10-701: Introduction to

~~Deep Neural Networks~~

Machine Learning

http://www.cs.cmu.edu/~10701

Organizational info

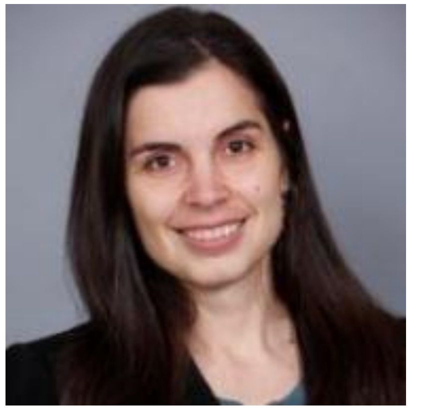
- All up-to-date info is on the course web page (follow links from my page).
- Instructors

  - Nina balcan

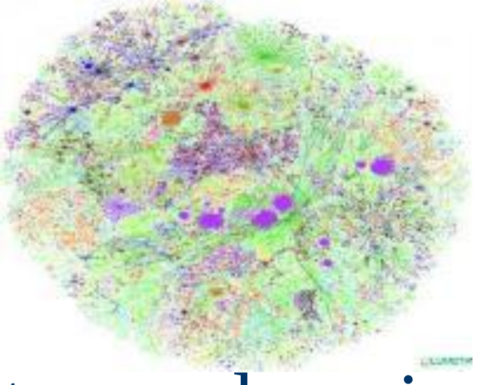
  - Ziv Bar-Joseph

- TAs: See info on website for recitations, office hours etc.
- See web page for contact info, office hours, etc.
- Piazza would be used for questions / comments and likely for class quizzes. Make sure you are subscribed.

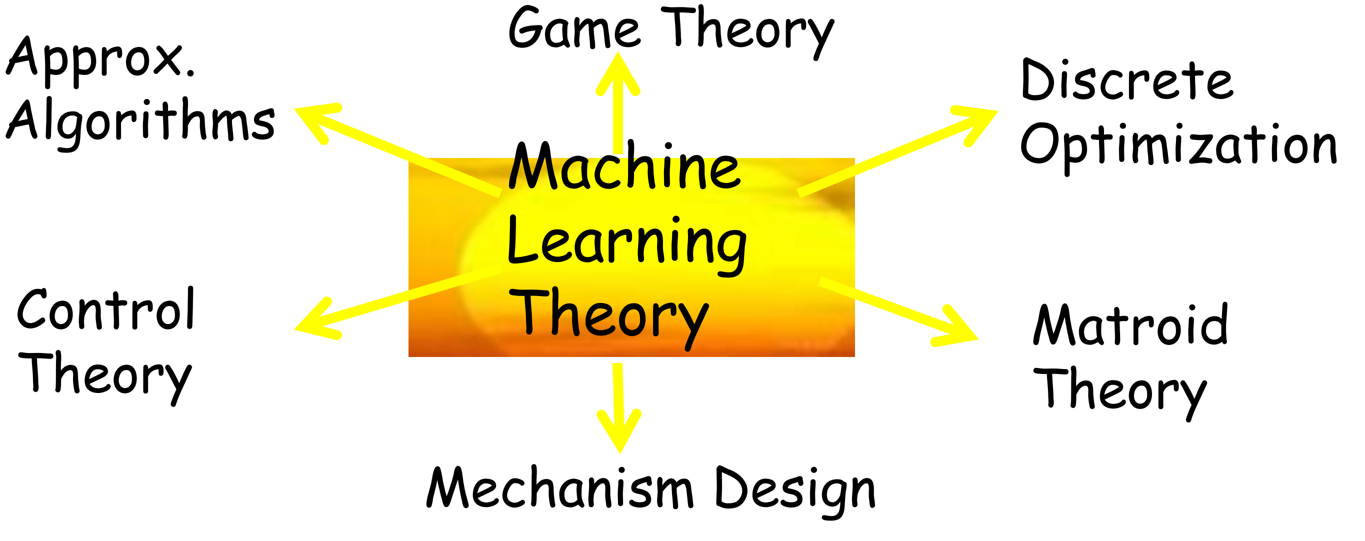
Maria-Florina Balcan: Nina



- Foundations for Modern Machine Learning
- E.g., interactive, semi-supervised, distributed, multi-task, never-ending, privacy preserving learning



- Connections between learning theory & other fields (algorithms, algorithmic game theory)



- Program Committee Chair for ICML 2016 (main general machine learning conference), COLT 2014 (main learning theory conference)

Sarah Schultz
(Assistant Lecturer)

sschultz@cs.cmu.edu

GHC 8110

Research Interests:

Educational data mining and Intelligent Tutoring Systems

Ellen Vitercik

Email: vitercik@cs.cmu.edu

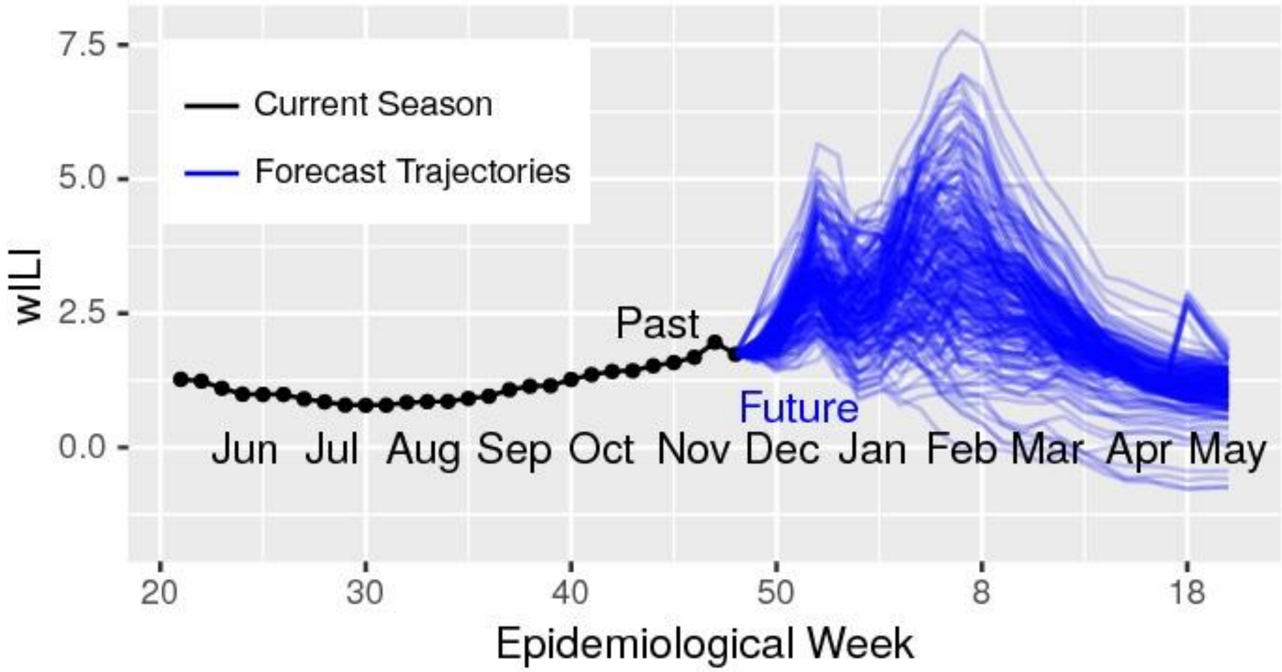
Office hours: Friday 10-11 in GHC 7511

Research interests:

- Theoretical machine learning
- Computational economics

Logan Brooks (lcbrooks@andrew)

- Office space: GHC 6219
- Office hours: Monday 10-11
- Research topic: epidemic forecasting
  - Time series
  - Ensembles



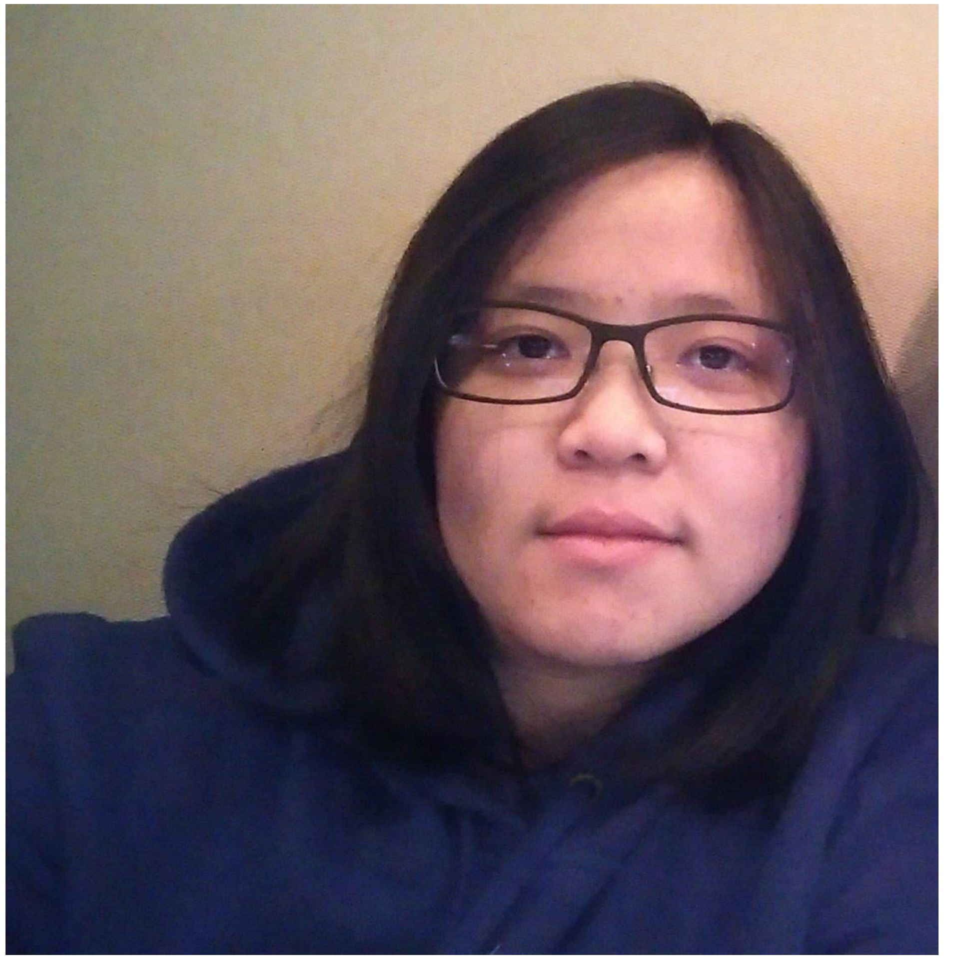
Yujie Xu (yujiex@andrew.cmu.edu)

- GHC 5th floor common area near entrance
- Office Hours: Mon 4:30-5:30
- Research topic:
  data-driven building energy models
  - regression
  - impact evaluation

Easwaran Ramamurthy
eramamur@andrew.cmu.edu

- Find Me: GHC 7405
- Office Hours: Tuesday 4-5
- Interests:
  - Computational Genomics
  - Deep learning applications in regulatory genomics
  - Alzheimer’s Disease

Chieh Lin (chiehl1@cs.cmu.edu)



Office:

GHC 8021

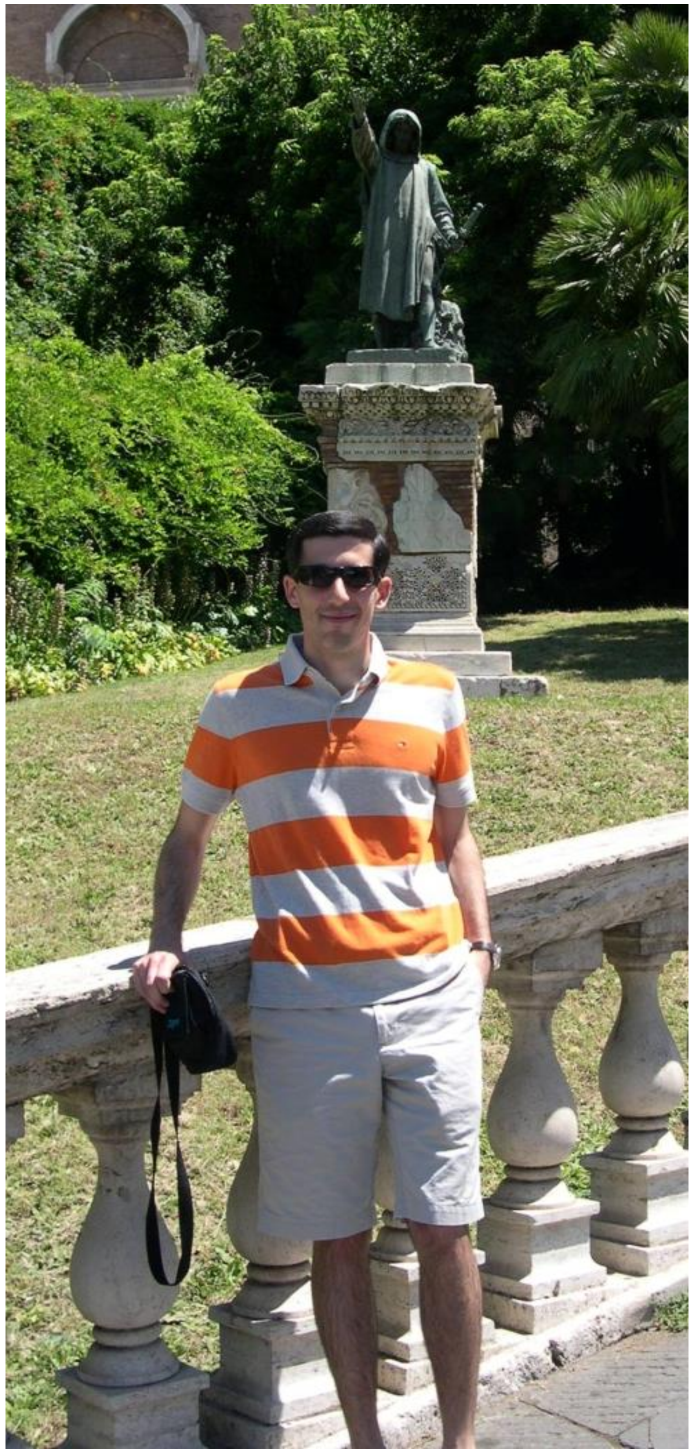
Office Hours:

Thursday 10:30-11:30

Research Interest:

1. ML Applications in biological/medical Data
2. Neural Networks/Deep Learning

Matt Oresky (moresky@andrew.cmu.edu)



Office Hours:
Tuesday 9:30 – 10:30 AM
GHC 6th floor common area
(by the kitchenette)

Interest:
Natural Language Processing

Akash Ramachandran (akashr1@andrew.cmu.edu)

- Office hours : Friday 3-4pm
- Interests:
  - Application of ML in Biology
  - Software Development in Java
  - Playing the *tabla* (an Indian drum)

Guoquan Zhao (guoquanz@andrew.cmu.edu)

Find me: GHC 6th floor common area

Office Hours:

Thursday 3.30 pm – 4.30 pm

Interest:

Active Learning

Distributed ML system

- 8/28 Introduction, MLE
- 8/30 Classification, KNN
- 9/4 – no class, labor day
- 9/6 – Decision trees / problem set 1 out
- 9/11 – Naïve Bayes
- 9/13 – Linear regression
- 9/18 – Logistic regression
- 9/20 – Graphical Models, MRF/ PS1 due, PS2 out
- 9/25 – Gr
- 9/27 –Graphical Models, BN 2
- 10/2 – Perceptron
- 10/4 – Kernel Methods/ PS2 due, PS3 out
- 10/9 – Support Vector Machines
- 10/11 – Neural networks 1: Backpropagation
- 10/16– Neural networks 2: Deep NN/ project proposals due
- 10/18 – Ensemble Learning, Boosting / PS3 due
- 10/23 – Active Learning
- 10/25 Midterm/ PS4 out
- 10/30 – Dimensionality Reduction
- 11/1 – Unsupervised learning (clustering)
- 11/6 – Semi supervised learning
- 11/8 - Generalization, overfitting I / PS 4 due, PS 5 out
- 11/13 – Model Selection.
- 11/15 – Hidden markov models – learning
- 11/20 – HMM – inference
- 11/22 – no class, thanksgiving break
- 11/27 – MDPS
- 11/29 –Reinforcement Learning / PS 5 due
- 12/4 – Distributed ML?
- 12/6 – Final review

10/25 (Wednesday): Midterm

Intro and classification (A.K.A. 'supervised learning')

Graphical models

Non linear and kernel methods

Unsupervised learning

Theoretical considerations

Reasoning under uncertainty

Grading

- **5 Problem sets (5th has a higher weight) - 45%**
- **Final - 30%**
- **Midterm - 20%**
- **Class participation - 5%**

Class assignments

- 5 Problem sets

  - Most containing both theoretical and programming assignments

  - Last problems set: mini project

- Exams

  - Midterm (10/25)

  - Final

Recitations

  - Twice a week (same content in both)

  - Expand on material learned in class, go over problems from previous classes etc.

What is Machine Learning?

Easy part: Machine

Hard part: Learning

- Short answer: Methods that can help generalize information from the observed data so that it can be used to make better decisions in the future

What is Machine Learning?

Longer answer: The term Machine Learning is used to characterize a number of different approaches for generalizing from observed data:

- Supervised learning
  - Given a set of features and labels learn a model that will predict a label to a new feature set

- Unsupervised learning
  - Discover patterns in data

- Reasoning under uncertainty
  - Determine a model of the world either from samples or as you go along

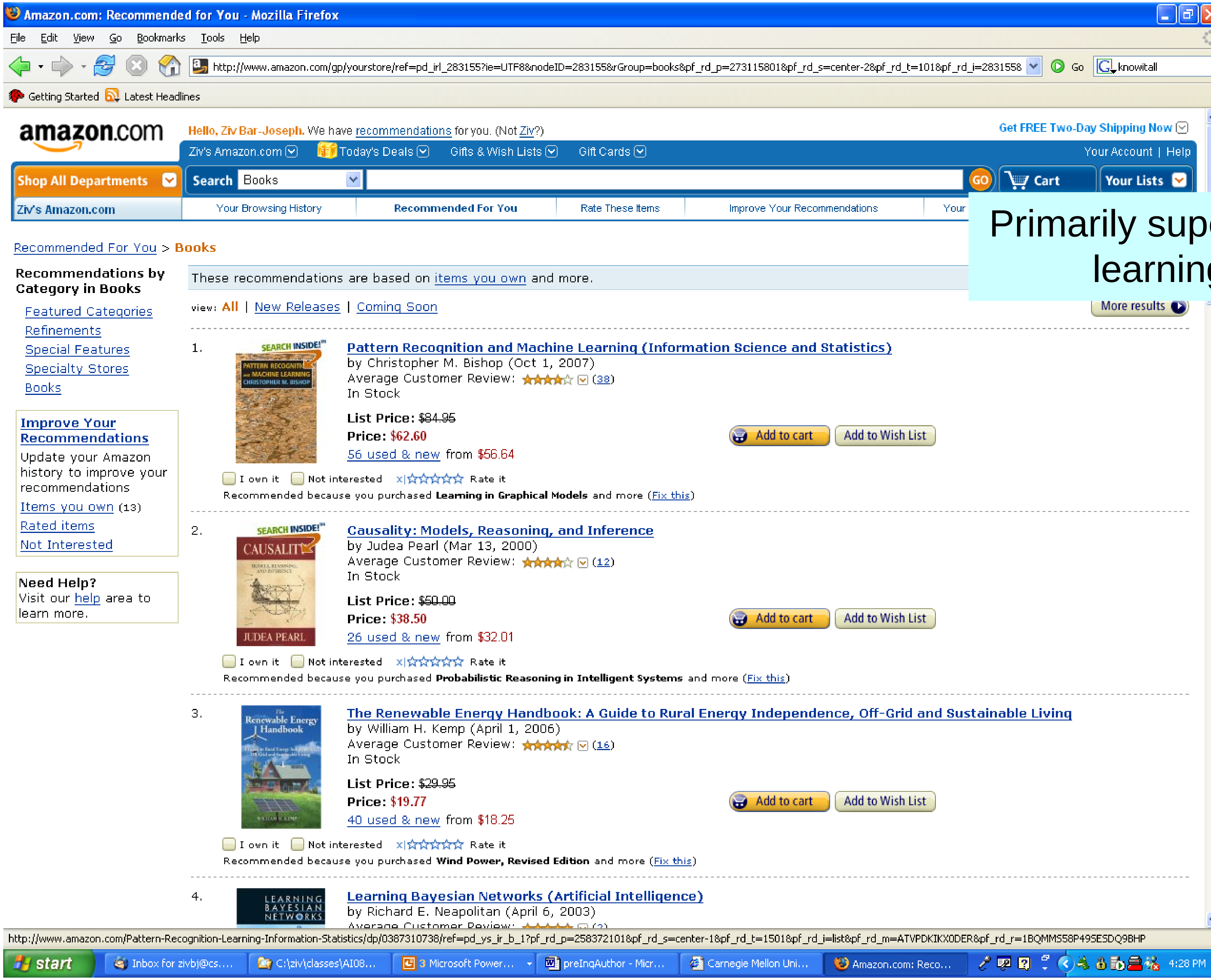
- Active learning
  - Select not only model but also which examples to use

Paradigms of ML

- Supervised learning
  - Given $D = \{X_i,Y_i\}$ learn a model (or function) $F: X_k$ -> Y_k
- Unsupervised learning
  Given $D = \{X_i\}$ group the data into Y classes using a model (or function) $F: X_i$ -> Y_j
- Reinforcement learning (reasoning under uncertainty)
  Given D = {environment, actions, rewards} learn a policy and utility functions:

  policy: $F1: \{e,r\}$ - > a
  utility: $F2: \{a,e\}$- > R
- Active learning
  - Given $D = \{X_i,Y_i\}$, $\{X_j\}$ learn a function $F1$: $\{X_j\}$ -> x_k *to* maximize the success of the supervised learning function $F2: \{X_i , x_k\}$-> Y

Recommender systems



Primarily supervised learning

NELL: Never-Ending Language Learning

Can computers learn to read? We think so. "Read the Web" is a research project that attempts to create a computer system that learns over time to read the web. Since January 2010, our computer system called NELL (Never-Ending Language Learner) has been running continuously, attempting to perform two tasks each day:

- First, it attempts to "read," or extract facts from text found in hundreds of millions of web pages (e.g., `playsInstrument(George_Harrison, guitar)`).
- Second, it attempts to improve its reading competence, so that tomorrow it can extract more facts from the web, more accurately.

semi supervised learning

So far, NELL has accumulated over 50 million candidate beliefs by reading the web, and it is considering thes confidence. NELL has high confidence in 3,938,530 of these beliefs — these are displayed on this website. It is not perfect, but NELL is learning. You can track NELL's progress below or @cmunell on Twitter, browse and download its knowledge base, read more about our technical approach, or join the discussion group.

Recently-Learned Facts twitter

Refresh

instance	iteration	date learned	confidence
glass_window_restoration is a household item	1069	03-aug-2017	97.5
bracelets_curb is a kind of clothing	1069	03-aug-2017	90.9
hillsborough_lista_d_attesa_crea_un_gruppo_meetup is a visualizable thing	1069	03-aug-2017	99.1
parison_levitra_viagra_cialis is a drug	1069	03-aug-2017	97.7
the_democratic_daily is a newspaper	1069	03-aug-2017	100.0
barcelona_international_airport is an airport in the city barcelona	1073	22-aug-2017	100.0
john003 has brother james	1073	22-aug-2017	100.0
omaha_world_herald is a newspaper in the city new_york	1073	22-aug-2017	93.8
abc is a company headquartered in the city new_york	1073	22-aug-2017	100.0
arachnids001 is an arthropod as well as mites also is	1073	22-aug-2017	93.8

Driveless cars

Supervised and
reinforcement learning

Helicopter control

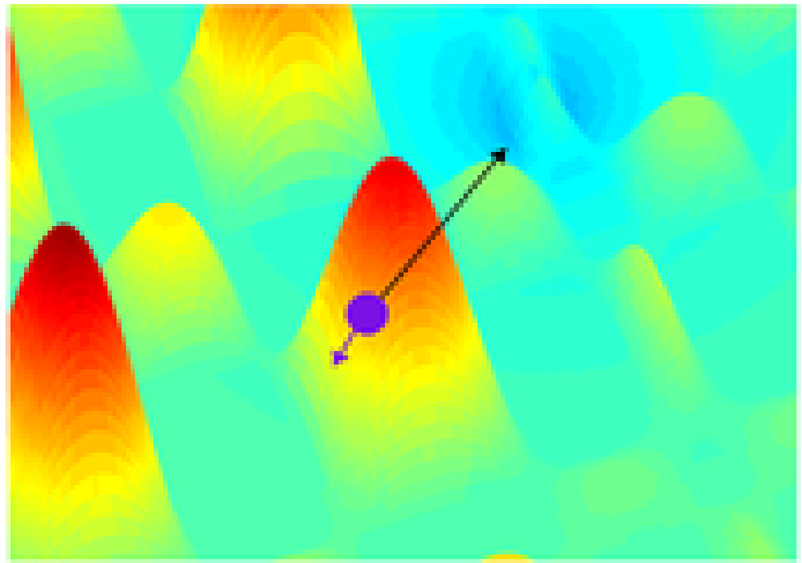
Reinforcement learning

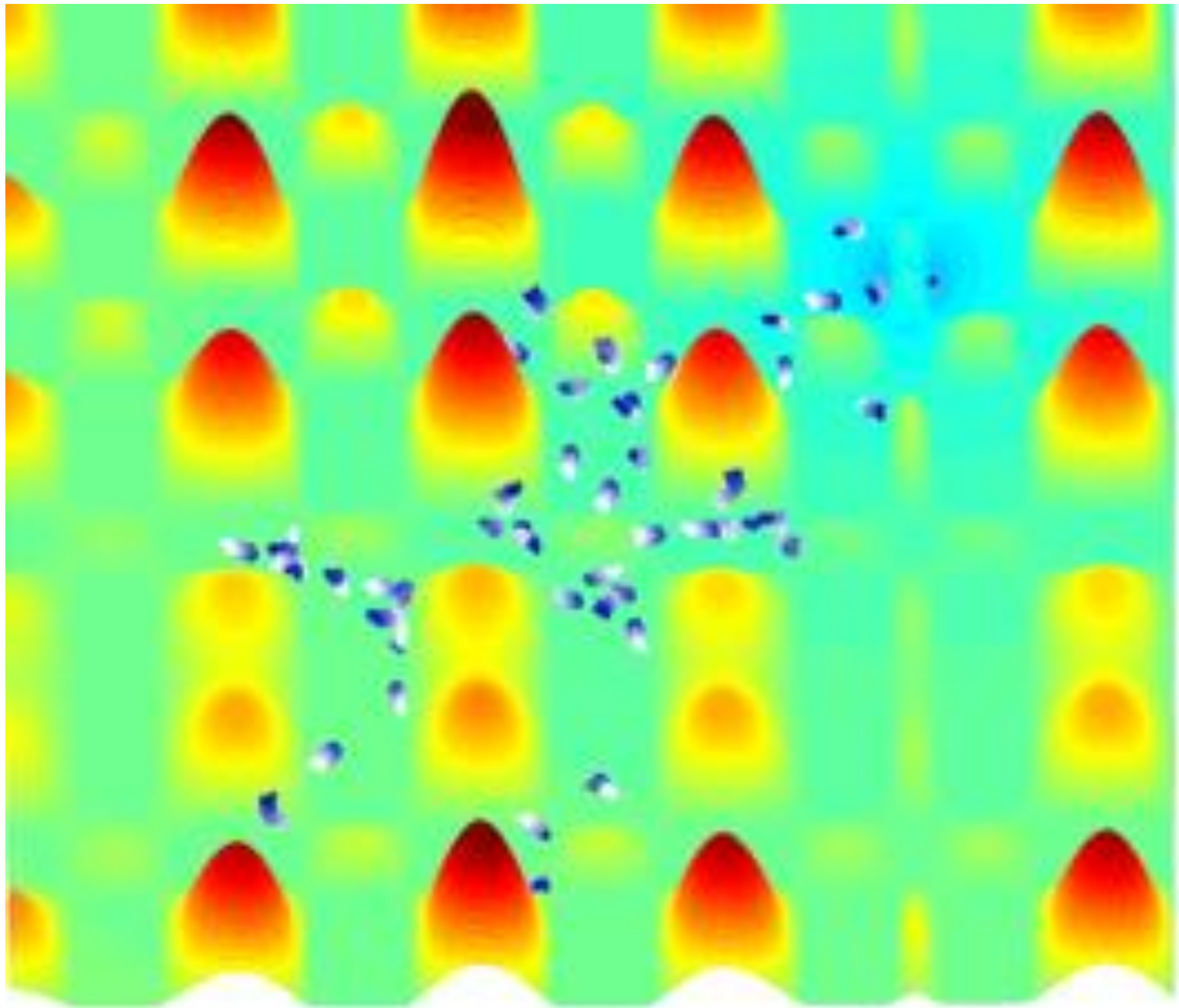
Deep neural networks

Supervised learning (though can also be trained in an unsupervised way)

Distributed gradient descent based on bacterial movement

Reasoning under uncertainty





Biology

```
ACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAACGCTGAGCAATTCGATAGCAATTC
GATAACGCTGAGCAATCGGATAACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAACG
CTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAATCGGATATCGATAGCAATTCGATAAATC
GGATAACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAACGCTGAGCAATTCGATAGC
AATTCGATAACGCTGAGCAATCGGATATCGATAGCAATTCGATAACGCTGAGCAACGCTGAGCA
ATTCGATAGCAATTCGATAACGCTGAGCAATCGGATAACGCTGAGCAATTCGATAGCATTCGAT
AACGCTGAGCAACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAATCGGATAACGCTG
AGCAATTCGATAGCAATTCGATAACGCTGAGCTGAGCAATTCGATAGCAATTCGATAACGCTGA
GCAATCGGATAACGCTGAGCA…TTCGATAGCAATTCGATAACGCTGAGCAACGCTGAGCAATTC
GATAGCAATTCGATAGCAAT…GATAGCAATTCGATAACGCTGAGCAACGCTGAGCAATTCGAT
AGCAATTCGATAACGCTGAG…CAATCGGATAACGCTGAGCAATTCGATAGCAATTCGATAACGCT
GAGCAACGCTGAGCAATTC…GATAGCAATTCGATAACGCTGAGCAATCGGATATCGATAGCAATT
CGATAACGCTGAGCAACG…TGAGCAATTCGATAGCAATTCGATAACGCTGAGCAATCGGATAAC
GC…CGCTGAGCTGAGCAATTCGATAGCAATTCGATAACG
CT…CGATAGCAATTCGATAACGCTGAGCAACGCTGAGCA
AT…AATCGGATATCGATAGCAATTCGATAACGCTGAGCA
ACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAATCGGATAACGCTGAGCAATTCGAT
AGCATTCGATAACGCTGAGCAACGCTGAGCAATTCGATAGCAATTCGATCGGATAACGCTGAGC
AATTCGATAGCAATTCGATAACGCTGAGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCA
ATCGGATAACGCTGAGCAATTCGATAGCA…TGAGCAATTCGAT
AGCAATTCGATAACGCTGAGCAATCGGAT…GAGCAACGCTGA
GCAATTCGATAGCAATTCGATAACGCTGA…TTCGATAGCATTC
GATAACGCTGAGCAACGCTGAGCAATTCG…CAATCGGATAACG
CTGAGCAATTCGATAGCAATTCGATAACG…ATTCGATAACGC
TGAGCAATCGGATAACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAACGCTGAGCAA
TTCGATAGCAATTCGATAGCAATTCGATAGCAATTCGATAACGCTGAGCAACGCTGAGCAATTC
GATAGCAATTCGATAACGCTGAGCAATCGGATAACGCTGAGCAATTCGATAGCAATTCGATAAC
GCTGAGCAACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAATCGGATATCGATAGCA
ATTCGATAACGCTGAGCAACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAATCGGAT
AACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCTGAGCAATTCGATAGCAATTCGATA
ACGCTGAGCAATCGGA
```

Which part is the gene?

Supervised and unsupervised learning (can also use active learning)

Common Themes

- Mathematical framework

  - Well defined concepts based on explicit assumptions

- Representation

  - How do we encode text? Images?

- Model selection

  - Which model should we use? How complex should it be?

- Use of prior knowledge

  - How do we encode our beliefs? How much can we assume?

(brief) intro to probability

Basic notations

- Random variable

  - referring to an element / event whose status is unknown:

    A = “it will rain tomorrow”

- Domain (usually denoted by Ω)

  - The set of values a random variable can take:

    - “A = The stock market will go up this year”: Binary

    - “A = Number of Steelers wins in 2015”: Discrete

    - “A = % change in Google stock in 2015”: Continuous

Axioms of probability (Kolmogorov's axioms)

A variety of useful facts can be derived from just three axioms:

1. $0 \leq P(A) \leq 1$

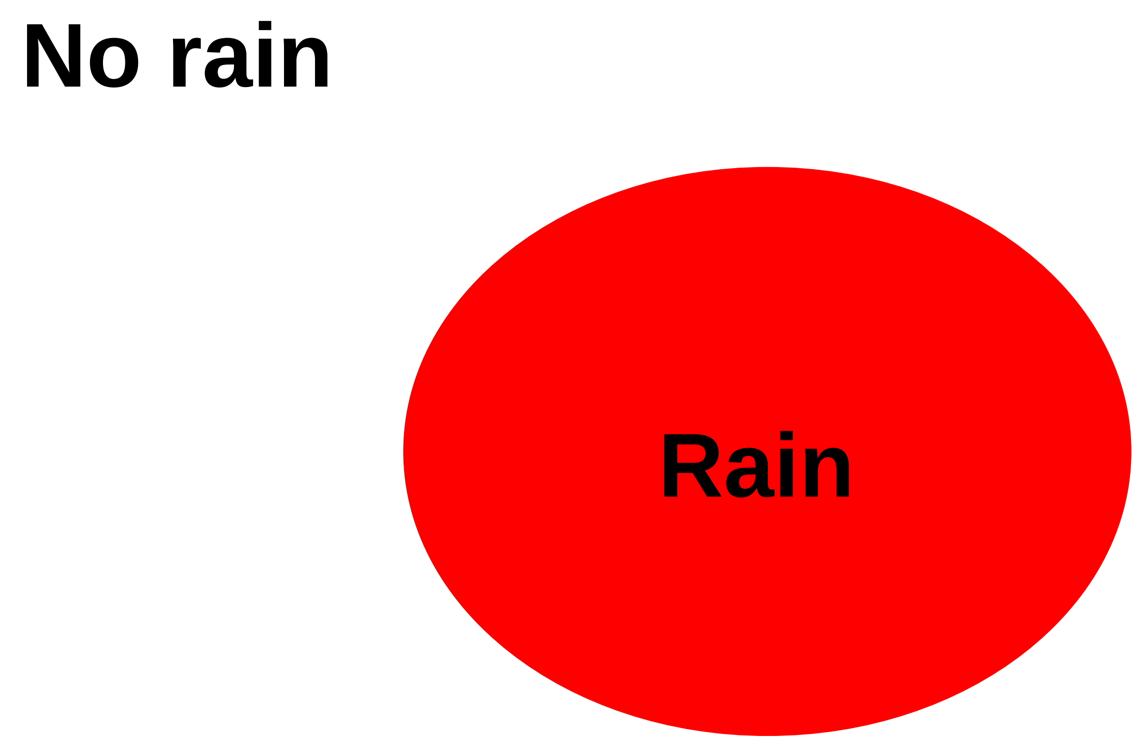
2. $P(true) = 1$, $P(false) = 0$

3. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

There have been several other attempts to provide a foundation for probability theory. Kolmogorov's axioms are the most widely used.

Priors

Degree of belief in an event in the absence of any other information



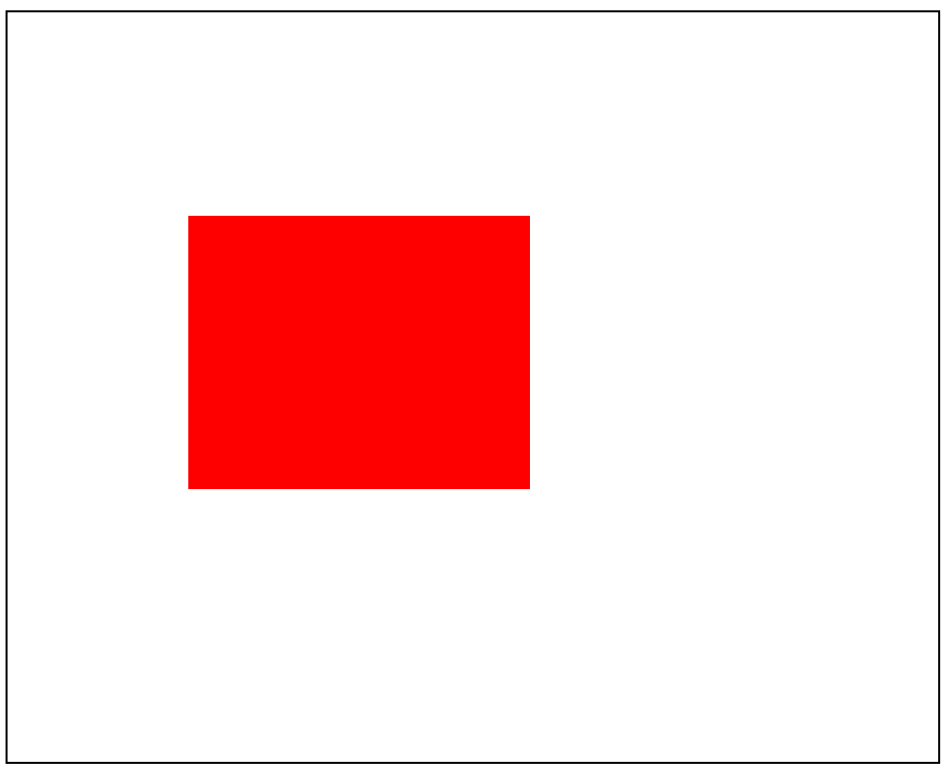
P(rain tomorrow) = 0.2

P(no rain tomorrow) = 0.8

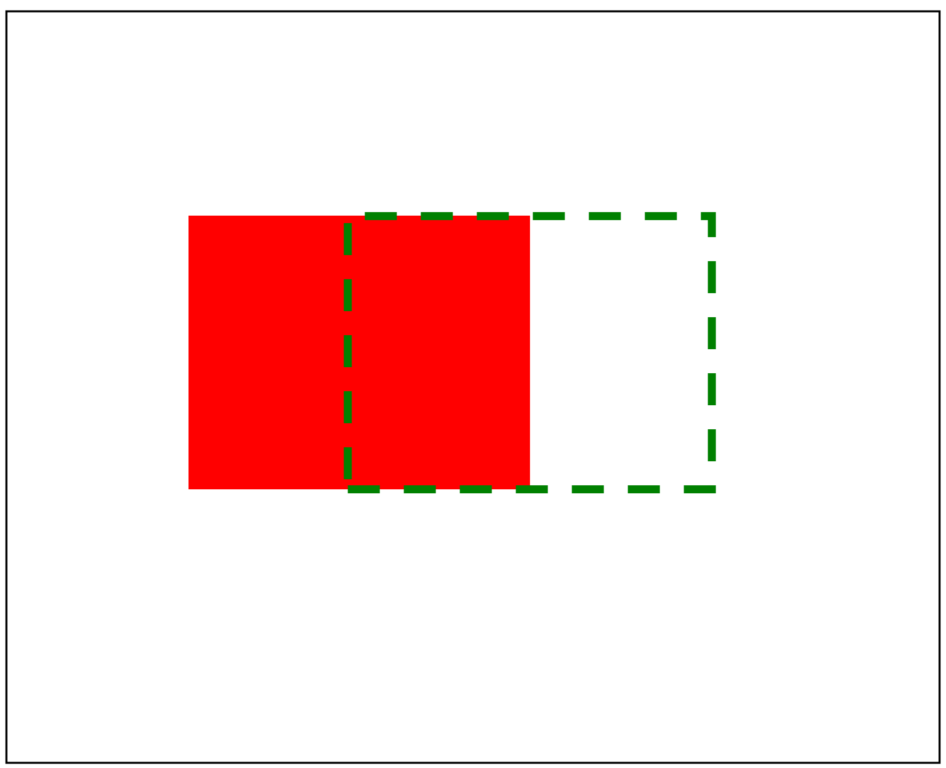
Conditional probability

- P(A = 1 | B = 1): The fraction of cases where A is true if B is true

P(A = 0.2)



P(A|B = 0.5)



Conditional probability

- In some cases, given knowledge of one or more random variables we can improve upon our prior belief of another random variable
- For example:

p(slept in movie) = 0.5

p(slept in movie | liked movie) = 1/4

p(didn't sleep in movie | liked movie) = 3/4

Slept	**Liked**
1	0
0	1
1	1
1	0
0	0
1	0
0	1
0	1

Joint distributions

- The probability that a *set* of random variables will take a specific value is their joint distribution.
- Notation: $P(A \wedge B)$ or $P(A,B)$
- Example: P(liked movie, slept)

If we assume independence then

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

However, in many cases such an assumption may be too strong (more later in the class)

Joint distribution (cont)

P(class size > 20) = 0.6

P(summer) = 0.4

P(class size > 20, summer) = ?

Evaluation of classes

Size	Time	Eval
30	R	2
70	R	1
12	S	2
8	S	3
56	R	1
24	S	2
10	S	3
23	R	3
9	R	2
45	R	1

Joint distribution (cont)

P(class size > 20) = 0.6

P(summer) = 0.4

P(class size > 20, summer) = 0.1

Evaluation of classes

Size	Time	Eval
30	R	2
70	R	1
12	S	2
8	S	3
56	R	1
24	S	2
10	S	3
23	R	3
9	R	2
45	R	1

Joint distribution (cont)

P(class size > 20) = 0.6

P(eval = 1) = 0.3

P(class size > 20, eval = 1) = 0.3

Size	Time	Eval
30	R	2
70	R	1
12	S	2
8	S	3
56	R	1
24	S	2
10	S	3
23	R	3
9	R	2
45	R	1

Joint distribution (cont)

P(class size > 20) = 0.6

P(eval = 1) = 0.3

P(class size > 20, eval = 1) = 0.3

Evaluation of classes

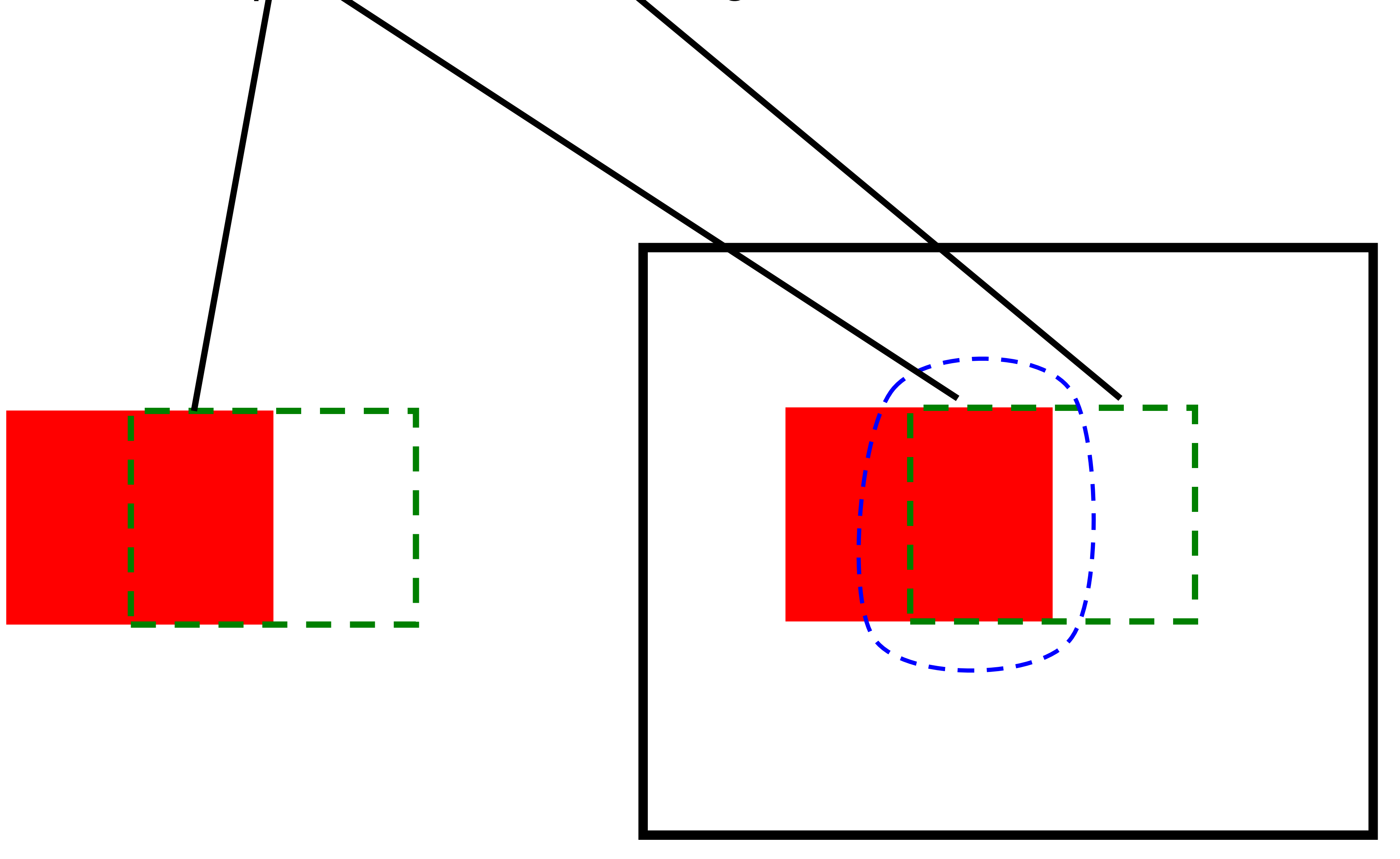
Size	Time	Eval
30	R	2
70	R	1
12	S	2
8	S	3
56	R	1
24	S	2
10	S	3
23	R	3
9	R	2
45	R	1

Chain rule

- The joint distribution can be specified in terms of conditional probability:

  $$P(A,B) = P(A|B)*P(B)$$

- Together with Bayes rule (which is actually derived from it) this is one of the most powerful rules in probabilistic reasoning



Bayes rule

- One of the most important rules for this class.
- Derived from the chain rule:

  P(A,B) = P(A | B)P(B) = P(B | A)P(A)

- Thus,

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

Thomas Bayes was an English clergyman who set out his theory of probability in 1764.

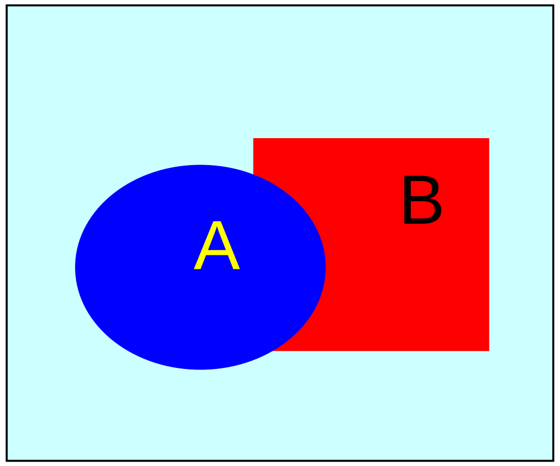
Bayes rule (cont)

Often it would be useful to derive the rule a bit further:

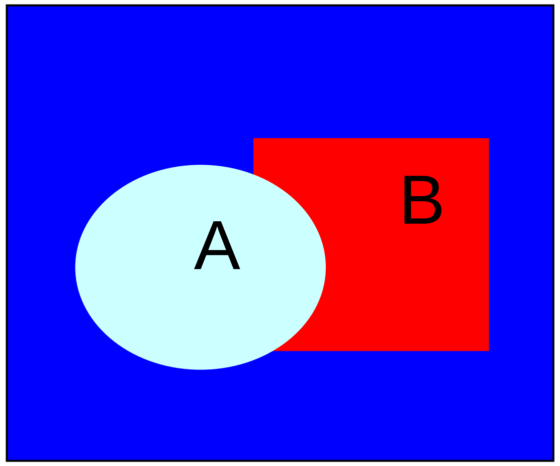
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(B|A)P(A)}{\sum_A P(B|A)P(A)}$$

This results from:
P(B) = $\sum_A$P(B,A)

P(B,A=1)



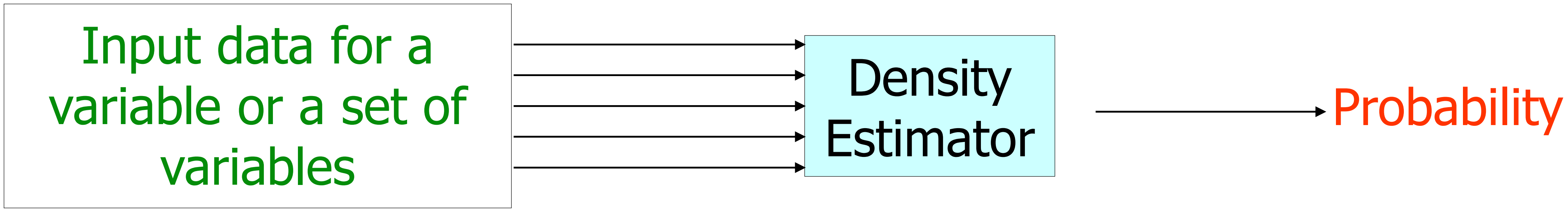
P(B,A=0)



Density estimation

Density Estimation

- A Density Estimator learns a mapping from a set of attributes to a Probability



Density estimation

- Estimate the distribution (or conditional distribution) of a random variable
- Types of variables:

- Binary

coin flip, alarm

- Discrete

dice, car model year

- Continuous

height, weight, temp.,

When do we need to estimate densities?

- Density estimators are critical ingredients in several of the ML algorithms we will discuss
- In some cases these are combined with other inference types for more involved algorithms (i.e. EM) while in others they are part of a more general process (learning in BNs and HMMs)

Density estimation

- Binary and discrete variables:

Easy: Just count!

- Continuous variables:

Harder (but just a bit): Fit a model

Learning a density estimator for discrete variables

$$\hat{P}(x_i = u) = \frac{\#\text{records in which } x_i = u}{\text{total number of records}}$$

A trivial learning algorithm!

But why is this true?

Maximum Likelihood Principle

We can define the likelihood of the data given the model as follows:

$$\hat{P}(\text{dataset} \mid M) = \hat{P}(x_1 \wedge x_2 \ldots \wedge x_n \mid M) = \prod_{k=1}^{n} \hat{P}(x_k \mid M)$$

M is our model (usually a collection of parameters)

For example M is

- The probability of ‘head’ for a coin flip
- The probabilities of observing 1,2,3,4 and 5 for a dice
- etc.

Maximum Likelihood Principle

$$\hat{P}(\text{dataset} \mid M) = \hat{P}(x_1 \wedge x_2 \ldots \wedge x_n \mid M) = \prod_{k=1}^{n} \hat{P}(x_k \mid M)$$

- Our goal is to determine the values for the parameters in *M*
- We can do this by maximizing the probability of generating the observed samples
- For example, let Θ *be the probabilities for a coin flip*
- Then

$$L(x_1, \ldots, x_n \mid \Theta) = p(x_1 \mid \Theta) \ldots p(x_n \mid \Theta)$$

- The observations (different flips) are assumed to be independent
- For such a coin flip with $P(H)=q$ the best assignment for Θ_h is

$$argmax_q = \#H/\#samples$$

- Why?

Maximum Likelihood Principle: Binary variables

- For a binary random variable A with P(A=1)=q
  argmax_q = #1/#samples

- Why?

Data likelihood: $P(D \mid M) = q^{n_1}(1-q)^{n_2}$

We would like to find: $\arg\max_q q^{n_1}(1-q)^{n_2}$

Omitting terms that do not depend on *q*

Maximum Likelihood Principle

Data likelihood: $P(D \mid M) = q^{n_1}(1-q)^{n_2}$

We would like to find: $\arg\max_q q^{n_1}(1-q)^{n_2}$

$$\frac{\partial}{\partial q} q^{n_1}(1-q)^{n_2} = n_1 q^{n_1-1}(1-q)^{n_2} - q^{n_1} n_2 (1-q)^{n_2-1}$$

$$\frac{\partial}{\partial q} = 0 \Rightarrow$$

$$n_1 q^{n_1-1}(1-q)^{n_2} - q^{n_1} n_2 (1-q)^{n_2-1} = 0 \Rightarrow$$

$$q^{n_1-1}(1-q)^{n_2-1}(n_1(1-q) - q n_2) = 0 \Rightarrow$$

$$n_1(1-q) - q n_2 = 0 \Rightarrow$$

$$n_1 = n_1 q + n_2 q \Rightarrow$$

$$q = \frac{n_1}{n_1 + n_2}$$

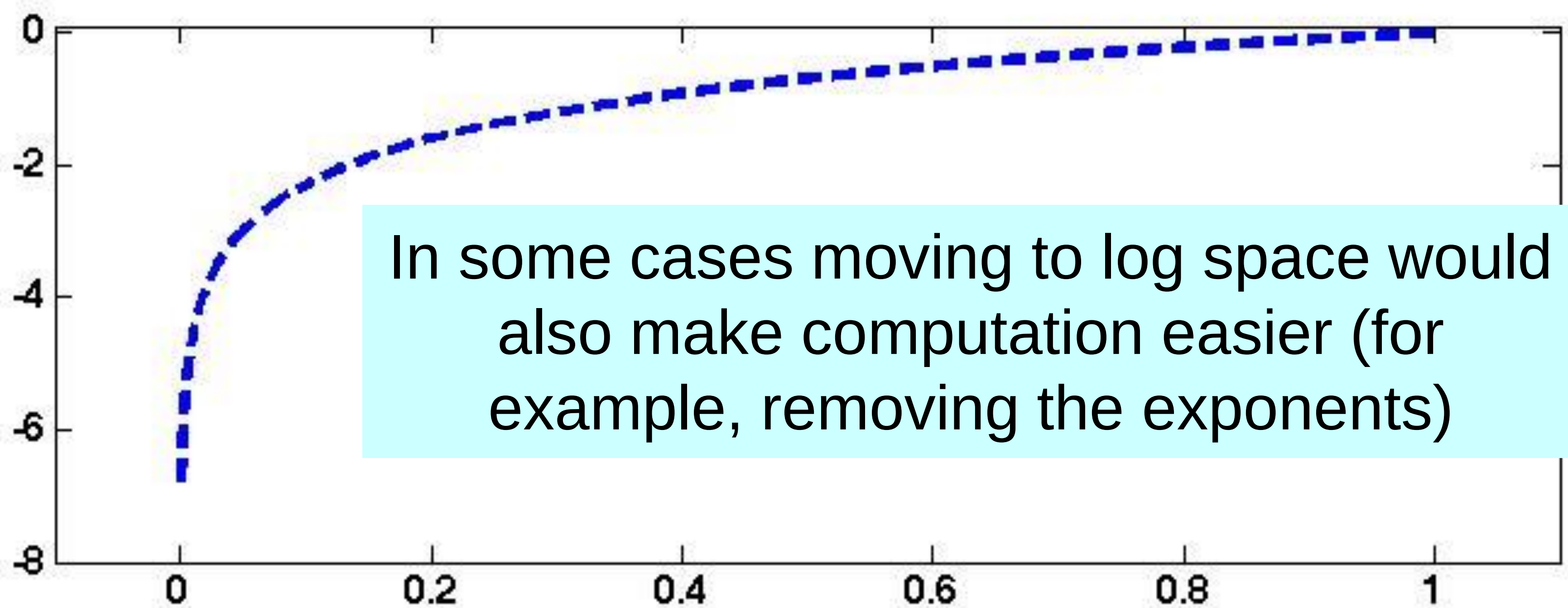
Log Probabilities

When working with products, probabilities of entire datasets often get too small. A possible solution is to use the log of probabilities, often termed ‘log likelihood’

$$\log \hat{P}(\text{dataset} \mid M) = \log \prod_{k=1}^{n} \hat{P}(x_k \mid M) = \sum_{k=1}^{n} \log \hat{P}(x_k \mid M)$$

Maximizing this likelihood function is the same as maximizing P(dataset | M)

Log values between 0 and 1



In some cases moving to log space would also make computation easier (for example, removing the exponents)

How much do grad students sleep?

- Lets try to estimate the distribution of the time students spend sleeping (outside class).

Possible statistics

- **X**

Sleep time

- **Mean of X:**

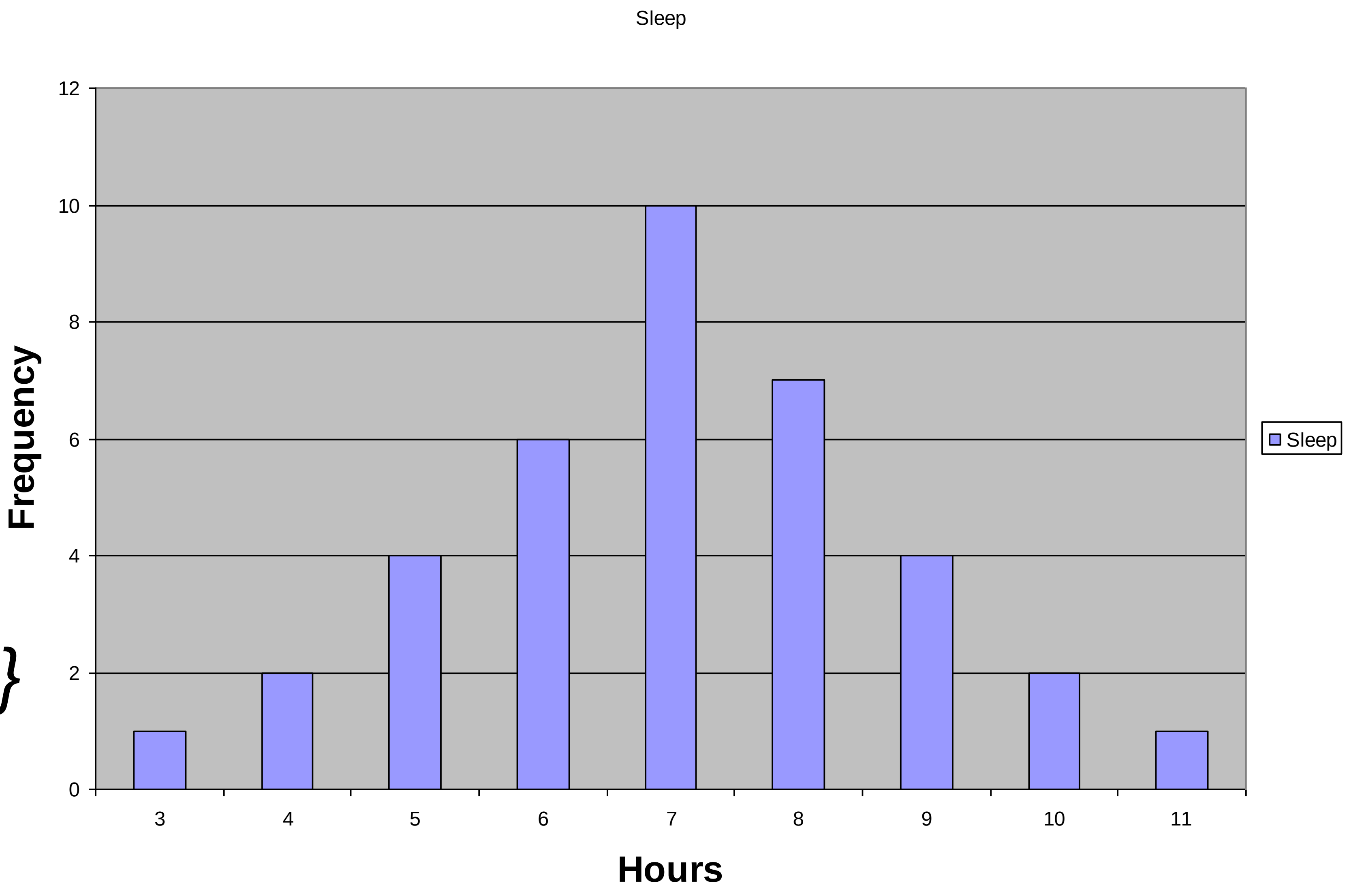
E{X}

7.03

- **Variance of X:**

Var{X} = E{(X-E{X})^2}

3.05

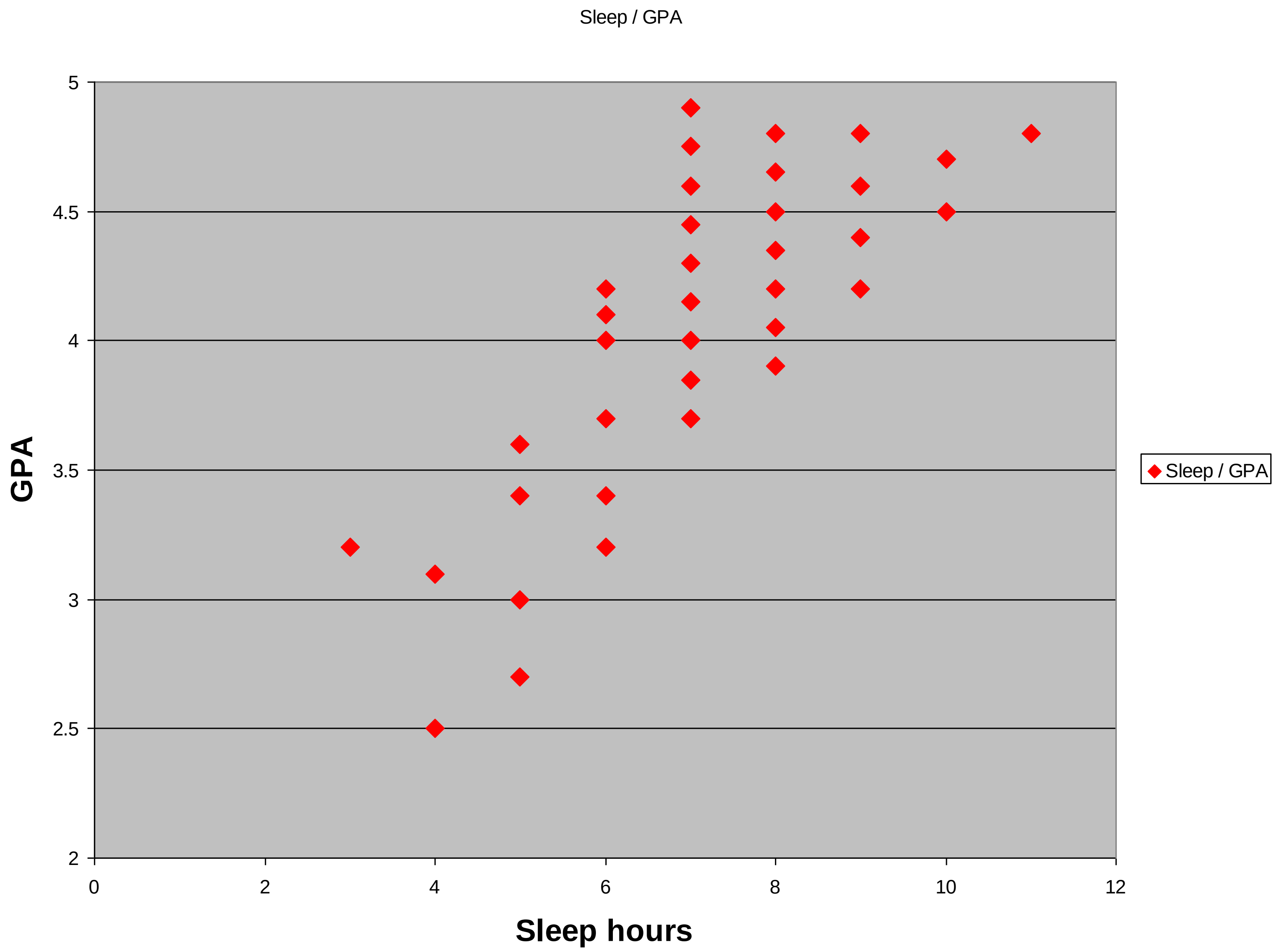


Covariance: Sleep vs. GPA

- **Co-Variance of X1, X2:**

Covariance{X1,X2} = E{(X1-E{X1})(X2-E{X2})}

= 0.88



Statistical Models

• Statistical models attempt to characterize properties of the population of interest

• For example, we might believe that repeated measurements follow a normal (Gaussian) distribution with some mean μ and variance σ^2 , x ~ N(μ,σ^2)

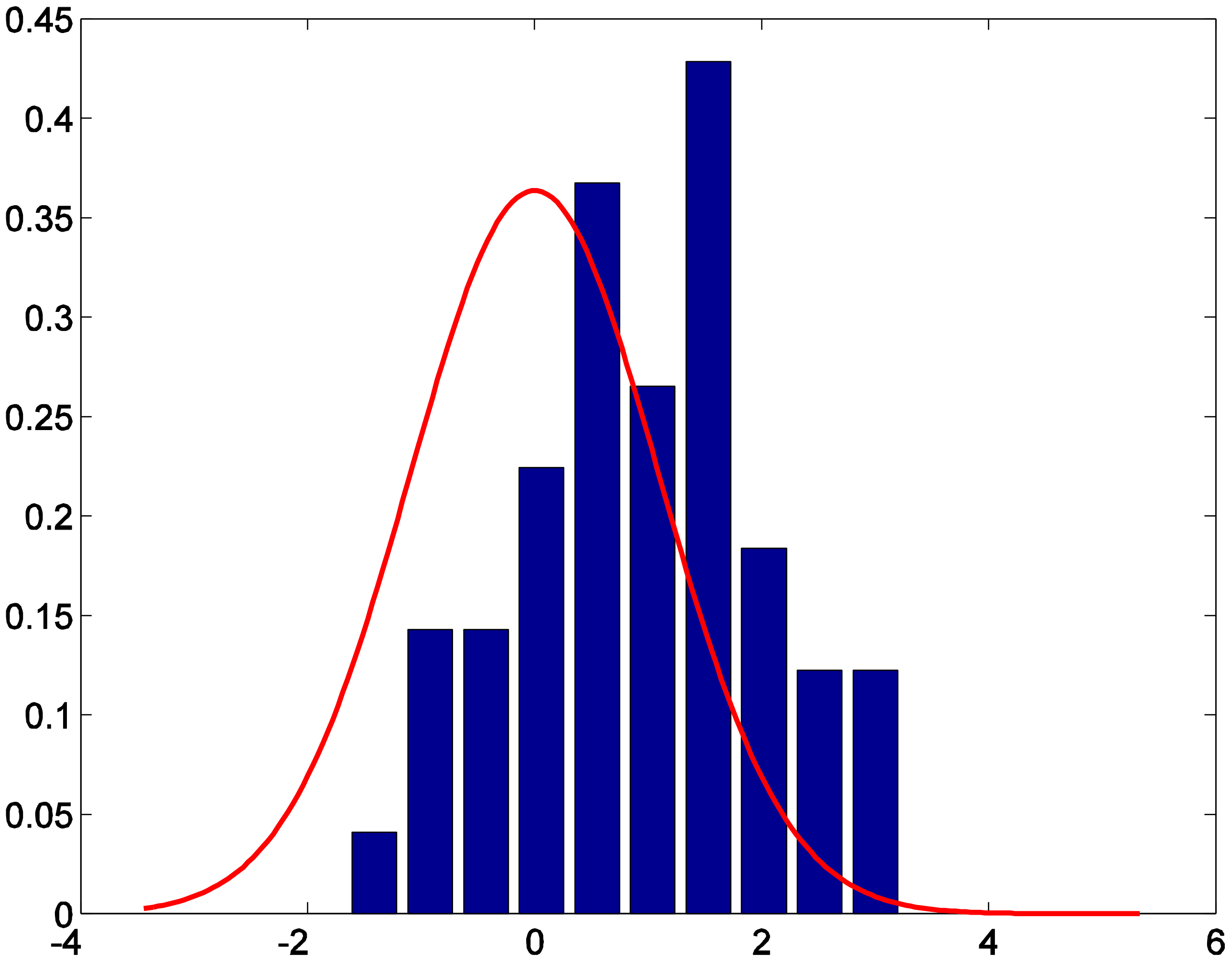
where

$$p(x\,|\,\Theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

and $\Theta=(\mu,\sigma^2)$ defines the parameters (mean and variance) of the model.

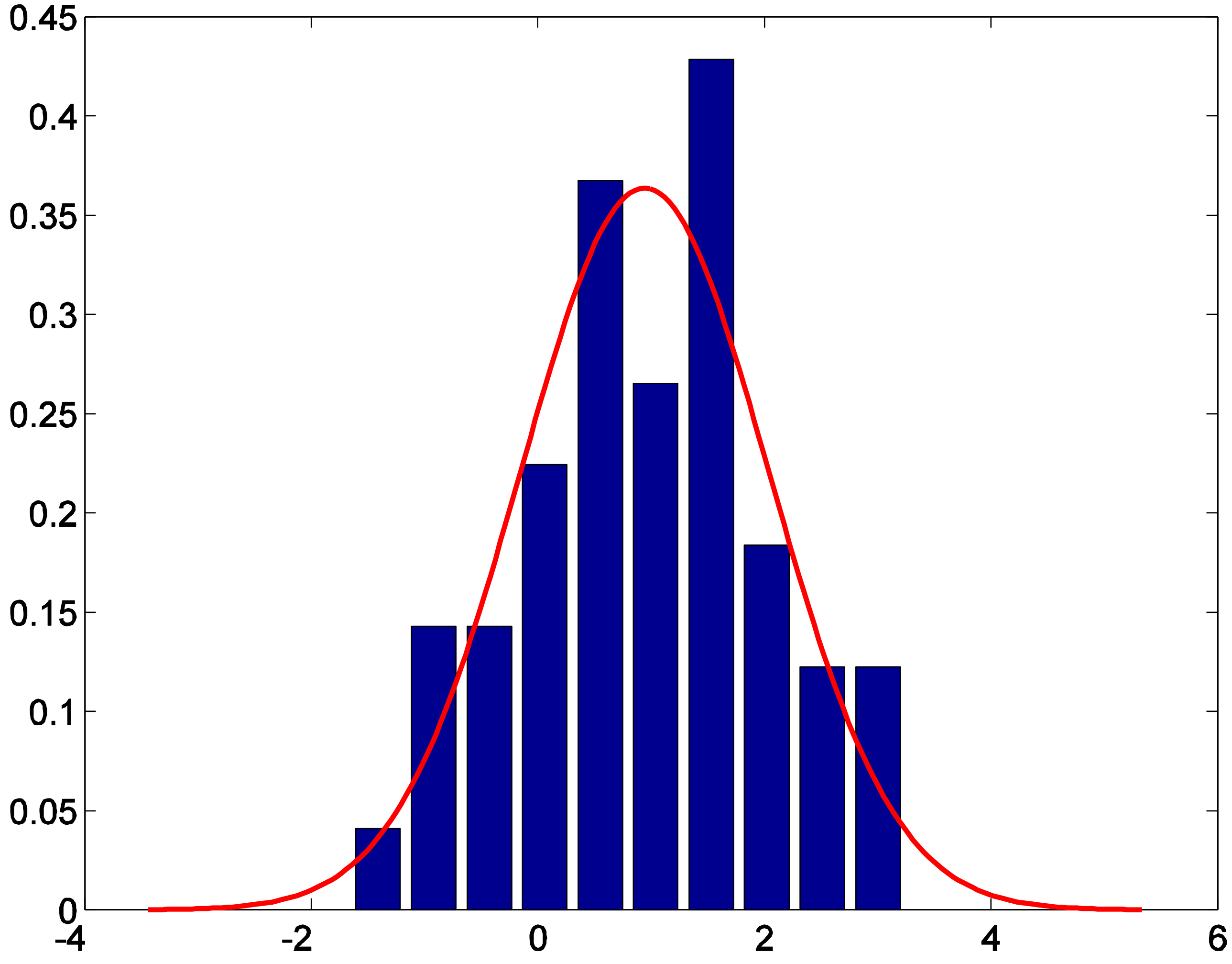
The Parameters of Our Model

- A statistical model is a **collection** of distributions; the **parameters** specify individual distributions x ~ N(μ,σ^2)
- We need to adjust the parameters so that the resulting distribution **fits** the data well



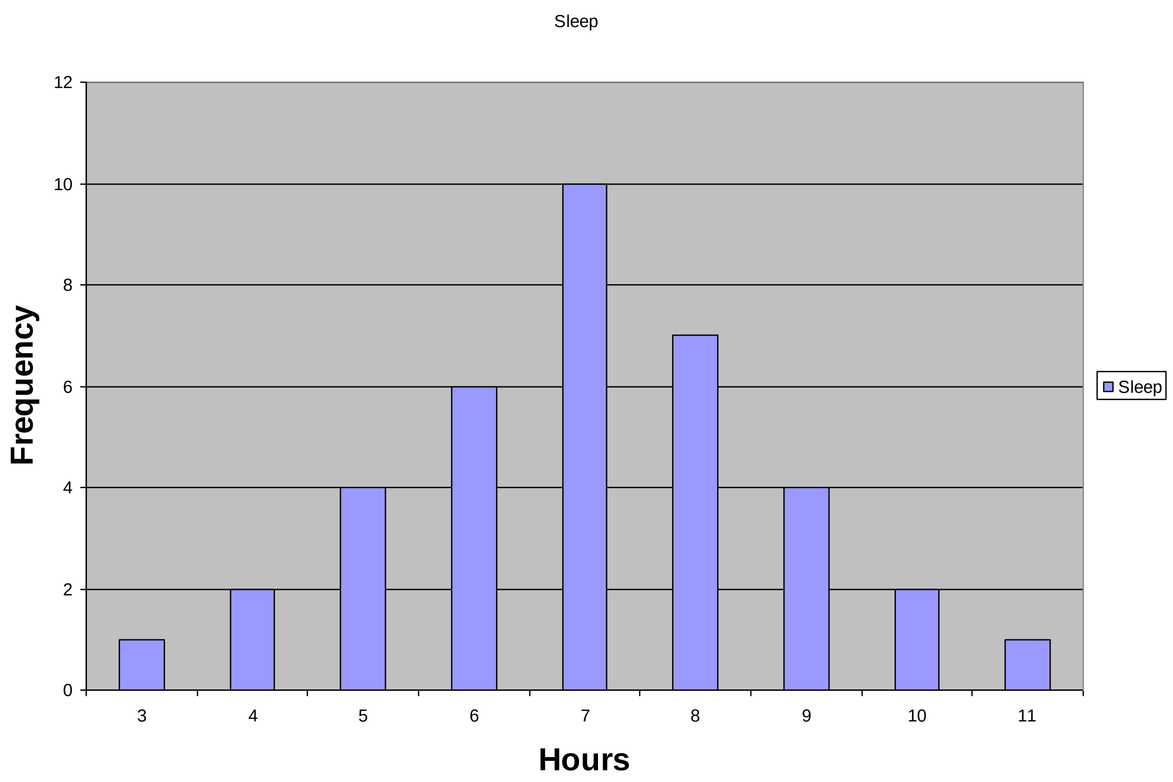
The Parameters of Our Model

- A statistical model is a **collection** of distributions; the **parameters** specify individual distributions $x \sim N(\mu, \sigma^2)$
- We need to adjust the parameters so that the resulting distribution **fits** the data well



Computing the parameters of our model

- Lets assume a Guassian distribution for our sleep data
- How do we compute the parameters of the model?



Maximum Likelihood Principle

• We can fit statistical models by maximizing the probability of generating the observed samples:
$L(x_1, \ldots, x_n \mid \Theta) = p(x_1 \mid \Theta) \ldots p(x_n \mid \Theta)$
(the samples are assumed to be independent)

• In the Gaussian case we simply set the mean and the variance to the sample mean and the sample variance:

$$\overline{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \overline{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{\mu})^2$$

Why?

Density estimation

- Binary and discrete variables:

Easy: Just count!

- Continuous variables:

Harder (but just a bit): Fit a model

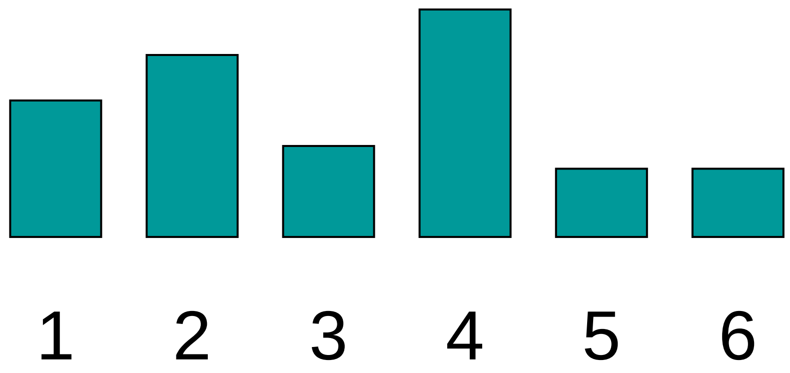
But what if we only have very few samples?

Important points

- Random variables
- Chain rule
- Bayes rule
- Joint distribution, independence, conditional independence
- MLE

Probability Density Function

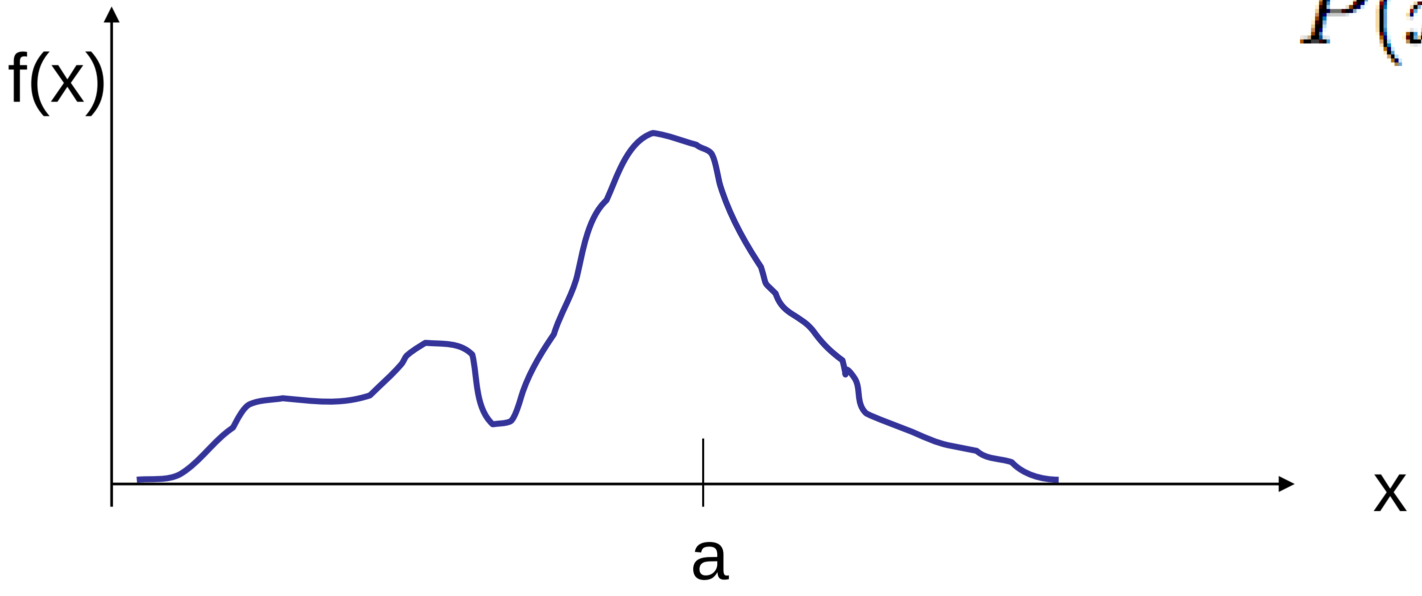
- Discrete distributions



$$\sum_i P(X = x_i) = 1$$

- Continuous: Cumulative Density Function (CDF): *F(a)*

$$P(x \leq a) = \int_{-\infty}^{a} f(\tau) d\tau$$

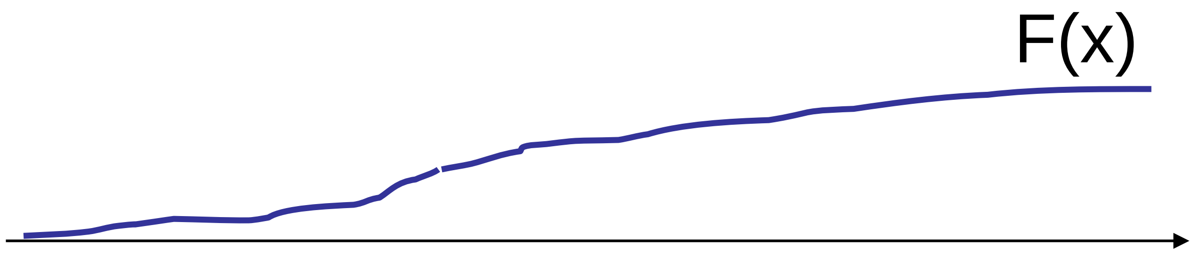


Cumulative Density Functions

- Total probability
$$P(\Omega) = \int_{-\infty}^{\infty} f(x)dx = 1$$
- Probability Density Function (PDF)
$$\frac{d}{dx}F(x) = f(x)$$
- Properties:
$$P(a \leq x \leq b) = \int_b^a f(x)dx = F(b) - F(a)$$

$$\lim_{x \to -\infty} F(x) = 0$$

$$\lim_{x \to \infty} F(x) = 1$$



$$F(a) \geq F(b)\ \forall a \geq b$$

Expectations

- Mean/Expected Value:

$$E[x] = \bar{x} = \int x f(x) dx$$

- Variance:

$$Var(x) = E[(x - \bar{x})^2] = E[x^2] - (\bar{x})^2$$

- In general:

$$E[x^2] = \int x^2 f(x) dx$$

$$E[g(x)] = \int g(x) f(x) dx$$

Multivariate

- Joint for (x,y)

$$P\left((x,y)\in A\right) = \int\int_A f(x,y)dxdy$$

- Marginal:

$$f(x) = \int f(x,y)dy$$

- Conditionals:

$$f(x|y) = \frac{f(x,y)}{f(y)}$$

- Chain rule:

$$f(x,y) = f(x|y)f(y) = f(y|x)f(x)$$

Bayes Rule

- Standard form:

$$f(x|y) = \frac{f(y|x)f(x)}{f(y)}$$

- Replacing the bottom:

$$f(x|y) = \frac{f(y|x)f(x)}{\int f(y|x)f(x)dx}$$

Binomial

- Distribution:

$$x \sim Binomial(p, n)$$

$$P(x = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

- Mean/Var:

$$E[x] = np$$

$$Var(x) = np(1-p)$$

Uniform

- Anything is equally likely in the region [a,b]

- Distribution:

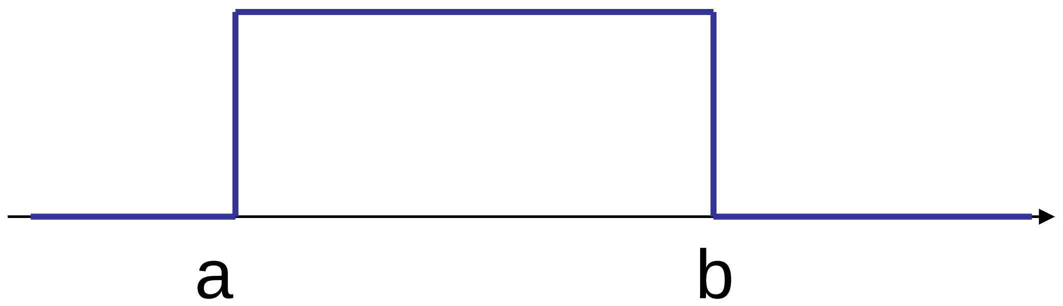
$$x \sim U(a, b)$$

- Mean/Var

$$f(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & otherwise \end{cases}$$

$$E[x] = \frac{a+b}{2}$$

$$Var(x) = \frac{a^2 + ab + b^2}{3}$$



Gaussian (Normal)

- If I look at the height of women in country xx, it will look approximately Gaussian
- Small random noise errors, look Gaussian/Normal

- Distribution:

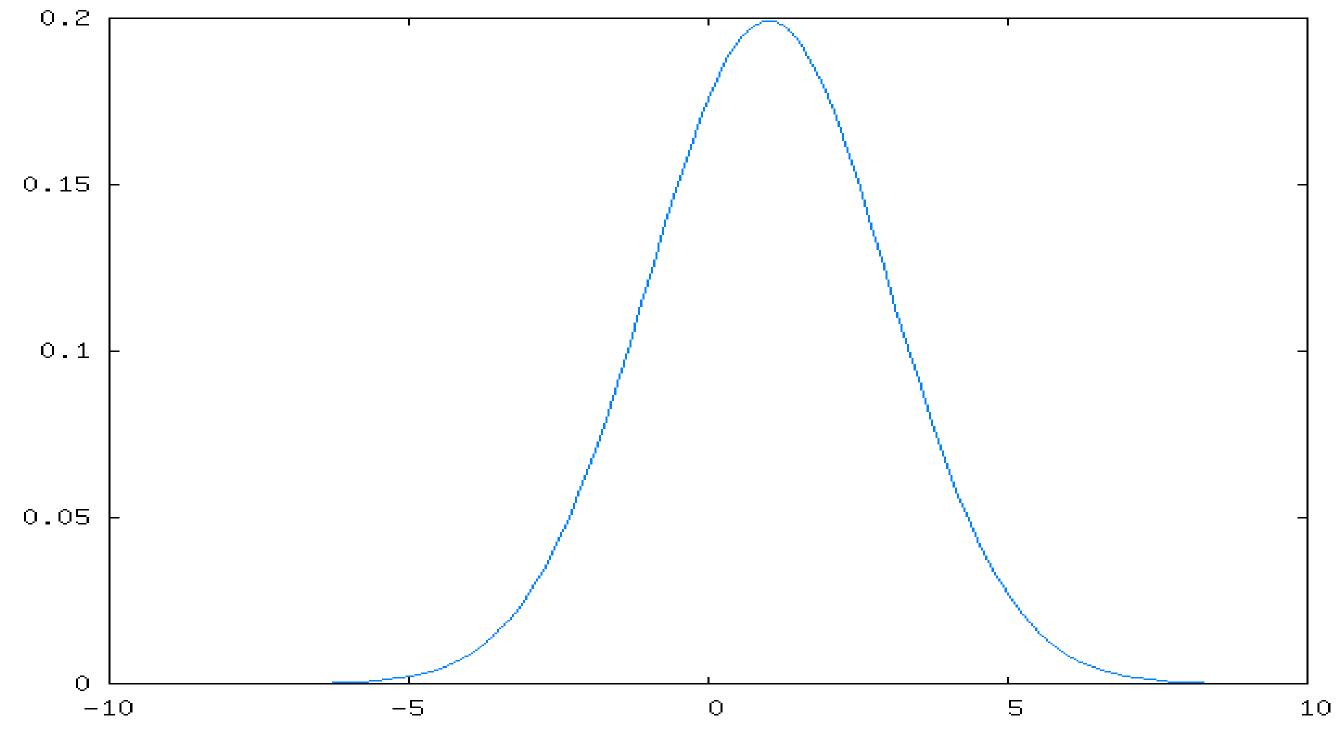
$$x \sim N(\mu, \sigma^2)$$

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

- Mean/var

$$E[x] = \mu$$

$$Var(x) = \sigma^2$$



Why Do People Use Gaussians

- Central Limit Theorem: (loosely)
  - Sum of a large number of IID random variables is approximately Gaussian

Multivariate Gaussians

- Distribution for vector x

$$x = (x_1, \ldots, x_N)^T, \quad x \sim N(\mu, \Sigma)$$

- PDF:

$$f(x) = \frac{1}{(2\pi)^{\frac{N}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

$$E[x] = \mu = (E[x_1], \ldots, E[x_N])^T$$

$$Var(x) \rightarrow \Sigma = \begin{pmatrix} Var(x_1) & Cov(x_1, x_2) & \ldots & Cov(x_1, x_N) \\ Cov(x_2, x_1) & Var(x_2) & \ldots & Cov(x_2, x_N) \\ \vdots & & \ddots & \vdots \\ Cov(x_N, x_1) & Cov(x_N, x_2) & \ldots & Var(x_N) \end{pmatrix}$$

Multivariate Gaussians

$$f(x) = \frac{1}{(2\pi)^{\frac{N}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

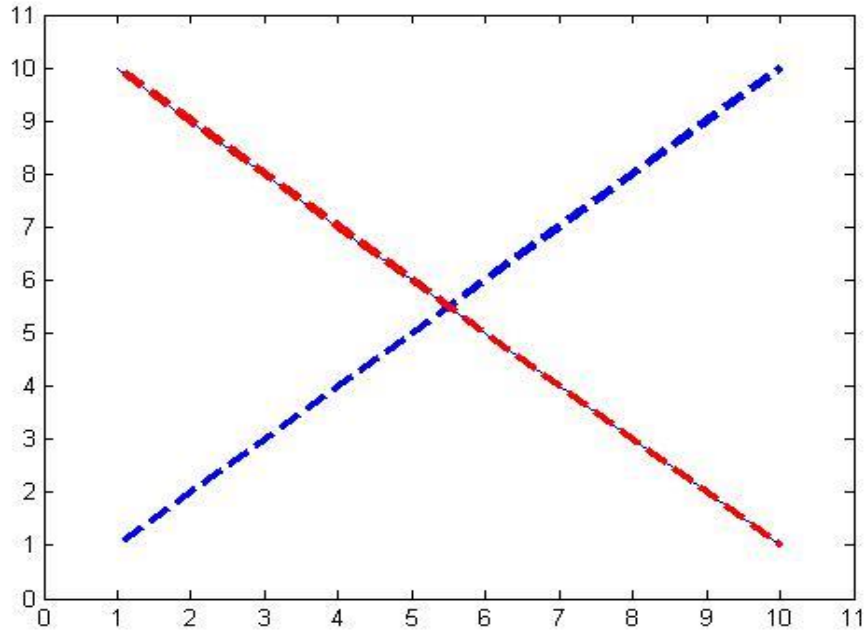
$$E[x] = \mu = (E[x_1], \ldots, E[x_N])^T$$

$$Var(x) \rightarrow \Sigma = \begin{pmatrix} Var(x_1) & Cov(x_1, x_2) & \ldots & Cov(x_1, x_N) \\ Cov(x_2, x_1) & Var(x_2) & \ldots & Cov(x_2, x_N) \\ \vdots & & \ddots & \vdots \\ Cov(x_N, x_1) & Cov(x_N, x_2) & \ldots & Var(x_N) \end{pmatrix}$$

$$\mathrm{cov}(\chi_1, \chi_2) = \frac{1}{n} \sum_{i=1}^{n} (x_{1,i} - \mu_1)(x_{2,i} - \mu_2)$$

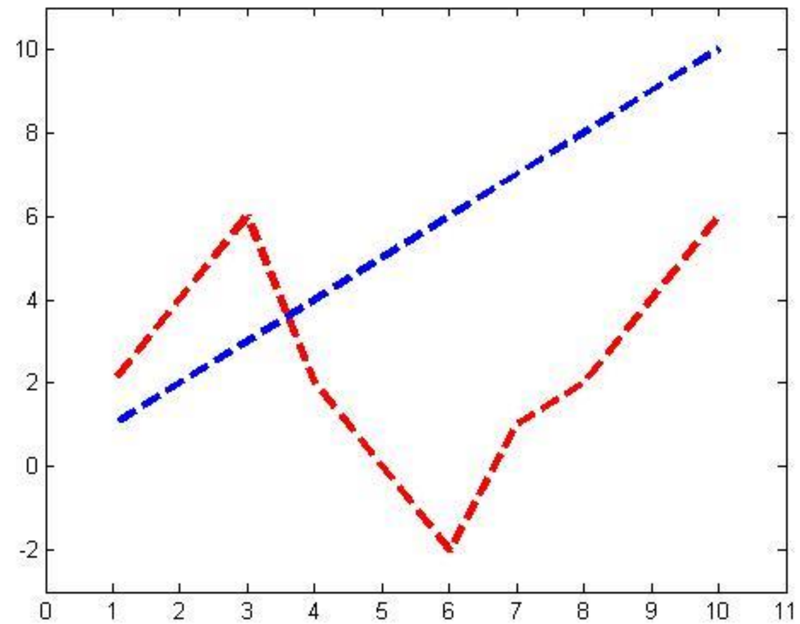
Covariance examples

Anti-correlated



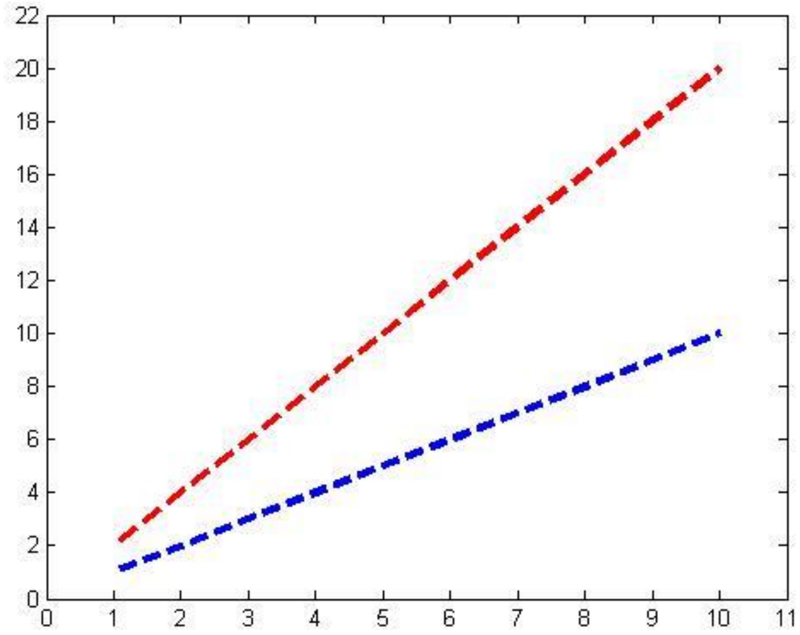
Covariance: -9.2

Independent (almost)



Covariance: 0.6

Correlated



Covariance: 18.33

Sum of Gaussians

- The sum of two Gaussians is a Gaussian:

$$x \sim N(\mu, \sigma^2) \quad y \sim N(\mu_y, \sigma_y^2)$$

$$ax + b \sim N(a\mu + b, (a\sigma)^2)$$

$$x + y \sim N(\mu + \mu_y, \sigma^2 + \sigma_y^2)$$