E_{n-1}, \ldots, E_2, E_1)$$

let's further decompose!

$$\ldots\ldots\ldots\ldots\ldots\ldots = P(E_n \mid E_{n-1}, \ldots, E_2, E_1) \ldots P(E_2 \mid E_1) P(E_1)$$

- **For example, derive using the chain rule**

$$\mathrm{P}(A, B, C, D) = \mathrm{P}(A \mid B, C, D)\mathrm{P}(B \mid C, D)\mathrm{P}(C \mid D)\mathrm{P}(D)$$

Chain Rule Example

- **The rule is useful in the study of Bayesian networks, which describe a probability distribution in terms of conditional probabilities.**
- **Assume Urn 1 has 1 black balls and 2 white balls and Urn 2 has 1 black ball and 3 white balls. Suppose we pick an urn at random and then select a ball from that urn.**
- **Let event A be choosing the first urn: P(A) = P(~A) = 1/2. Let event B be the chance we choose a white ball. Chance of choosing a white ball, given that we've chose the first urn, is P(B|A) = 2/3. Chance of choosing a white ball, given that we've chosen the second urn is P(B|~A) = 3/4. Event A, B would be their intersection; choosing the first urn and a white ball from it. The probability can be found by the chain rule for probability:**

$$\mathrm{P}(A, B) = \mathrm{P}(B \mid A)\mathrm{P}(A) = 2/3 \times 1/2 = 1/3$$

[Russell, Stuart J.; Norvig, Peter (2003), *Artificial Intelligence: A Modern Approach* (2nd ed.), Upper Saddle River, NJ: Prentice Hall, ISBN 0-13-790395-2, http://aima.cs.berkeley.edu/ , p. 496.]

Bayes Rule

1. $P(A \mid B) = \dfrac{P(A,B)}{P(B)}$ $\quad P(B \mid A) = \dfrac{P(A,B)}{P(A)}$ Cond[al] Prob

2. $P(A,B) = P(A \mid B)P(B)$ $\quad P(A,B) = P(B \mid A)P(A)$ Product Rule

3. $P(A \mid B)P(B) = P(B \mid A)P(A)$

Prior Probability

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Posterior probability

Posterior

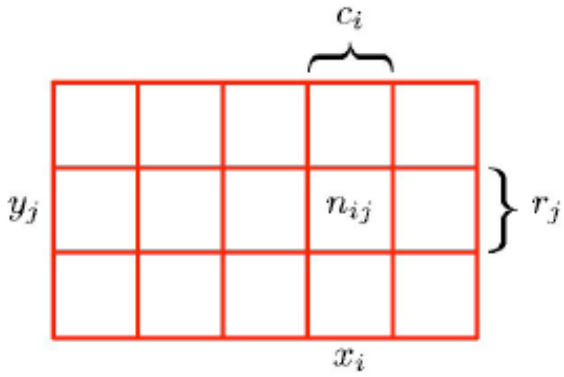
Likelihood Prior

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{\sum_{i=1}^{n} P(B \mid A = a_i)P(A = a_i)}$$

Marginalize B

Rules of Probability

- Given random variables X and Y
- Sum Rule gives Marginal Probability



$$p(X = x_i) = \sum_{j=1}^{L} p(X = x_i, Y = y_j) = \frac{c_i}{N}$$

- Product Rule: joint probability in terms of conditional and marginal

$$p(X,Y) = \frac{n_{ij}}{N} = p(Y \mid X)p(X) = \frac{n_{ij}}{c_i} \times \frac{c_i}{N}$$

- Combining we get Bayes Rule

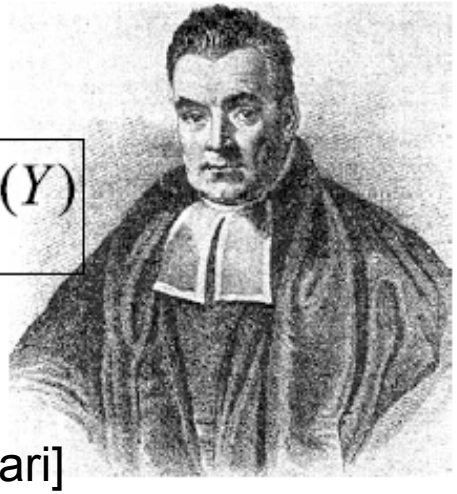
$$p(Y \mid X) = \frac{p(X \mid Y)p(Y)}{p(X)}$$

where

$$p(X) = \sum_{Y} p(X \mid Y)p(Y)$$

Viewed as

Posterior α likelihood x prior



[Srihari]

Maximum a Posteriori (MAP)

- **Any such maximally probable hypothesis is called a *maximum a posteriori (MAP) hypothesis***

Bayes Rule Example

test with two possible outcomes: $\oplus$ (positive) and $\ominus$ (negative). We have prior knowledge that over the entire population of people only .008 have this disease. Furthermore, the lab test is only an imperfect indicator of the disease. The test returns a correct positive result in only 98% of the cases in which the disease is actually present and a correct negative result in only 97% of the cases in which the disease is not present. In other cases, the test returns the opposite result. The above situation can be summarized by the following probabilities:

$$P(cancer) = .008, \quad P(\neg cancer) = .992$$
$$P(\oplus|cancer) = .98, \quad P(\ominus|cancer) = .02$$
$$P(\oplus|\neg cancer) = .03, \quad P(\ominus|\neg cancer) = .97$$

Patient has cancer

Suppose we now observe a new patient for whom the lab test returns a positive result. Should we diagnose the patient as having cancer or not? The maximum a posteriori hypothesis can be found using Equation (6.2):

$$P(\oplus|cancer)P(cancer) = (.98).008 = .0078$$
$$P(\oplus|\neg cancer)P(\neg cancer) = (.03).992 = .0298$$

Thus, $h_{MAP} = \neg cancer$. The exact posterior probabilities can also be determined by normalizing the above quantities so that they sum to 1 (e.g., $P(cancer|\oplus) = \frac{.0078}{.0078+.0298} = .21$). This step is warranted because Bayes theorem states that the posterior probabilities are just the above quantities divided by the probability of the data, $P(\oplus)$. Although $P(\oplus)$ was not provided directly as part of the problem statement, we can calculate it in this fashion because we know that $P(cancer|\oplus)$ and $P(\neg cancer|\oplus)$ must sum to 1 (i.e., either the patient has cancer or they do not). Notice that while the posterior probability of *cancer* is significantly higher than its prior probability, the most probable hypothesis is still that the patient does

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case)
- **Naïve Bayes (next live session)**
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
  - Continuos input variables (Cover later)
- **Case Study: Spam detector in Naïve Bayes**

Bayes Theorem for Machine Learning

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

- P(h) = Prior probability of hypothesis
- P(D) = Prior probability of training data D.
- P(h|D) = Probability of h given D.
- P(D|h) = Probability of D given h.

- **Bayesian Learning via**
  - Max Likelihood
  - Bayesian

Bayesian Learning: Discrete input/output variables

- **Here we consider the relationship between supervised learning, or function approximation problems, and Bayesian reasoning.**

- **We begin by considering how to design learning algorithms based on Bayes rule.**
  - Consider a supervised learning problem in which we wish to approximate an unknown target function
    - $f : X \rightarrow Y$, or equivalently $P(Y|X)$.
  - To begin, we will assume Y is a boolean-valued random variable, and X is a vector containing n boolean attributes.
  - In other words, X =<X1,X2 ...,Xn>, where Xi is the boolean random variable denoting the ith attribute of X.

Confidence interval for a binomial distribution

- In statistics, a binomial proportion confidence interval is a confidence interval for a proportion in a statistical population. It uses the proportion estimated in a statistical sample and allows for sampling error. There are several formulas for a binomial confidence interval, but all of them rely on the assumption of a binomial distribution.
- In general, a binomial distribution applies when an experiment is repeated a fixed number of times, each trial of the experiment has two possible outcomes (labeled arbitrarily success and failure), the probability of success is the same for each trial, and the trials are statistically independent.
- A simple example of a binomial distribution is the set of various possible outcomes, and their probabilities, for the number of heads observed when a (not necessarily fair) coin is flipped ten times.
- The observed binomial proportion is the fraction of the flips which turn out to be heads.
- Given this observed proportion, the confidence interval for the true proportion innate in that coin is a range of possible proportions which may contain the true proportion.
- A 95% confidence interval for the proportion, for instance, will contain the true proportion 95% of the times that the procedure for constructing the confidence interval is employed.
- Note that this does not mean that a calculated 95% confidence interval will contain the true proportion with 95% probability. Instead, one should interpret it as follows: the process of drawing a random sample and calculating an accompanying 95% confidence interval will generate a confidence interval that contains the true proportion in 95% of all cases.
- There are several ways to compute a confidence interval for a binomial proportion. The normal approximation interval is the simplest formula, and the one introduced in most basic Statistics classes and textbooks. This formula, however, is based on an approximation that does not always work well.

$$\hat{p} \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{1}{n}\hat{p}\left(1-\hat{p}\right)}$$

https://en.wikipedia.org/wiki/Binomial_proportion_confidence_interval

Confidence interval for a binomial distribution

https://en.wikipedia.org/wiki/Binomial_proportion_confidence_interval

Confidence interval for a binomial distribution

$$\hat{p} \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{1}{n}\hat{p}\left(1-\hat{p}\right)}$$

Election Poll of 1000 people (N=1000)
Assume 45% (P=0.45, Q=1-P=55%) favor candidate A (Donald) versus 49% for Candidate B (Hillary)
Standard error about the mean = SQRT(PQ/N) = SQRT(0.45*0.55/1000)=0.015=1.5%
So the 95% confidence interval surrounding Donald's support of 45% is 45%+/- 2*1.5 = [42,…48]

In machine learning:
Pr(Y=Business) = 100/1000 = 0.1
SQRT(0.1*0.9/1000) =0.01 = 1%
yielding Pr(Y=Business) Confidence interval = 0.1+/- 0.01*2
[0.08,0.12] Note we need 1,000 examples to get this tight CI;

How much data do we need to estimate Pr(Y|X) with confidence?

Learning Classifiers based on Bayes Rule

Applying Bayes rule, we see that $P(Y = y_i|X)$ can be represented as

$$P(Y = y_i|X = x_k) = \frac{P(X = x_k|Y = y_i)P(Y = y_i)}{\sum_j P(X = x_k|Y = y_j)P(Y = y_j)}$$

LHS **RHS**

where y_m denotes the mth possible value for Y, x_k denotes the kth possible vector value for X, and where the summation in the denominator is over all legal values of the random variable Y.

One way to learn $P(Y|X)$ is to use the training data to estimate $P(X|Y)$ and $P(Y)$. We can then use these estimates, together with Bayes rule above, to determine $P(Y|X = x_k)$ for any new instance x_k.

Confidence interval for a binomial distribution

$$\hat{p} \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{1}{n}\hat{p}\,(1-\hat{p})}$$

Estimating P(Y), P(X|Y) from data

- If we are going to train a Bayes classifier by estimating P(X|Y) and P(Y), then it is reasonable to ask how much training data will be required to obtain reliable estimates of these distributions.
- Let us assume training examples are generated by drawing instances at random from an unknown underlying distribution P(X), then allowing a teacher to label this example with its Y value.
- A 1,000 independently drawn training examples will usually suffice to obtain a maximum likelihood estimate of P(Y) that is within a few percent of its correct value when Y is a boolean variable.

SE(P(Y))=SQRT(0.1*0.9/1000)=0.01 for business labeled docs; assume 100 out 1000

- However, accurately estimating P(X|Y) typically requires many more examples

Impractical: so what to do?

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case)
- **Naïve Bayes (next live session)**
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
  - Continuos input variables (Cover later)
- **Case Study: Spam detector in Naïve Bayes**

Supervised Learning Via Bayes Rule

- **Consider a supervised learning problem in which we wish to approximate an unknown target function *f : X →Y, or equivalently P(Y_j | X).***
- ***For pedagogical reasons assume discrete binary variables for both and input (X)and output (Y)***
  - *To begin, we will* assume *Y is a boolean-valued random variable, and X is a vector containing n* boolean attributes. In other words, *X = hX1;X2 : : : ;Xni, where Xi is the boolean* random variable denoting the *ith attribute of X.*
- **Applying Bayes rule, we see that *P(Y = y_i | X) can be represented as***

$$P(Y = y_i | X = x_k) = \frac{P(X = x_k | Y = y_i) P(Y = y_i)}{\sum_j P(X = x_k | Y = y_j) P(Y = y_j)}$$

- **where *y_m represents the m^{th} possible value for Y, and where the summation in the* denominator is over all legal values of the random variable *Y***

Learning a (FULL) Bayesian Model

- **One way to learn *P(Y | X)***
  1. *Estimate the joint probability distribution P(X |Y) and P(Y) from the training data.*
  2. *Use these estimates, together with Bayes rule above, to determine P(Y | X = x_k) for any new instance x_k.*
- ***A maximum likelihood estimate of P(Y) can be accomplished with just a few hundred examples***
  - *#ExamplesWithLabel/#TotalNumberOfExamples. E.g., 45 examples are in class 1 and 55 are in class2 then Pr(Y=Class1) = 45/100=0.45.*
- ***However, estimating the joint probability distribution P(X |Y) requires an exponential amount training examples (even with this assumption of binary input and output variable)***
  - *Assume n input attributes X_i take 2 discrete values and Y has 2 possible class values; 2^n * 2 possible states of the world (parameters)*

Learning a Bayesian Model

- ***2^n * 2 possible states of the world***
  - *Assume n input attributes X_i take 2 discrete values and Y has 2 possible class values; 2^n * 2 possible states of the world (parameters) that* we need to estimate from data
  - $\theta_{iJ} = P(X = x_i | Y = y_J)$; *$2^n$ possible states of the input world 2 possible output states*
- ***Can reduce 2^n -1 * 2 parameters by exploiting the sum-to-1***
  - *Class conditional multinomial needs to sum to 1 so we exploit this and can infer one of the class conditional probs from the rest*
  - *Each state is a complex combination of feature values*
- *So for 10 input variables and one output variable we have to estimate 2046 states to estimate probabilities …..requiring potentially 204,600 examples (or more) for reliable estimates…..*
- *So not very realistic even in these WWW times*
  - *Suffer from the curse of dimensionality*

Estimating the Joint Prob. Directly?

- **2^n * 2 possible states of the world (parameter estimates)**
  - Assume n input attributes X_i take 2 discrete values and Y has 2 possible class values; 2^n * 2 possible states of the world (parameters) that we need to estimate from data
  - $\theta_{iJ} = P(X = x_i | Y = y_J)$; 2^n possible states of the input world 2 possible output states
- **Can reduce 2^n -1 * 2 parameters by exploiting the sum-to-1**
  - Class conditional multinomial needs to sum to 1 so we exploit this and can infer one of the class conditional probs from the rest
  - Each state is a complex combination of feature values
- **Require 200k examples (or more) for reliable estimates of 10 binary variable problem**
  - So for 10 input variables and one output variable we have to estimate 2046 states. To estimate probabilities requiring at least 204,600 examples for reliable estimates.
- **So not very realistic, even in these WWW times**

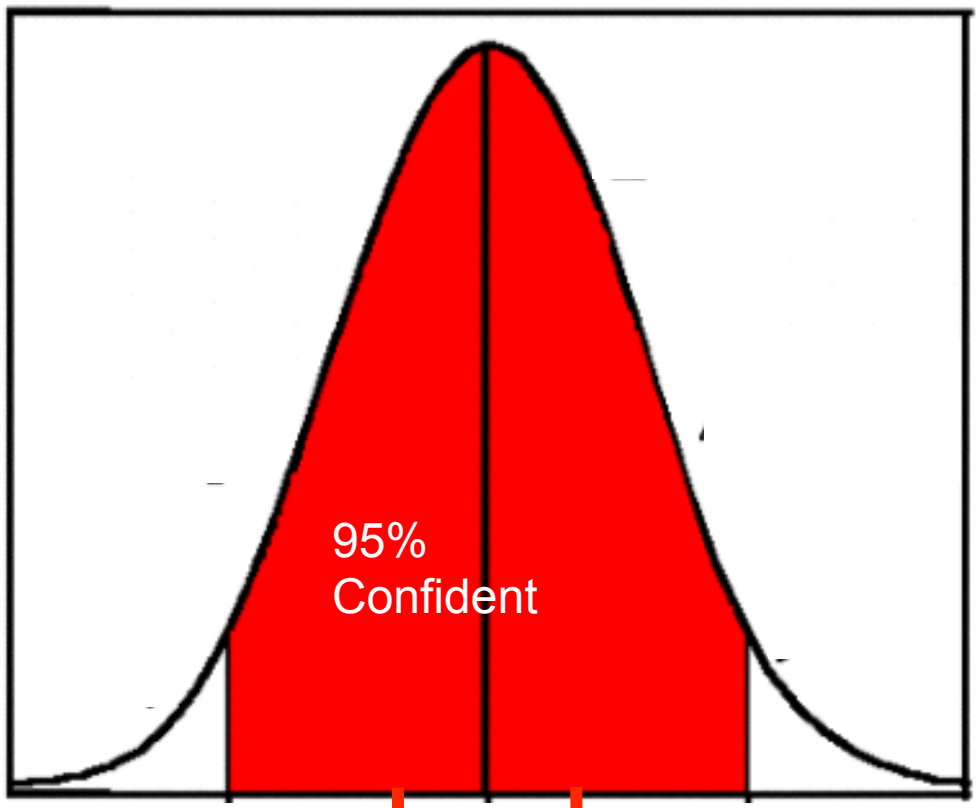
Learning a Bayesian Model

- *2^n * 2 possible states of the world*
  - *Assume n input attributes X_i take 2 discrete values and Y has 2 possible class values; 2^n * 2 possible states of the world (parameters) that* we need to estimate from data
    - $\theta_{iJ} = P(X = x_i \mid Y = y_J)$; *$2^n$ possible states of the input world 2 possible output states*

Index $X=X^i$	X1 $X=x_1^i$)	X2 $X=x_2^i$)	Y $Y=y_j$	$\theta_{ij}=P(X=x^i \mid y=y_j)$
1	1	1	1	θ_{11}
2	0	1	1	θ_{21}
3	1	0	1	θ_{31}
4	0	0	1	θ_{41}
5	1	1	0	θ_{50}
6	0	1	0	θ_{60}
7	1	0	0	θ_{70}
8	0	0	0	θ_{80}

Sum to 1

Estimating Reliable Probabilites



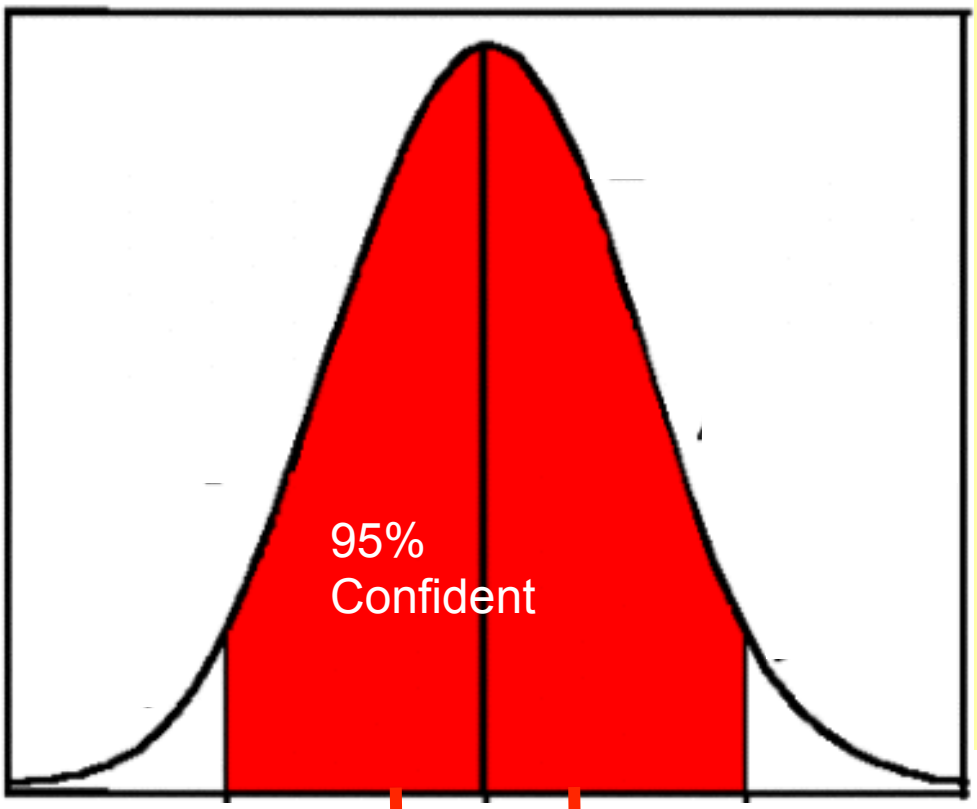
Estimate using Binomial MLE Estimates
I.e., #Clicks/#Impression

$40/1,000 @CPC of $1.60
$400/10,000

Standard error of the mean of one sample is the estimate of the standard deviation that would be obtained from the means of a large number of samples drawn from that population.

StdErr= sqrt(.026*(1-0.026)/1000) * 1.96=±1%

Estimating CTR (and later AR)



For a network of
~10^9 target pages,
~10^6 ads
~10^7 users
……
- Cannot afford this evaluation/auditioning
- Borrow strength, marginalize
- CoD (curse of dim^{ality})

Confidence Intervals in a Nutshell

- **The assumptions required for CI for a population proportion to be valid:**
  - the sample size *n* is large enough (check: $n\hat{p} \geq 10$ and $n(1-\hat{p}) \geq 10$)
  - the data are a *random sample* from that population
- **General Confidence Interval for the Population Proportion *p*:**

$$\hat{p} \pm 2\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

- **Approximate 95% Confidence Interval for the Population Proportion *p*:**

$$\hat{p} \pm z^* \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

```
> 2*sqrt(.4*.6/1000)  #president's satisfaction rating
[1] 0.03098387
```

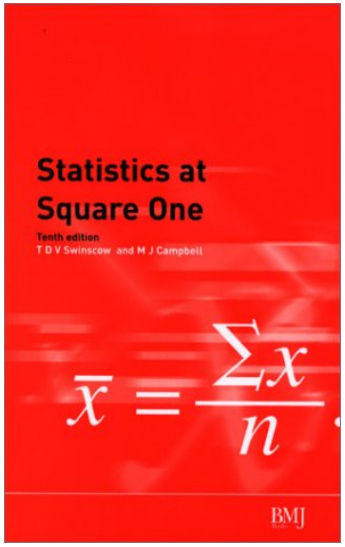
  - Note: standard error of is which is largest when $\hat{p} = \frac{1}{2}$
- **Conservative Confidence Interval for the Population Proportion *p*:**

$$\hat{p} \pm \frac{z^*}{2\sqrt{n}}$$

1/sqrt(n), e.g., 1/sqrt(1000)= ± 3%

Sample Size Needed

- **Sample Size Needed for Desired Confidence Level and Error Margin where *m* is the desired margin of error.**



$$n = \left(\frac{z^*}{2m} \right)^2$$

Measure CTRs confidently p± 0.001
> (1.96/(2*.001))^2
[1] 960,400 sample size

[http://www.bmj.com/statsbk/ Statistics at Square One (Ninth Edition), T D V Swinscow]

Sample Size and Stats

- More later in the context of AB testing…..

Cant Estimate Joint Probability

- **Look for ways to combat the intractable data needs for learning a Bayesian Classifier**
  - Leverage the chain rule and other assumptions (Markov, Naïve Bayes); this leads to Bayesian Networks
  - Make a conditional independence assumption; this leads to a Naïve Bayes classifier
    - Reduces the number of parameters from 2^n -1 * 2 parameters to 2n

Bayesian Networks

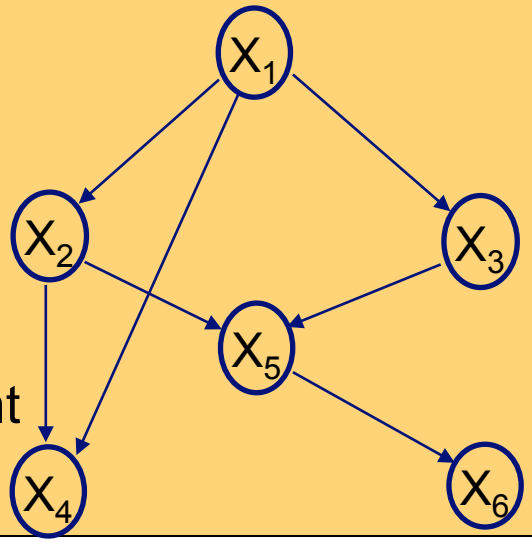
2^N Possibilities -> 2N

P: Joint Probability Distribution

#	X1	X2	X3	X4	X5	X6	Pr(X1,X2, X3, X4, X5, X6)
1	1						
2	1						
..	1						
4..	1						
5	1						
6	1						
7	1						
64							

Factorize the joint

G: Directed Acyclic Graph



2. By Chain Rule

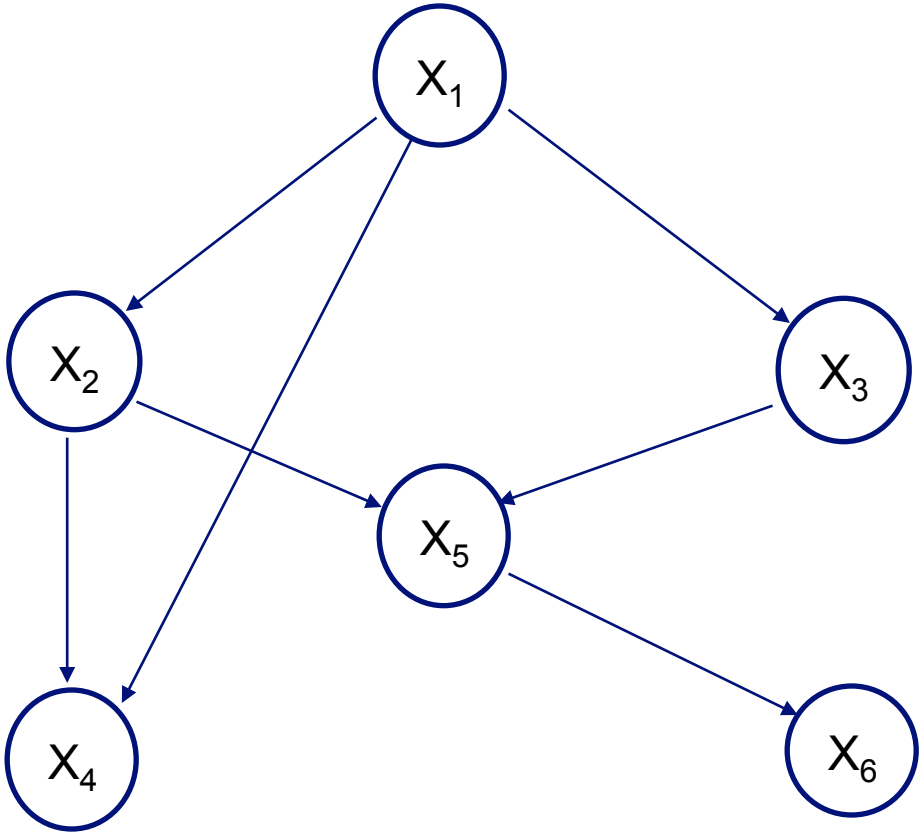
$p(x_1, x_2, x_3, x_4, x_5, x_6)$ 1. Partial Order

$= p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1, x_2)\, p(x_4 \mid x_1, x_2, x_3)\, p(x_5 \mid x_4, x_3, x_2, x_1)\, p(x_6 \mid x_5, x_4, x_3, x_2, x_1)$

$= p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1)\, p(x_4 \mid x_2, x_1)\, p(x_5 \mid x_3, x_2)\, p(x_6 \mid x_5)$ 3. Markov Property

4: Naïve Bayes

1, 2, 3 Example of Factorization



2. By Chain Rule

$p(x_1, x_2, x_3, x_4, x_5, x_6)$ 1. Partial Order

$= p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1, x_2)\, p(x_4 \mid x_1, x_2, x_3)\, p(x_5 \mid x_4, x_3, x_2, x_1)\, p(x_6 \mid x_5, x_4, x_3, x_2, x_1)$

$= p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1)\, p(x_4 \mid x_2, x_1)\, p(x_5 \mid x_3, x_2)\, p(x_6 \mid x_5)$ 3. Markov Property

Factorization

- Given a DAG G, topologically sort the variables:

  $X_1, \ldots, X_n$ (s.t. if i<j, then X_j is not an ancestor of X_i)

- For any joint distribution P that is Markov to G, factorize it as follows:

  $P(X_1, \ldots, X_n)$

  $= P(X_1)\, P(X_2|X_1) \ldots P(X_n|X_1, \ldots, X_{n-1})$ By Chain Rule

  $= \prod_i P(X_i \mid Pa(X_i))$ Exploit Local Markov Property

- Markov property: every variable is independent of its non-descendants given its parents.
- Markov Condition requires that every conditional independence in the graph is in the joint probability distribution.

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case)
- **Naïve Bayes (next live session)**
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
  - Continuos input variables (Cover later)
- **Case Study: Spam detector in Naïve Bayes**

- **Independence**

Independence

- **In probability theory, two events are independent, statistically independent, or stochastically independent if the occurrence of one does not affect the probability of the other.**
  - Similarly, two random variables are independent if the realization of one does not affect the probability distribution of the other.
  - The concept of independence extends to dealing with collections of more than two events or random variables, in which case the events are pairwise independent if each pair are independent of each other, and the events are mutually independent if each event is independent of each other combination of events.

https://en.wikipedia.org/wiki/Independence_(probability_theory)

Two events [edit]

Two events *A* and *B* are **independent** (often written as $A \perp B$ or $A \perp\!\!\!\perp B$) if and only if their joint probability equals the product of their probabilities:

$$\mathrm{P}(A \cap B) = \mathrm{P}(A)\mathrm{P}(B).$$

Why this defines independence is made clear by rewriting with conditional probabilities:

$$\mathrm{P}(A \cap B) = \mathrm{P}(A)\mathrm{P}(B) \Leftrightarrow \mathrm{P}(A) = \frac{\mathrm{P}(A)\mathrm{P}(B)}{\mathrm{P}(B)} = \frac{\mathrm{P}(A \cap B)}{\mathrm{P}(B)} = \mathrm{P}(A \mid B)$$

and similarly

$$\mathrm{P}(A \cap B) = \mathrm{P}(A)\mathrm{P}(B) \Leftrightarrow \mathrm{P}(B) = \mathrm{P}(B \mid A).$$

Thus, the occurrence of *B* does not affect the probability of *A*, and vice versa. Although the derived expressions may seem more intuitive, they are not the preferred definition, as the conditional probabilities may be undefined if *P*(*A*) or *P*(*B*) are 0. Furthermore, the preferred definition makes clear by symmetry that when *A* is independent of *B*, *B* is also independent of *A*.

More than two events [edit]

A finite set of events $\{A_i\}$ is **pairwise independent** if and only if every pair of events is independent[2]—that is, if and only if for all distinct pairs of indices *m*, *k*,

$$\mathrm{P}(A_m \cap A_k) = \mathrm{P}(A_m)\mathrm{P}(A_k).$$

A finite set of events is **mutually independent** if and only if every event is independent of any intersection of the other events[2]—that is, if and only if for every *n*-element subset $\{A_i\}$,

$$\mathrm{P}\left(\bigcap_{i=1}^{n} A_i\right) = \prod_{i=1}^{n} \mathrm{P}(A_i).$$

This is called the *multiplication rule* for independent events. Note that it is not a single condition involving only the product of all the probabilities of all single events (see below for a counterexample); it must hold true for all subset of events.

For more than two events, a mutually independent set of events is (by definition) pairwise independent; but the converse is not necessarily true (see below for a counterexample).

Independence

Definition 1.3 *Two events* E *and* F *are independent if one of the following hold:*

1. $P(\mathsf{E}|\mathsf{F}) = P(\mathsf{E})$ *and* $P(\mathsf{E}) \neq 0,\ P(\mathsf{F}) \neq 0.$
2. $P(\mathsf{E}) = 0$ *or* $P(\mathsf{F}) = 0.$

Notice that the definition states that the two events are independent even though it is based on the conditional probability of E given F. The reason is that independence is symmetric. That is, if $P(\mathsf{E}) \neq 0$ and $P(\mathsf{F}) \neq 0$, then $P(\mathsf{E}|\mathsf{F}) = P(\mathsf{E})$ if and only if $P(\mathsf{F}|\mathsf{E}) = P(\mathsf{F})$. It is straightforward to prove that E and F are independent if and only if $P(\mathsf{E} \cap \mathsf{F}) = P(\mathsf{E})P(\mathsf{F})$.

Independence

Definition 1.6 *Suppose we have a probability space (Ω, P), and two sets* A *and* B *containing random variables defined on* Ω. *Then the sets* A *and* B *are said to be* ***independent*** *if, for all values of the variables in the sets* a *and* b, *the events* $\mathsf{A} = \mathsf{a}$ *and* $\mathsf{B} = \mathsf{b}$ *are independent. That is, either* $P(\mathsf{a}) = 0$ *or* $P(\mathsf{b}) = 0$ *or*

$$P(\mathsf{a}|\mathsf{b}) = P(\mathsf{a}).$$

When this is the case, we write

$$I_P(\mathsf{A}, \mathsf{B}),$$

where I_P *stands for independent in* P.

Independence

Example 1.18 *Let Ω be the set of all cards in an ordinary deck, and let P assign $1/52$ to each card. Define random variables as follows:*

Variable	Value	Outcomes Mapped to this Value
R	$r1$	*All royal cards*
	$r2$	*All nonroyal cards*
T	$t1$	*All tens and jacks*
	$t2$	*All cards that are neither tens nor jacks*
S	$s1$	*All spades*
	$s2$	*All nonspades*

Then we maintain the sets $\{R, T\}$ and $\{S\}$ are independent. That is,

$$I_P(\{R, T\}, \{S\}).$$

To show this, we need show for all values of r, t, and s that

$$P(r, t|s) = P(r, t).$$

(Note that it we do not show brackets to denote sets in our probabilistic expression because in such an expression a set represents the members of the set. See the discussion following Example 1.14.) The following table shows this is the case:

s	r	t	$P(r,t\|s)$	$P(r,t)$
$s1$	$r1$	$t1$	$1/13$	$4/52 = 1/13$
$s1$	$r1$	$t2$	$2/13$	$8/52 = 2/13$
$s1$	$r2$	$t1$	$1/13$	$4/52 = 1/13$
$s1$	$r2$	$t2$	$9/13$	$36/52 = 9/13$
$s2$	$r1$	$t1$	$3/39 = 1/13$	$4/52 = 1/13$
$s2$	$r1$	$t2$	$6/39 = 2/13$	$8/52 = 2/13$
$s2$	$r2$	$t1$	$3/39 = 1/13$	$4/52 = 1/13$
$s2$	$r2$	$t2$	$27/39 = 9/13$	$36/52 = 9/13$

Definition 1.7 *Suppose we have a probability space* (Ω, P), *and three sets* A, B, *and* C *containing random variable defined on* Ω. *Then the sets* A *and* B *are said to be conditionally independent given the set* C *if, for all values of the variables in the sets* a, b, and c, *whenever* $P(\mathsf{c}) \neq 0$, *the events* $\mathsf{A} = \mathsf{a}$ *and* $\mathsf{B} = \mathsf{b}$ *are conditionally independent given the event* $\mathsf{C} = \mathsf{c}$. *That is, either* $P(\mathsf{a}|\mathsf{c}) = 0$ *or* $P(\mathsf{b}|\mathsf{c}) = 0$ *or*

$$P(\mathsf{a}|\mathsf{b},\mathsf{c}) = P(\mathsf{a}|\mathsf{c}).$$

When this is the case, we write

$$I_P(\mathsf{A},\mathsf{B}|\mathsf{C}).$$

Independence Example

Example 1.6 *Again consider drawing the top card from a deck of cards, let* Queen *be the set of the* 4 *queens,* RoyalCard *be the set of the* 12 *royal cards, and* Spade *be the set of the* 13 *spades. Then*

$$P(\text{Queen}) = \frac{1}{13}$$

$$P(\text{Queen}|\text{RoyalCard}) = \frac{P(\text{Queen} \cap \text{RoyalCard})}{P(\text{RoyalCard})} = \frac{1/13}{3/13} = \frac{1}{3}$$

$$P(\text{Queen}|\text{Spade}) = \frac{P(\text{Queen} \cap \text{Spade})}{P(\text{Spade})} = \frac{1/52}{1/4} = \frac{1}{13}.$$

Notice in the previous example that $P(\text{Queen}|\text{Spade}) = P(\text{Queen})$. This means that finding out the card is a spade does not make it more or less probable that it is a queen. That is, the knowledge of whether it is a spade is irrelevant to whether it is a queen. We say that the two events are independent in this case, which is formalized in the following definition.

[Learning Bayesian Networks, Prentice-Hall, Richard E. Neapolitan]

Independence

- **Pairwise Independence**
  - P(A,B|C)=P(A|C)P(B|C)
    - Since P(A,B|C) = P(A|B,C)P(B|C) [Chain rule]
    - and P(A|B, C)= P(A|C)
- **Conditional Independence**
  - P(A|B)=P(A)
  - P(A|B, C)= P(A|C)
  - P(*A, B*)=P(*A*)P(*B*)

Probability Basics → Naïve Bayes Models

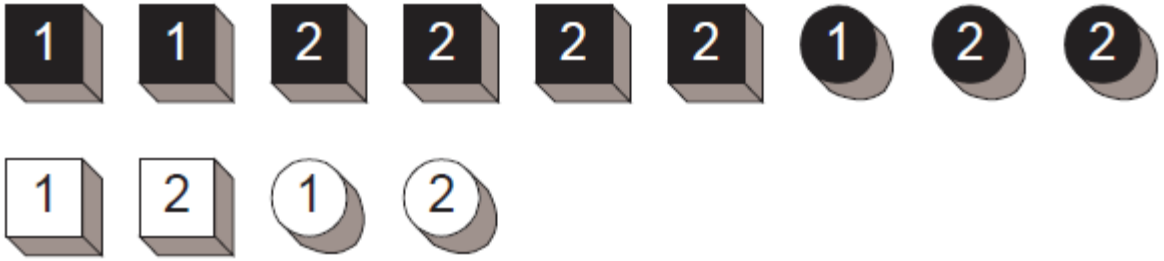
- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case)
- **Naïve Bayes (next live session)**
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
  - Continuos input variables (Cover later)
- **Case Study: Spam detector in Naïve Bayes**

Conditional Independence

- *Definition:* Given random variables X, Y and Z, we say X is **conditionally independent** of Y given Z, if and only if the probability distribution governing X is independent of the value of Y given Z; that is

$$(\forall i, j, k) P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

As an example, consider three boolean random variables to describe the current weather: *Rain*, *Thunder* and *Lightning*. We might reasonably assert that *Thunder* is independent of *Rain* given *Lightning*. Because we know *Lightning* causes *Thunder*, once we know whether or not there is *Lightning*, no additional information about *Thunder* is provided by the value of *Rain*. Of course there is a clear dependence of *Thunder* on *Rain* in general, but there is no *conditional* dependence once we know the value of *Lightning*.



Conditionally Independent

Figure 1.2: Containing a '1' and being a square are not independent, but they are conditionally independent given the object is black and given it is white.

Example 1.19 *Let Ω be the set of all objects in Figure 1.2, and let P assign $1/13$ to each object. Define random variables S (for shape), V (for value), and C (for color) as follows:*

Variable	Value	Outcomes Mapped to this Value
V	$v1$	*All objects containing a '1'*
	$v2$	*All objects containing a '2'*
S	$s1$	*All square objects*
	$s2$	*All round objects*
C	$c1$	*All black objects*
	$c2$	*All white objects*

Then we maintain that $\{V\}$ and $\{S\}$ are conditionally independent given $\{C\}$. That is,

$$I_P(\{V\}, \{S\}|\{C\}).$$

To show this, we need show for all values of v, s, and c that

$$P(v|s, c) = P(v|c).$$

The results in Example 1.8 show $P(v1|s1, c1) = P(v1|c1)$ and $P(v1|s1, c2) = P(v1|c2)$. The table that follows shows the equality holds for the other values of the variables too:

c	s	v	$P(v\|s,c)$	$P(v\|c)$
$c1$	$s1$	$v1$	$2/6 = 1/3$	$3/9 = 1/3$
$c1$	$s1$	$v2$	$4/6 = 2/3$	$6/9 = 2/3$
$c1$	$s2$	$v1$	$1/3$	$3/9 = 1/3$
$c1$	$s2$	$v2$	$2/3$	$6/9 = 2/3$
$c2$	$s1$	$v1$	$1/2$	$2/4 = 1/2$
$c2$	$s1$	$v2$	$1/2$	$2/4 = 1/2$
$c2$	$s2$	$v1$	$1/2$	$2/4 = 1/2$
$c2$	$s2$	$v2$	$1/2$	$2/4 = 1/2$

[Learning Bayesian Networks, Prentice-Hall, Richard E. Neapolitan]

Conditiona Independence (Wiki)

Conditional independence [edit]

Main article: Conditional independence

Intuitively, two random variables X and Y are conditionally independent given Z if, once Z is known, the value of Y does not add any additional information about X. For instance, two measurements X and Y of the same underlying quantity Z are not independent, but they are **conditionally independent given Z** (unless the errors in the two measurements are somehow connected).

The formal definition of conditional independence is based on the idea of conditional distributions. If X, Y, and Z are discrete random variables, then we define X and Y to be *conditionally independent given Z* if

$$\mathrm{P}(X \leq x, Y \leq y \mid Z = z) = \mathrm{P}(X \leq x \mid Z = z) \cdot \mathrm{P}(Y \leq y \mid Z = z)$$

for all x, y and z such that $\mathrm{P}(Z = z) > 0$. On the other hand, if the random variables are continuous and have a joint probability density function p, then X and Y are conditionally independent given Z if

$$p_{XY|Z}(x, y|z) = p_{X|Z}(x|z) \cdot p_{Y|Z}(y|z)$$

for all real numbers x, y and z such that $p_Z(z) > 0$.

If X and Y are conditionally independent given Z, then

$$\mathrm{P}(X = x|Y = y, Z = z) = \mathrm{P}(X = x|Z = z)$$

for any x, y and z with $\mathrm{P}(Z = z) > 0$. That is, the conditional distribution for X given Y and Z is the same as that given Z alone. A similar equation holds for the conditional probability density functions in the continuous case.

Independence can be seen as a special kind of conditional independence, since probability can be seen as a kind of conditional probability given no events.

https://en.wikipedia.org/wiki/Conditional_independence

Conditional independence

- **V is** conditionally independent **of set *Vi* given set *Vj***
  - if p(V | ***Vi, Vj***) = p(V | ***Vj***)
  - notation: I(V, ***Vi*** | ***Vj***) or V ⊥ ***Vi*** | ***Vj***
- **Intuition**
  - if I(V, ***Vi*** | ***Vj***) then knowing ***Vi & Vj*** tells nothing more about V than knowing ***Vj*** alone
  - if we know ***Vj*** we can ignore ***Vi***

$$P(Queen \mid C\,\text{lub}) = \frac{P(Queen, C\,\text{lub})}{P(C\,\text{lub})} = \frac{\frac{1}{52}}{\frac{13}{52}} = \frac{1}{13}$$

$$P(Queen) = \frac{4}{52} = \frac{1}{13}$$

Pairwise Independence

- **Single Vi conditionally independent of single Vj given *V***
  - that is, I(Vi, Vj | ***V***)
- **From definitions we have that**
  - p(Vi | Vj,**V**) = p(Vi | ***V***) and
  - p(Vi | Vj,**V**) p(Vj | ***V***) = p(Vi,Vj | ***V***)
    (Product Rule: $P(A,B) = P(A \mid B)P(B)$)
- **Thus**
  - p(Vi,Vj | ***V***) = p(Vi | ***V***) p(Vj | ***V***)
  - Vi and Vj is **pairwise independent**

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case)
- **Naïve Bayes (next live session)**
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
  - Continuos input variables (Cover later)
- **Case Study: Spam detector in Naïve Bayes**

Derive NB Algorithm

$$P(Y = y_k | X_1, X_2, ..., X_N) = \frac{P(Y=y_k) P(X_1, X_2, .., X_N | Y=y_k)}{\sum_j P(Y=y_j) P(X_1, X_2, .., X_N | Y=y_j)}$$

$$= \frac{P(Y=y_k) \Pi_i P(X_i | Y=y_k)}{\sum_j P(Y=y_j) \Pi_i P(X_i | Y=y_j)}$$

- ..

The Naive Bayes algorithm is a classification algorithm based on Bayes rule, that assumes the attributes $X_1 \ldots X_n$ are all conditionally independent of one another, given Y. The value of this assumption is that it dramatically simplifies the representation of $P(X|Y)$, and the problem of estimating it from the training data. Consider, for example, the case where $X = \langle X_1, X_2 \rangle$. In this case

$$\begin{aligned} P(X|Y) &= P(X_1, X_2|Y) \\ &= P(X_1|X_2, Y) P(X_2|Y) \\ &= P(X_1|Y) P(X_2|Y) \end{aligned}$$

Naïve Bayes

Where the second line follows from a general property of probabilities (product Rule), and the third line follows directly from our above definition of conditional independence.

Naïve Bayes Classifier for Text

100 business docs = Pr(Business) =0.1+/- CI 2^N

$$P(Y = y_k | X_1, X_2, ..., X_N) = \frac{P(Y=y_k)P(X_1, X_2, .., X_N | Y=y_k)}{\sum_j P(Y=y_j)P(X_1, X_2, .., X_N | Y=y_j)}$$

Y_1

Y_2

$$= \frac{P(Y=y_k)\Pi_i P(X_i | Y=y_k)}{\sum_j P(Y=y_j)\Pi_i P(X_i | Y=y_j)}$$

Pr(X="corporation"|Class=Business) = 1/100
10,000Words in the 10 business documents
"corporation" occurs 100 times

$$Y \leftarrow argmax_{y_k} P(Y = y_k)\Pi_i P(X_i | Y = y_k)$$

argmax_yk means find the value of yk that maximises the expression

- More generally, when X contains n attributes which are conditionally independent of one another given Y, we have

$$P(X_1 \ldots X_n|Y) = \prod_{i=1}^{n} P(X_i|Y) \quad (1)$$

Notice that when Y and the X_i are boolean variables, we need only $2n$ parameters to define $P(X_i = x_{ik}|Y = y_j)$ for the necessary i, j, k. This is a dramatic reduction compared to the $2(2^n - 1)$ parameters needed to characterize $P(X|Y)$ if we make no conditional independence assumption.

Naïve Bayes

- A generative, parametric model
- Computes the conditional a-posterior probabilities of a categorical class variable given independent predictor variables using the Bayes rule.
- The standard naive Bayes classifier (at least the R implementation) assumes independence of the predictor variables, and Gaussian distribution (given the target class) of metric predictors.
- For attributes with missing values, the corresponding table entries are omitted for prediction.

Naïve Bayes and Conditional Independence

- **Make a conditional independence assumption; this leads to a Naïve Bayes classifier**
  - Reduces the number of parameters from 2n -1 * 2 parameters to 2n
- ***Definition: Given random variables X;Y and Z, we say X is conditionally* independent of *Y given Z, if and only if the probability* distribution governing *X is independent of the value of Y given Z;***
  - $(\forall i; j; k)\ P(X = x_i \mid Y = y_j, Z = z_k) = P(X = x_i \mid Z = z_k)$

Naïve Bayes Classifier for Text

$$P(Y = y_k | X_1, X_2, ..., X_N) = \frac{P(Y=y_k) P(X_1, X_2, .., X_N | Y=y_k)}{\sum_j P(Y=y_j) P(X_1, X_2, .., X_N | Y=y_j)}$$

Y_1
Y_2

$$= \frac{P(Y=y_k) \Pi_i P(X_i | Y=y_k)}{\sum_j P(Y=y_j) \Pi_i P(X_i | Y=y_j)}$$

Pr("corporation"|Class=Business) = 1/100
10,000Words in the 10 business documents
"corporation" occurs 100 times

$$Y \leftarrow argmax_{y_k} P(Y = y_k) \Pi_i P(X_i | Y = y_k)$$

argmax_yk means find the value of yk that maximises the expression

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case)
- **Naïve Bayes (next live session)**
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
  - Continuous input variables (Cover later)
- **Case Study: Spam detector in Naïve Bayes**

- **End of live session**