10-701

Machine Learning

http://www.cs.cmu.edu/~epxing/Class/10701-15F/

Organizational info

- All up-to-date info is on the course web page (follow links from my page).
- Instructors
  - Eric Xing
  - 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. Make sure you are subscribed.

Zhiting Hu

- **Research:** large scale machine learning and their applications in NLP/CV.
- **Homepage:** http://www.cs.cmu.edu/~zhitingh/
- **Contact:** zhitinghu@gmail.com

Mrinmaya Sachan (mrinmays@cs.cmu.edu)

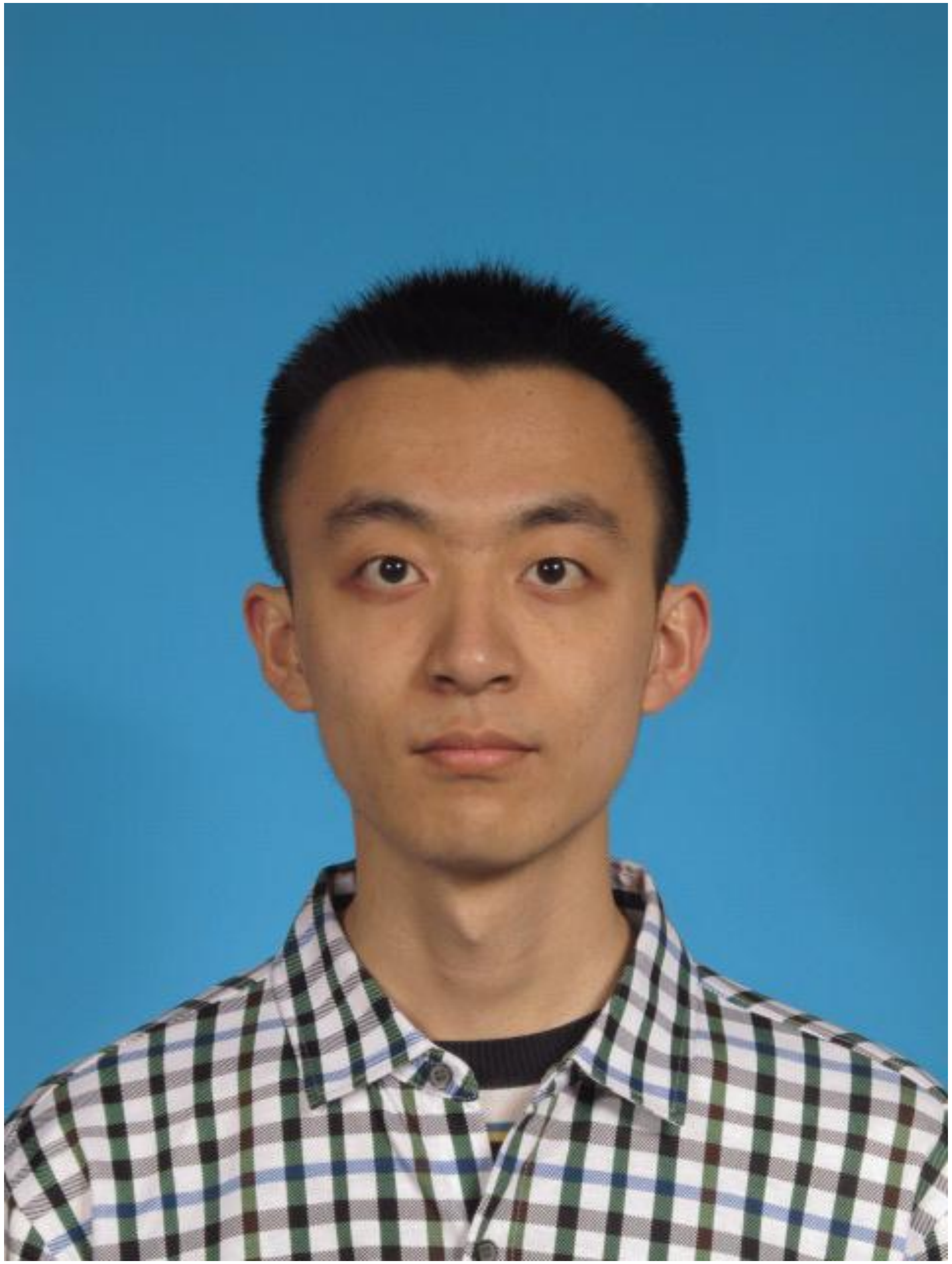
GHC 8013

Office Hours:

Thu 11AM-12Noon

I am interested in:

- Structured Prediction
- NLP



Yuntian Deng

- **Research:** large scale machine learning.
- **Contact:** yuntiand@cs.cmu.edu

Xun Zheng (xunzheng@cs.cmu.edu)

I work on…

- MCMC
- Distributed machine learning

Hao Zhang
(hao@cs.cmu.edu)

Find me: GHC 8116

Office Hours:

Friday 3.30 pm – 4.30 pm

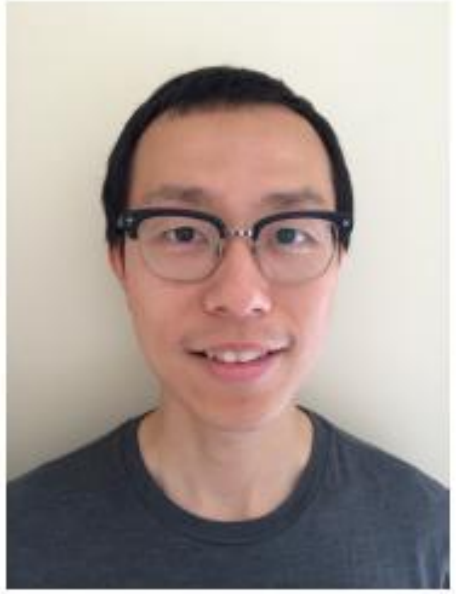
Interest:

Distributed Machine Learning

Deep Learning

Applications in computer vision

Yan Xia



2nd year ML Masters student

Research Interests:

- Machine learning applications in drug discovery and development
- Identifying and modeling biological interactions

- 9/3 Intro to probability, MLE
- 9/8 No class
- 9/10 Classification, KNN
- 9/15 No class, Jewish new year
- 9/17 Decision trees - PS1 out
- 9/22 Naïve Bayes
- 9/24 Linear regression
- 9/26 Logis
- 10/1 Perceptron, Neural networks - PS1 due / PS2 out
- 10/6 Deep learning, SVM1
- 10/10 SVM 2
- 10/13 Evaluating classifiers , Bias – Variance decomposition
- 10/15 Ensemble learning – Boosting, RF PS2 due / PS3 out
- 10/20 Unsupervised learning – clustering
- 10/22 Unsupervised learning – clustering / project proposal due
- 10/27 Semi-supervised learning
- 10/29 Learning theory 1 - PS3 due / PS4 out
- 11/3 PAC learning
- 11/5 Graphical models, BN
- 11/10 – BN
- 11/12 - Undirected graphical models / PS4 due
- 11/17 – Midterm
- 11/19 – HMM – PS5 out
- 11/24 – HMM inference
- 12/1 – MDPs / Reinforcement learning / ps5 due
- 12/3 – Topic models-
- 12/4 - Project poster session
- 12/8 –Computational Biology
- 12/10 – no class

11/17 (Monday): Midterm

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

Clustering ('Unsupervised learning')

Probabilistic representation and modeling ('reasoning under uncertainty')

Applications of ML

Grading

- **5 Problem sets - 40%**
- **Project - 35%**
- **Midterm - 25%**

Class assignments

- 5 Problem sets

  - Most containing both theoretical and programming assignments

- Projects

  - Groups of 1-2

  - Open ended. Would have to submit a proposal based on your interest. We will also provide suggestions on the website.

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) $F: X_k \rightarrow Y_k$

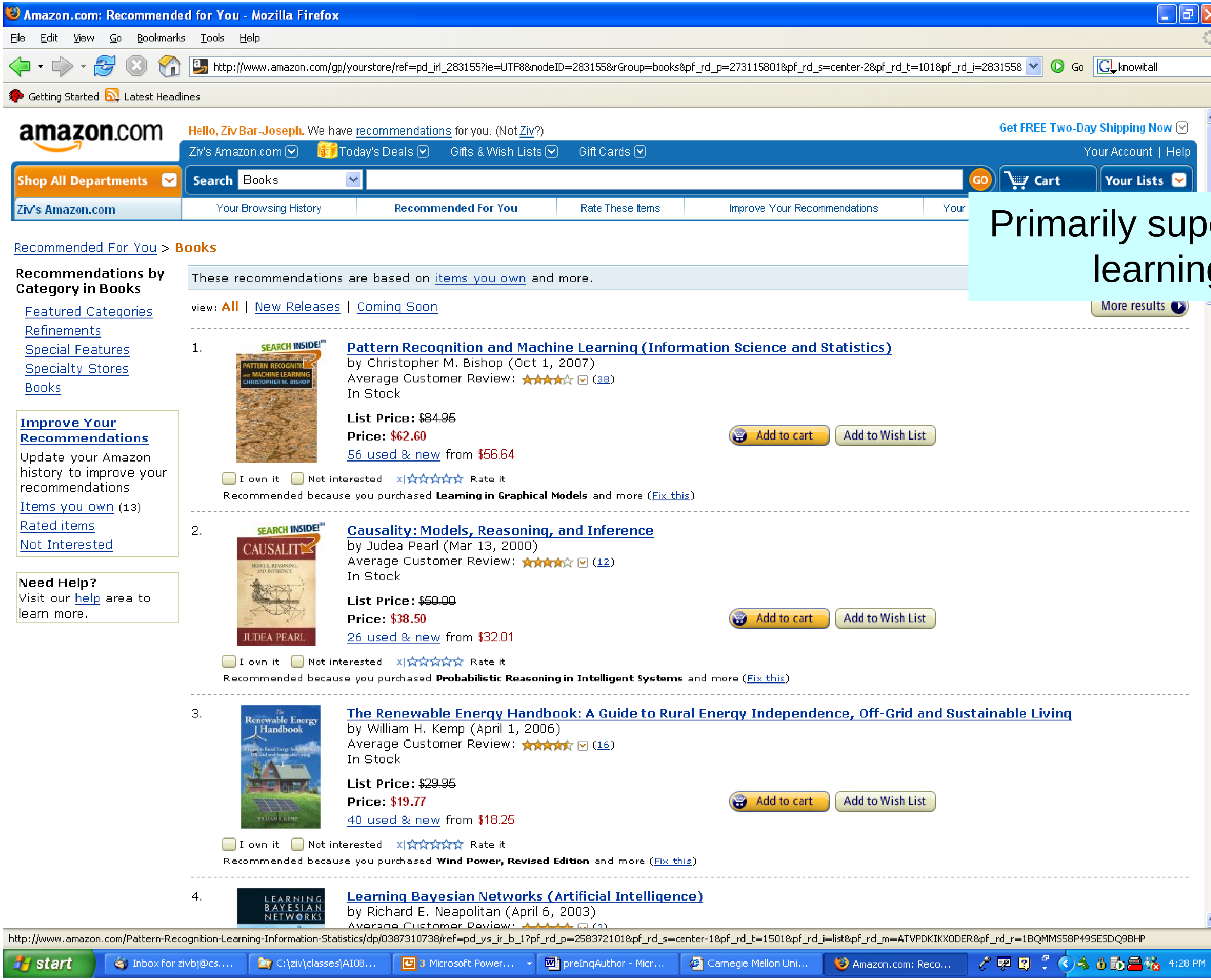
- Unsupervised learning
  Given $D = \{X_i\}$ group the data into Y classes using a model (or function) $F: X_i \rightarrow Y_j$

- Reinforcement learning (reasoning under uncertainty)
  Given D = {environment, actions, rewards} learn a policy and utility functions:

  policy: $F1: \{e,r\} \rightarrow a$
  utility: $F2: \{a,e\} \rightarrow R$

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

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.

Browse the Knowledge Base!

Semi supervised learning

At present, NELL has accumulated a knowledge base of 967,123 beliefs that it has read from various web pages. 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
robert_trent_jones_sr is an Australian person	473	27-dec-2011	100.0
quality_gift is a character trait	475	29-dec-2011	99.5
confectioners_sugar is a food	473	27-dec-2011	95.4
st__petersburg_times is a newspaper	472	26-dec-2011	100.0
scott_olynek is a Canadian person	473	27-dec-2011	94.1
perth is a city that lies on the river swan_river	472	26-dec-2011	99.2
florida_state_university is a sports team also known as state_university	472	26-dec-2011	100.0
press_enterprise is a newspaper in the city riverside	472	26-dec-2011	98.4

Driveless cars

Supervised and
reinforcement learning

Helicopter control

Reinforcement learning

Biology

ACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAACGCTGAGCAATTCGATAGCAATTC
GATAACGCTGAGCAATCGGATAACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAACG
CTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAATCGGATATCGATAGCAATTCGATAAATC
GGATAACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAACGCTGAGCAATTCGATAGC
AATTCGATAACGCTGAGCAATCGGATATCGATAGCAATTCGATAACGCTGAGCAACGCTGAGCA
ATTCGATAGCAATTCGATAACGCTGAGCAATCGGATAACGCTGAGCAATTCGATAGCATTCGAT
AACGCTGAGCAACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAATCGGATAACGCTG
AGCAATTCGATAGCAATTCGATAACGCTGAGCTGAGCAATTCGATAGCAATTCGATAACGCTGA
GCAATCGGATAACGCTGAGCA TTCGATAGCAATTCGATAACGCTGAGCAACGCTGAGCAATTC
GATAGCAAT GATAGCAATTCGATAACGCTGAGCAACGCTGAGCAATTCGAT
AGCAATTCGATAACGCTGAG CAATCGGATAACGCTGAGCAATTCGATAGCAATTCGATAACGCT
GAGCAACGCTGAGCAATTC GATAGCAATTCGATAACGCTGAGCAATCGGATATCGATAGCAATT
CGATAACGCTGAGCAACG TGAGCAATTCGATAGCAATTCGATAACGCTGAGCAATCGGATAAC
GC CGCTGAGCTGAGCAATTCGATAGCAATTCGATAACG

Which part is the gene?

CT CGATAGCAATTCGATAACGCTGAGCAACGCTGAGCA
AT AATCGGATATCGATAGCAATTCGATAACGCTGAGCA
ACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAATCGGATAACGCTGAGCAATTCGAT
AGCATTCGATAACGCTGAGCAACGCTGAGCAATTCGATAGCAATTCGATCGGATAACGCTGAGC
AATTCGATAGCAATTCGATAACGCTGAGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCA
ATCGGATAACGCTGAGCAATTCGATAGCA TGAGCAATTCGAT
AGCAATTCGATAACGCTGAGCAATCGGAT GAGCAACGCTGA
GCAATTCGATAGCAATTCGATAACGCTGA TTCGATAGCATTC
GATAACGCTGAGCAACGCTGAGCAATTCG CAATCGGATAACG
CTGAGCAATTCGATAGCAATTCGATAACG ATTCGATAACGC

Supervised and unsupervised learning (can also use active learning)

TGAGCAATCGGATAACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAACGCTGAGCAA
TTCGATAGCAATTCGATAGCAATTCGATAGCAATTCGATAACGCTGAGCAACGCTGAGCAATTC
GATAGCAATTCGATAACGCTGAGCAATCGGATAACGCTGAGCAATTCGATAGCAATTCGATAAC
GCTGAGCAACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAATCGGATATCGATAGCA
ATTCGATAACGCTGAGCAACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCAATCGGAT
AACGCTGAGCAATTCGATAGCAATTCGATAACGCTGAGCTGAGCAATTCGATAGCAATTCGATA
ACGCTGAGCAATCGGA

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 2012”: Discrete

    - “A = % change in Google stock in 2012”: Continuous

Axioms of probability (Kolmogorov's axioms)

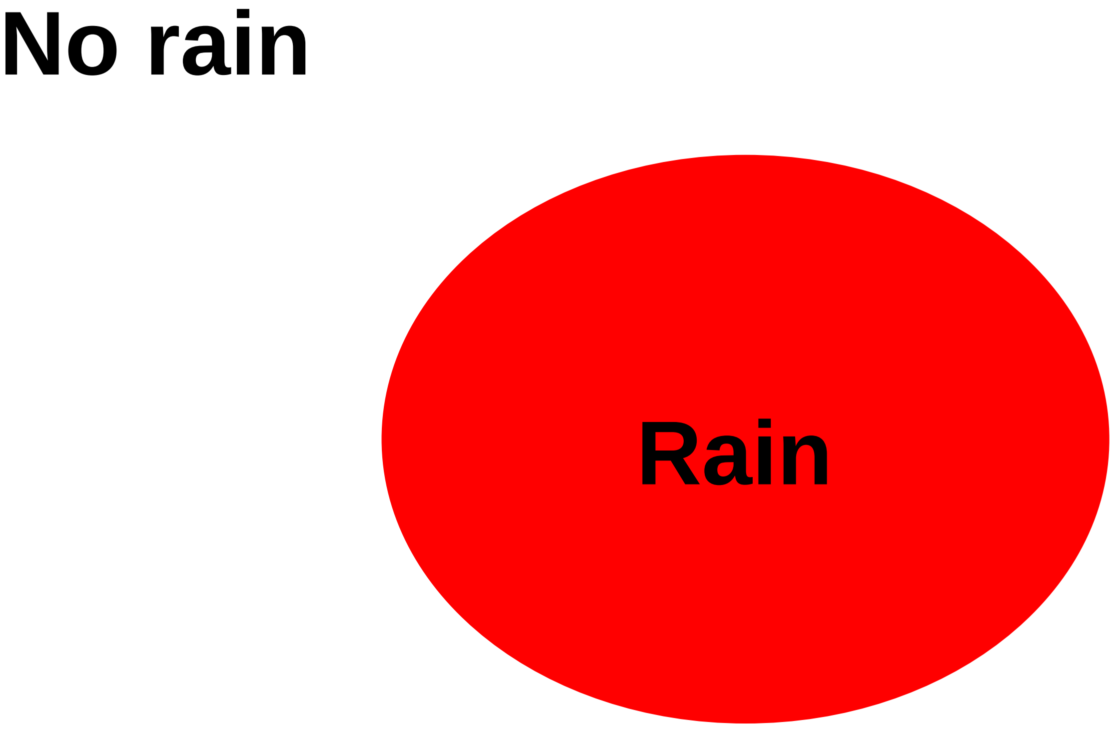
A variety of useful facts can be derived from just three axioms:

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



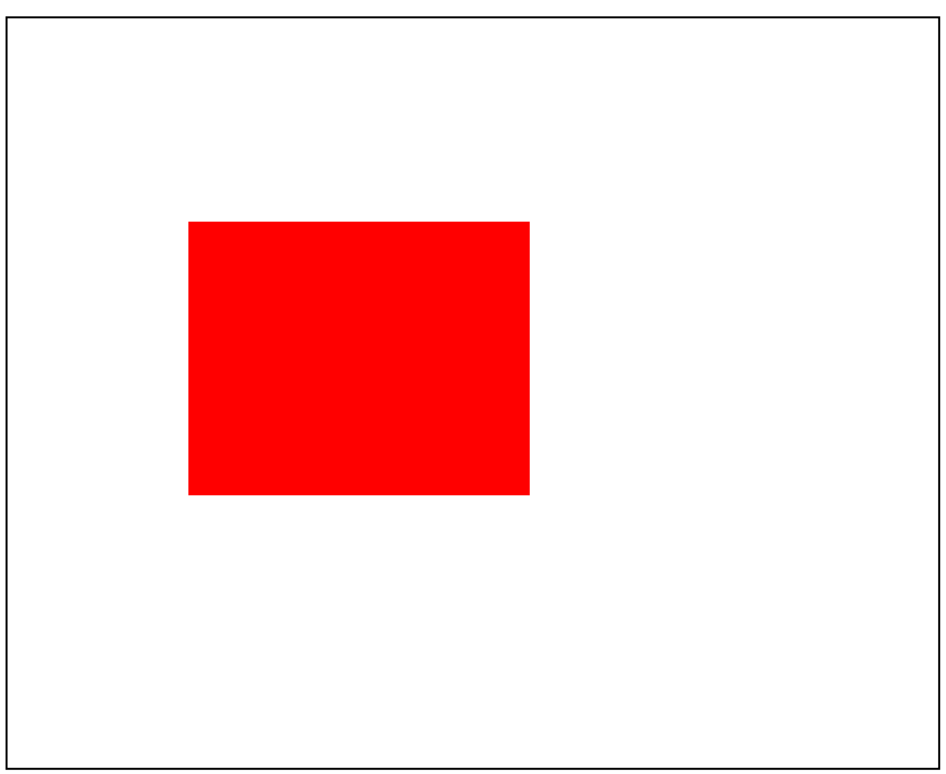
P(rain tomorrow) = 0.2

P(no rain tomorrow) = 0.8

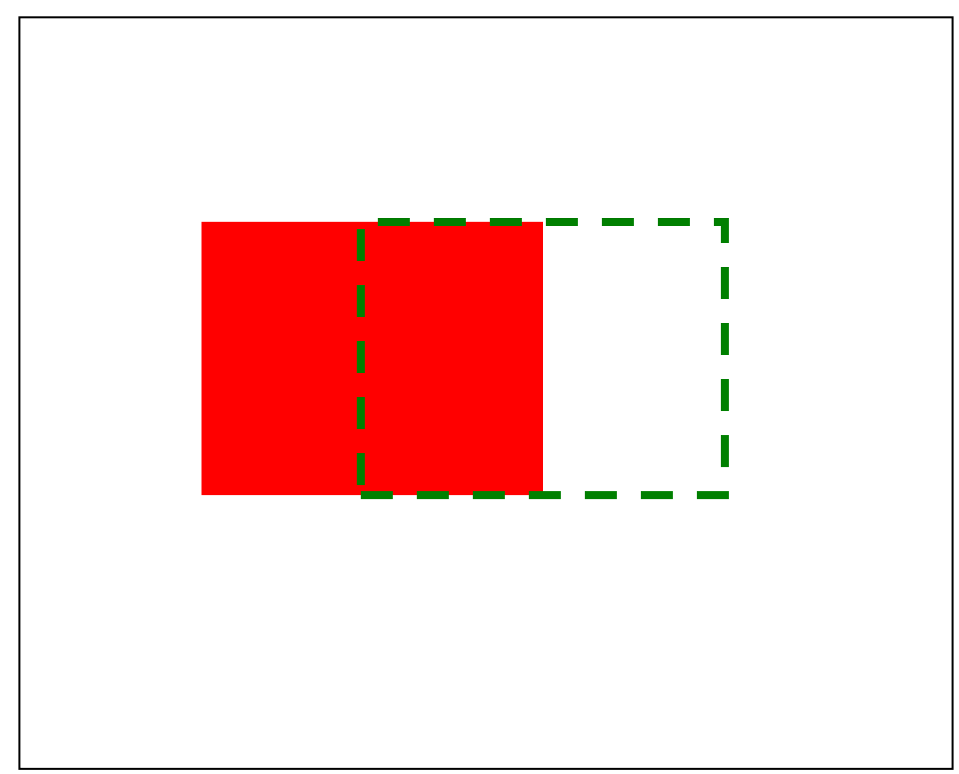
Conditional probability

- 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 maybe 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

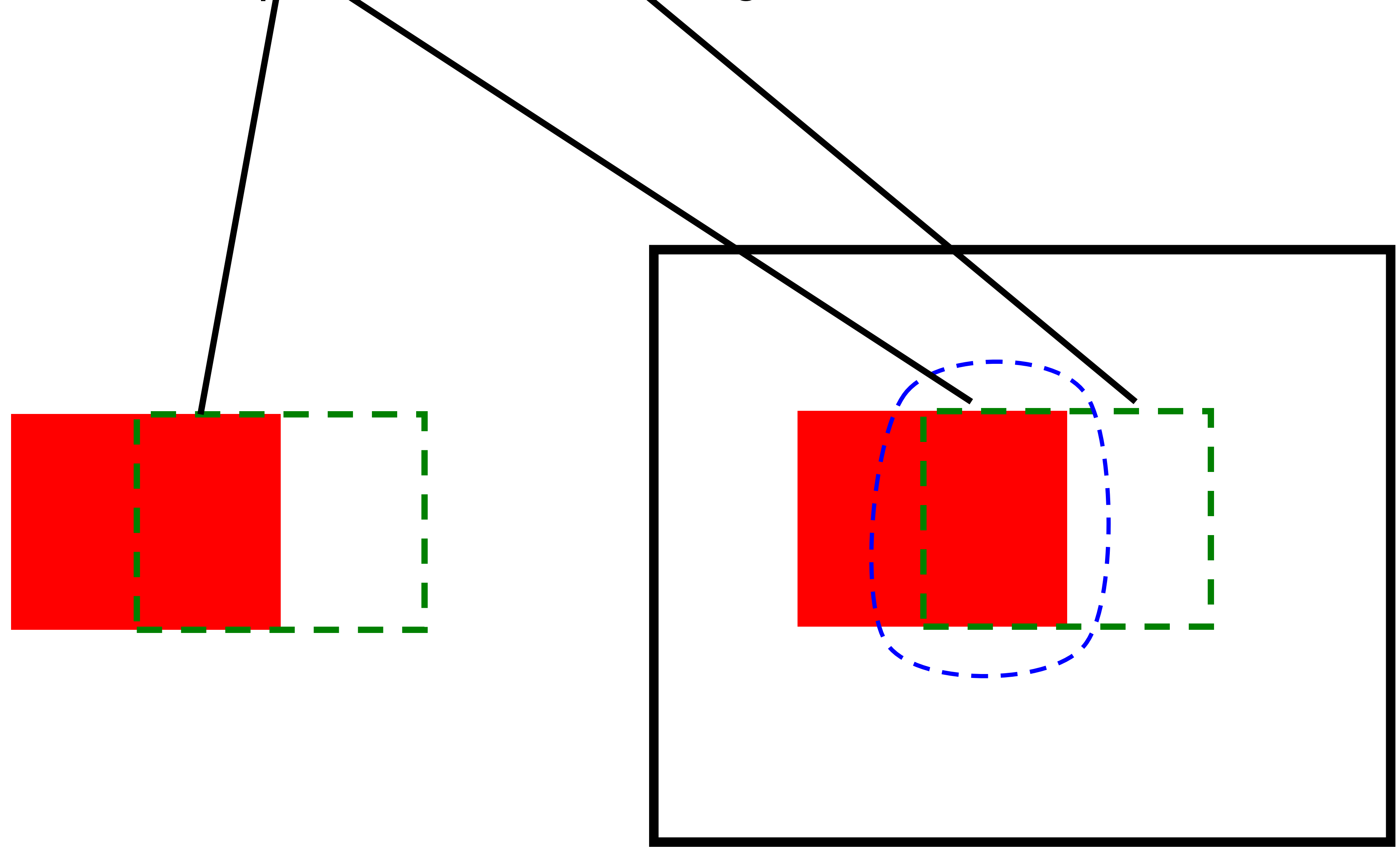
Size	Time	Eval
30	R	2
70	R	1
12	S	2
8	S	3
56	R	1
24	S	2
10	S	3
23	R	3
9	R	2
45	R	1

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.

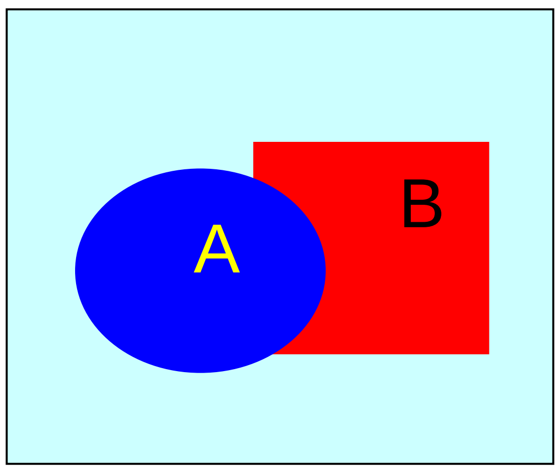
Bayes rule (cont)

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

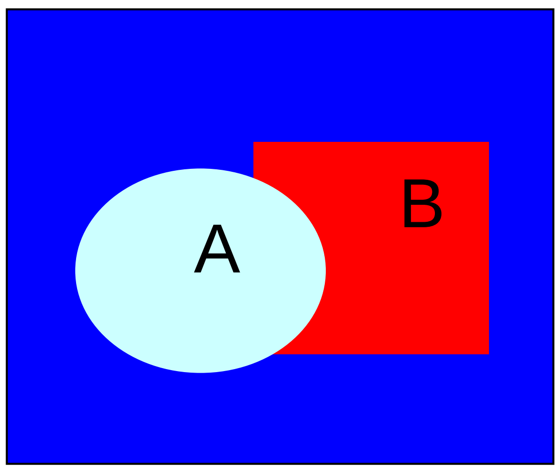
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(B|A)P(A)}{\sum_A P(B|A)P(A)}$$

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

P(B,A=1)



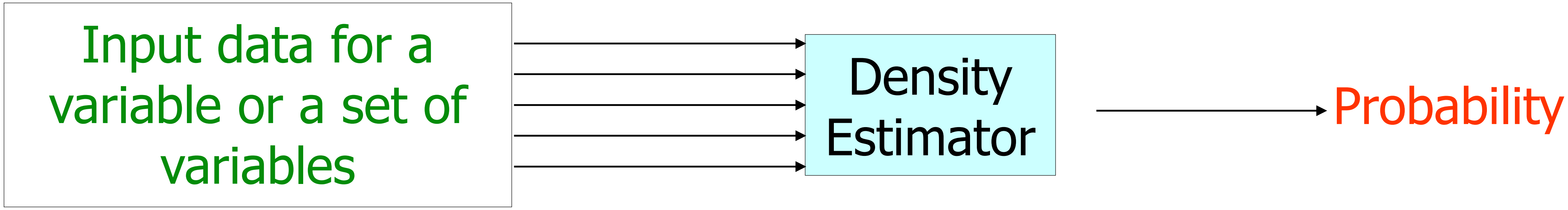
P(B,A=0)



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 can do many good things…
  - Can sort the records by probability, and thus spot weird records (anomaly detection)
  - Can do inference: P(E1|E2)

    Medical diagnosis / Robot sensors
  - Ingredient for Bayes networks and other types of ML methods

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:

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

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

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

- 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: $P(D|M) = q^{n_1}(1-q)^{n_2}$

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: $P(D|M) = q^{n_1}(1-q)^{n_2}$

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

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

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

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

$$q^{n_1-1}(1-q)^{n_2-1}(n_1(1-q) - qn_2) = 0 \Rightarrow$$

$$n_1(1-q) - qn_2 = 0 \Rightarrow$$

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

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

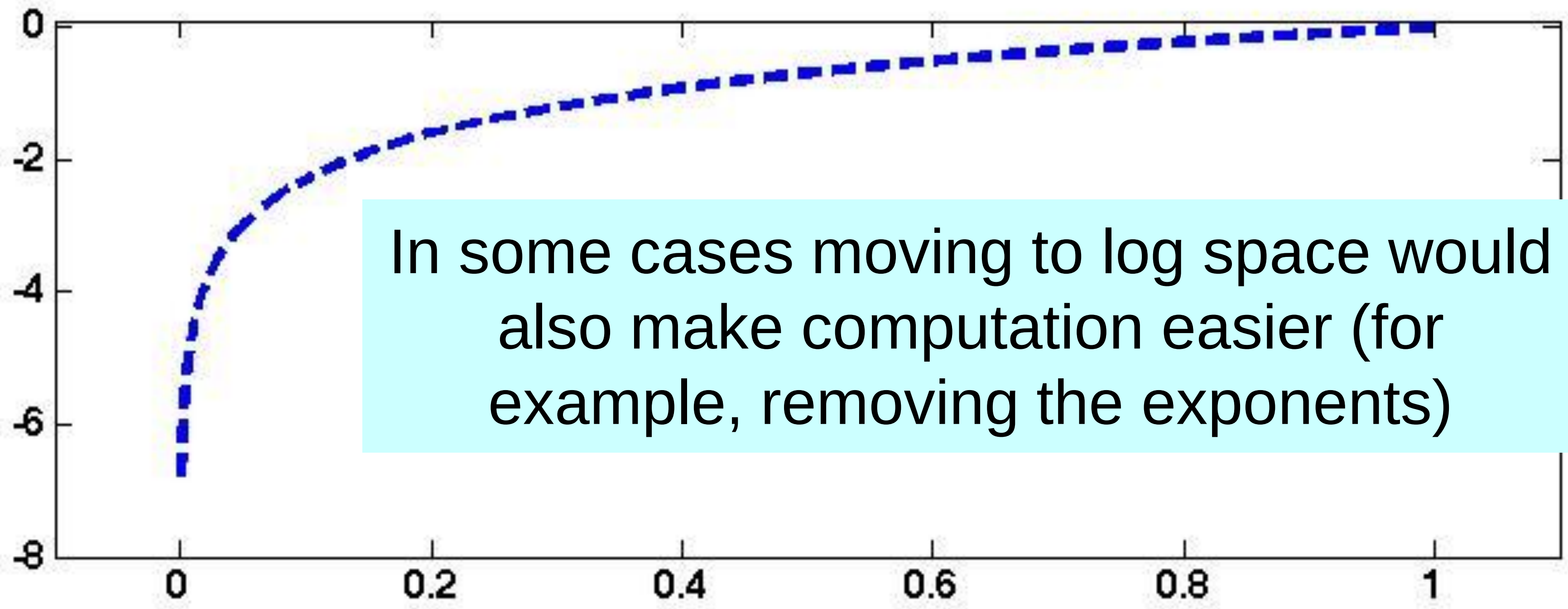
Log Probabilities

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)

Density estimation

- Binary and discrete variables:

- Continuous variables:

Easy: Just count!

Harder (but just a bit): Fit a model

But what if we only have very few samples?

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:**

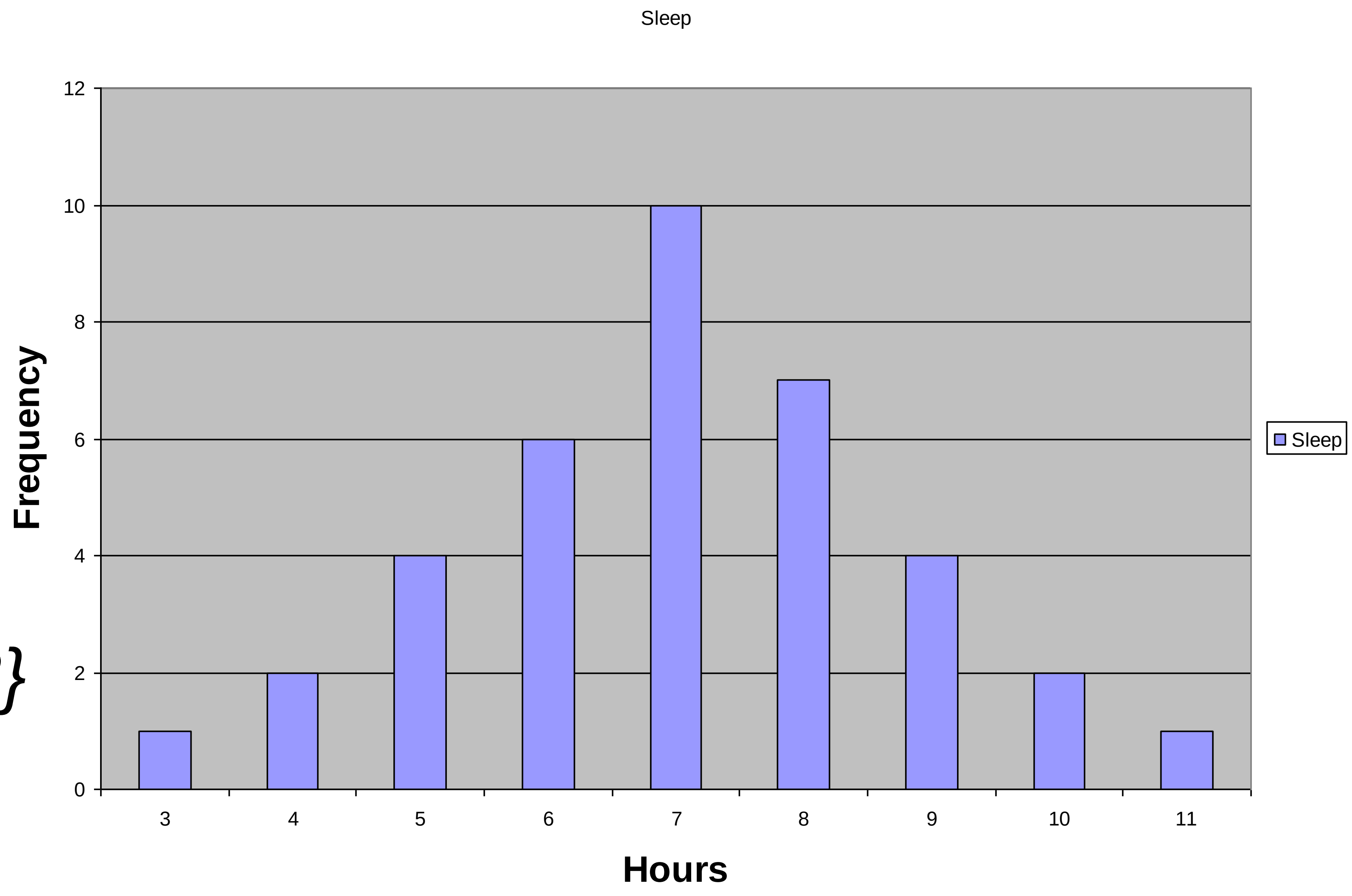
E{X}

7.03

- **Variance of X:**

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

3.05

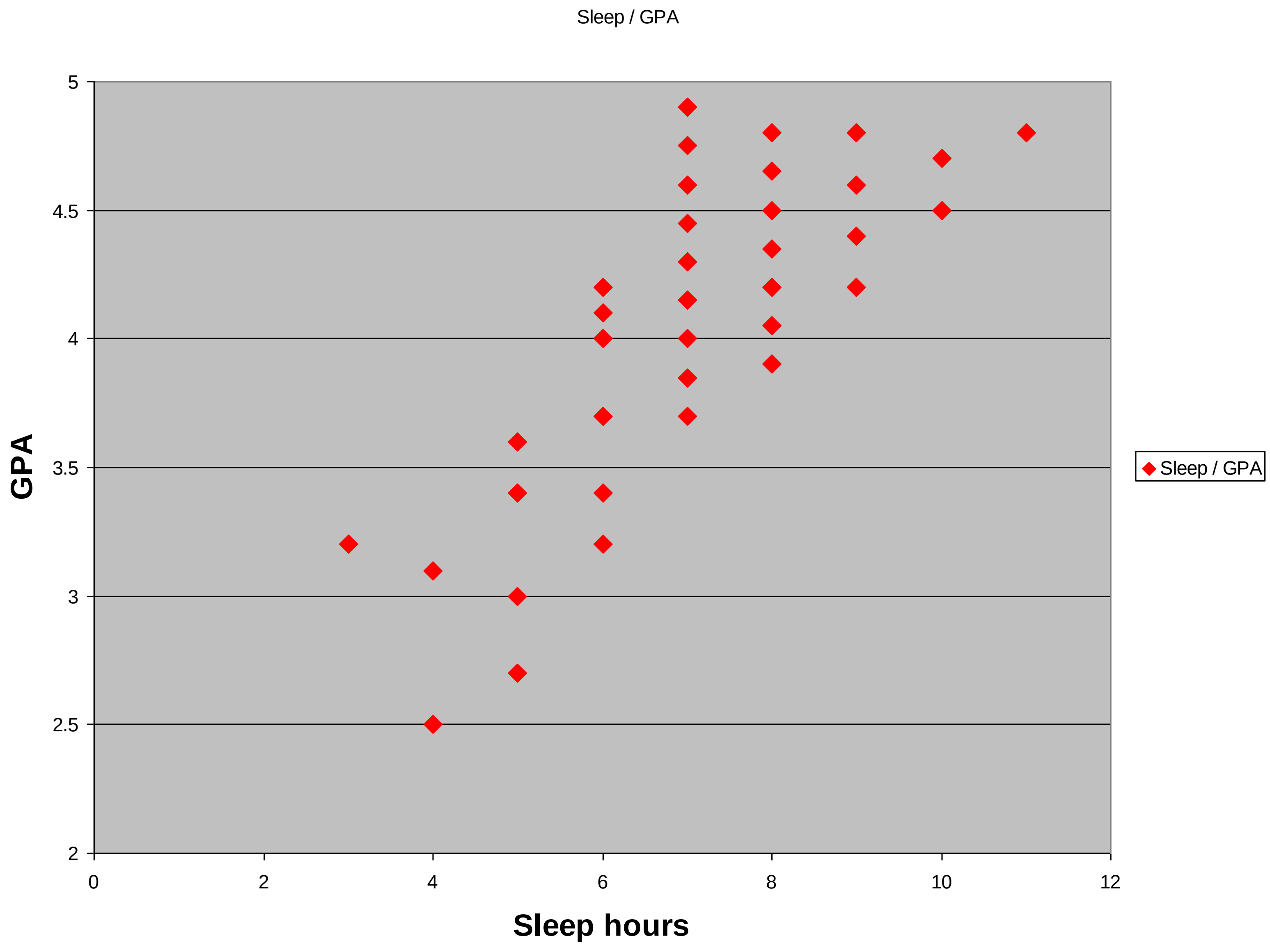


Covariance: Sleep vs. GPA

- **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)

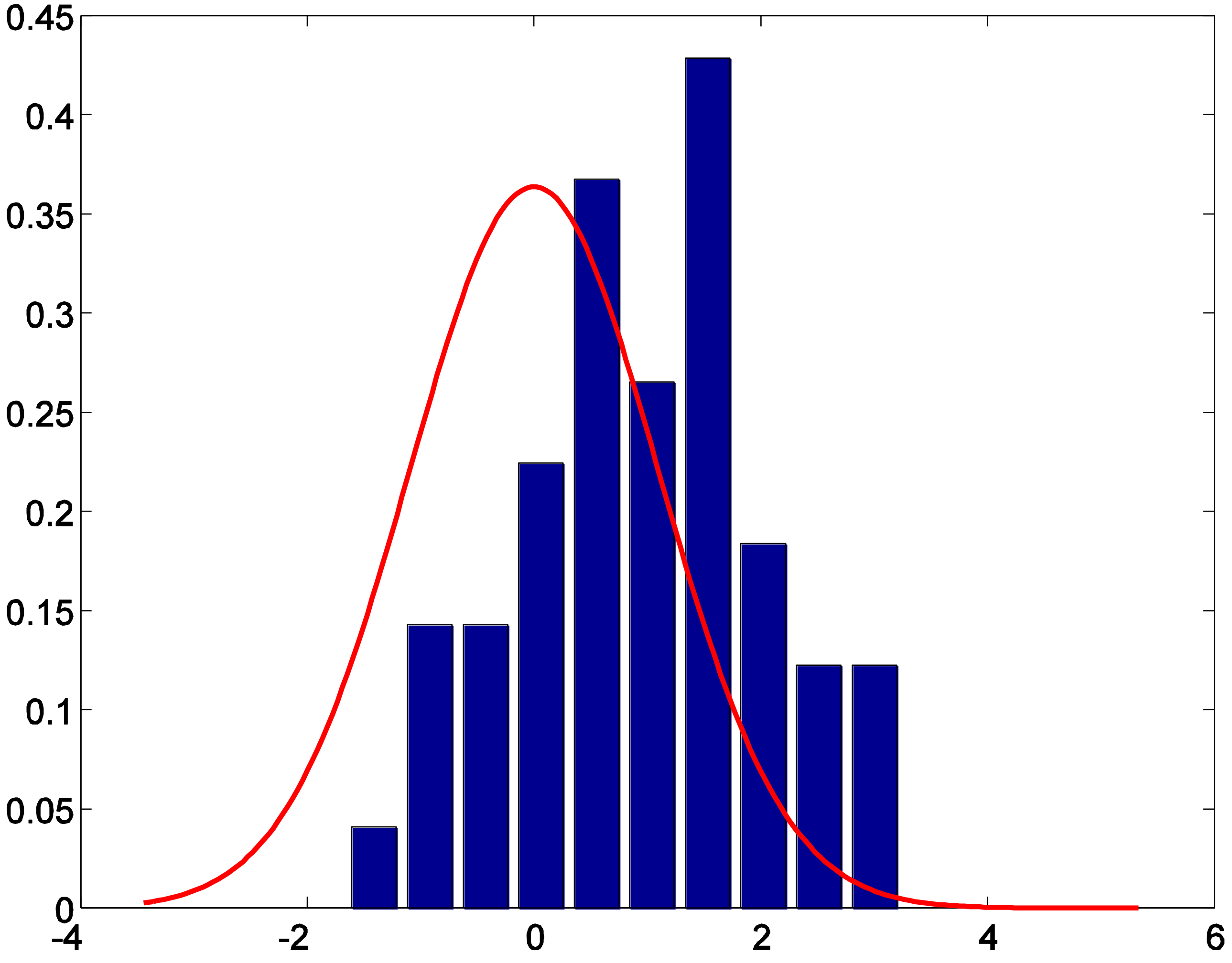
where

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

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

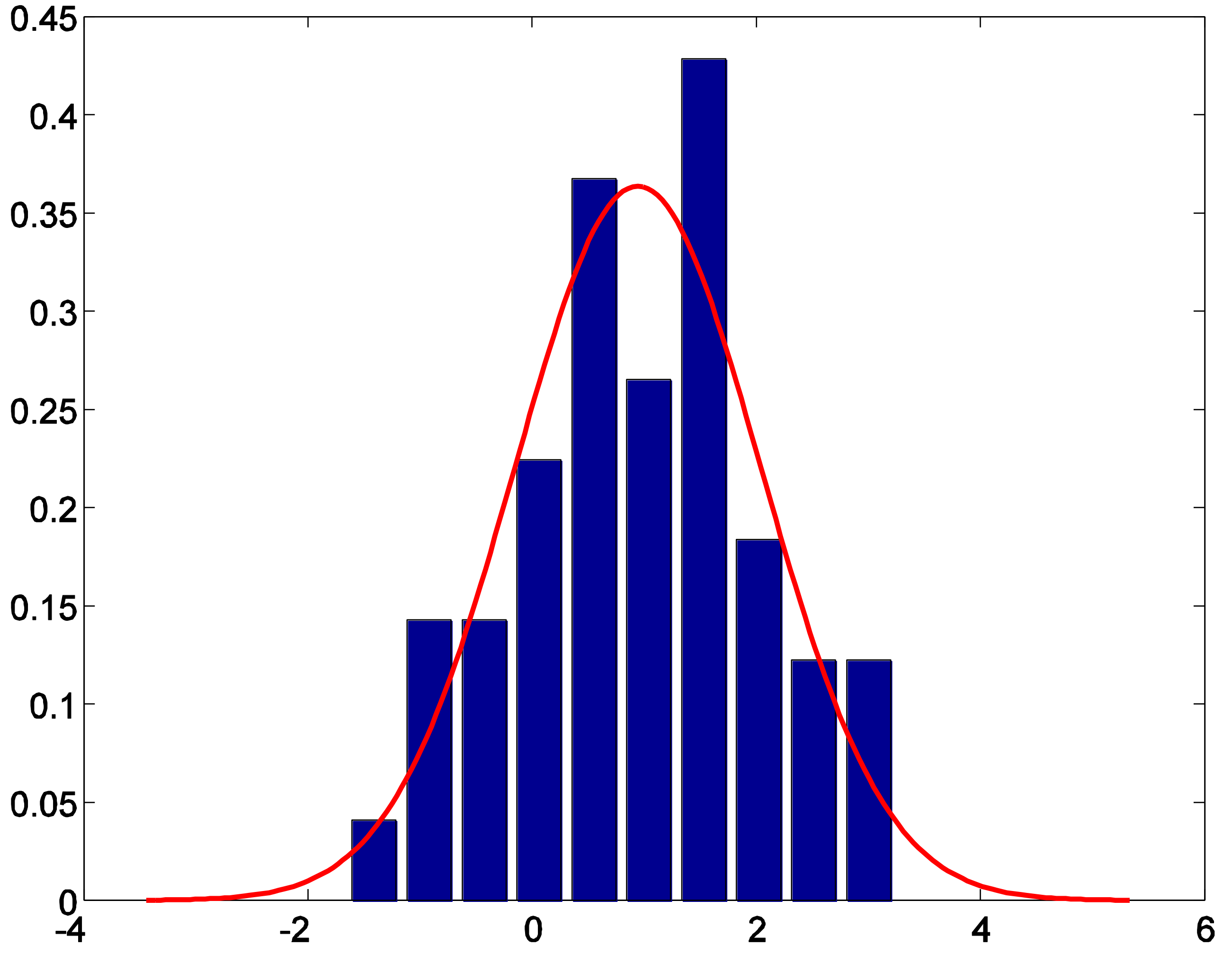
The Parameters of Our Model

- 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



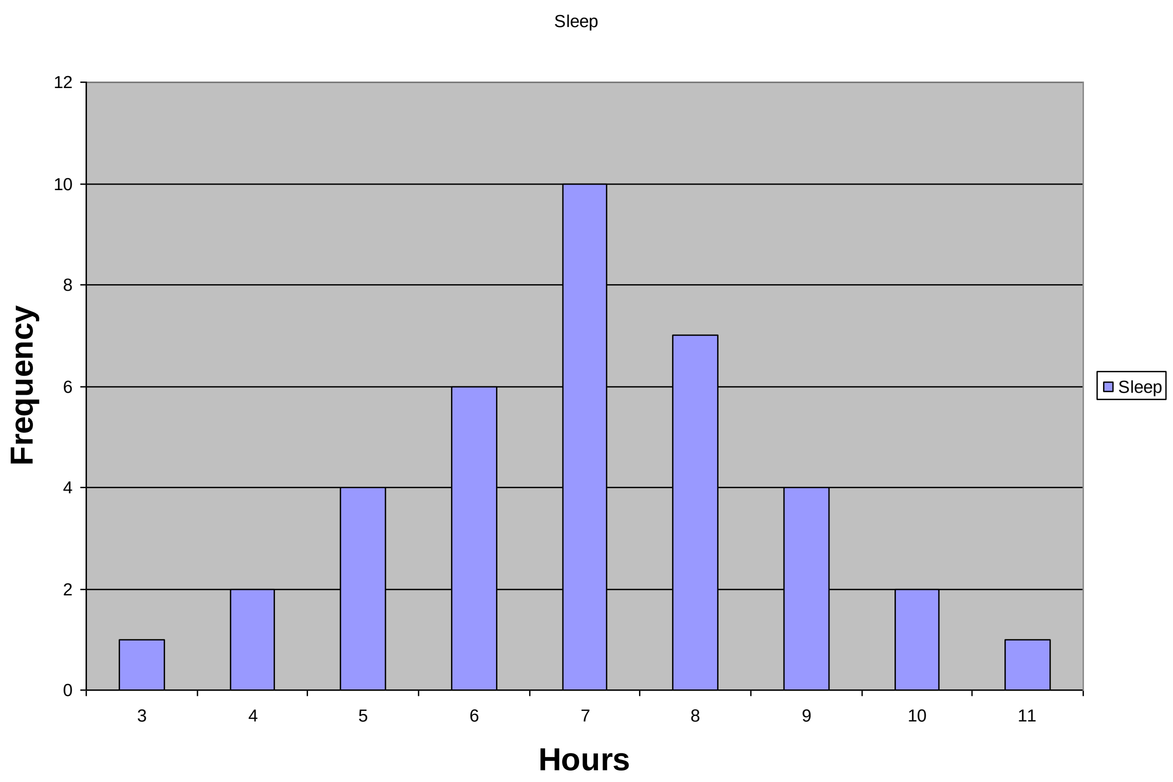
The Parameters of Our Model

- A statistical model is a **collection** of distributions; the **parameters** specify individual distributions x ~ N(μ,σ^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:

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

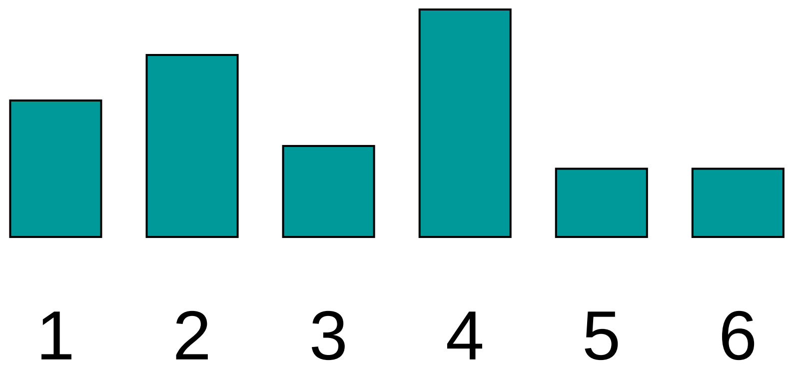
Why?

Important points

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

Probability Density Function

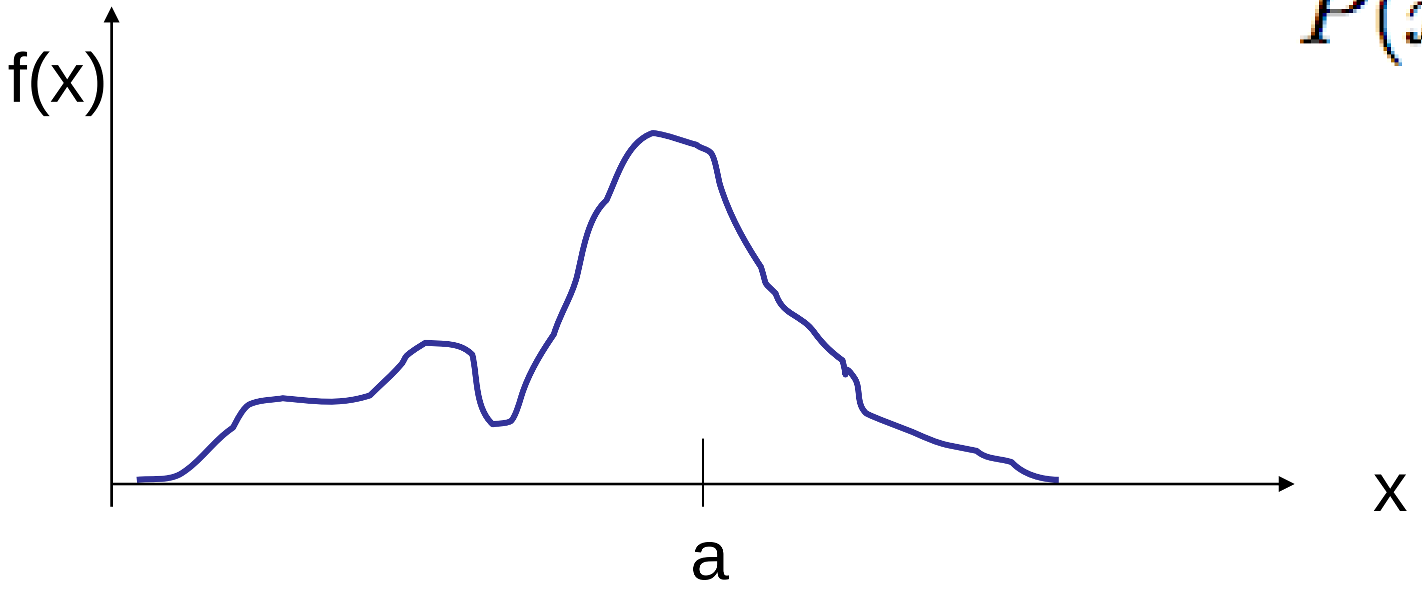
- Discrete distributions



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

- 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)

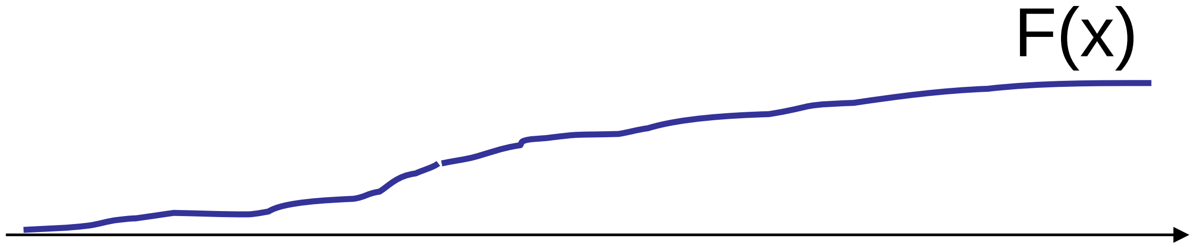
$$\frac{d}{dx}F(x) = f(x)$$

- 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:

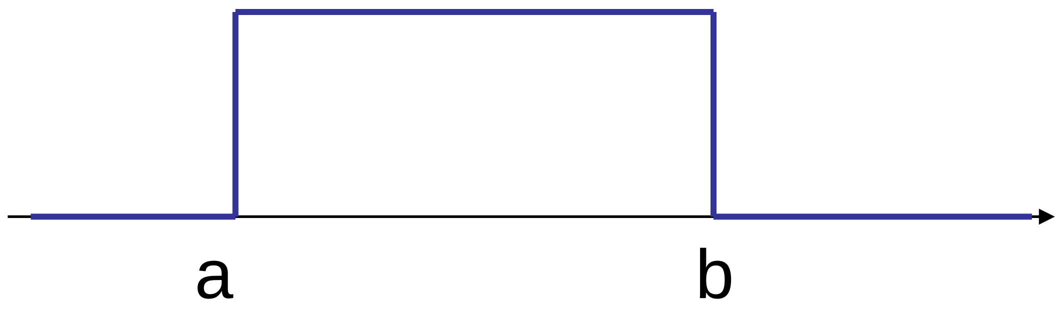
$$x \sim U(a,b)$$

- Mean/Var

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

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

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



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:

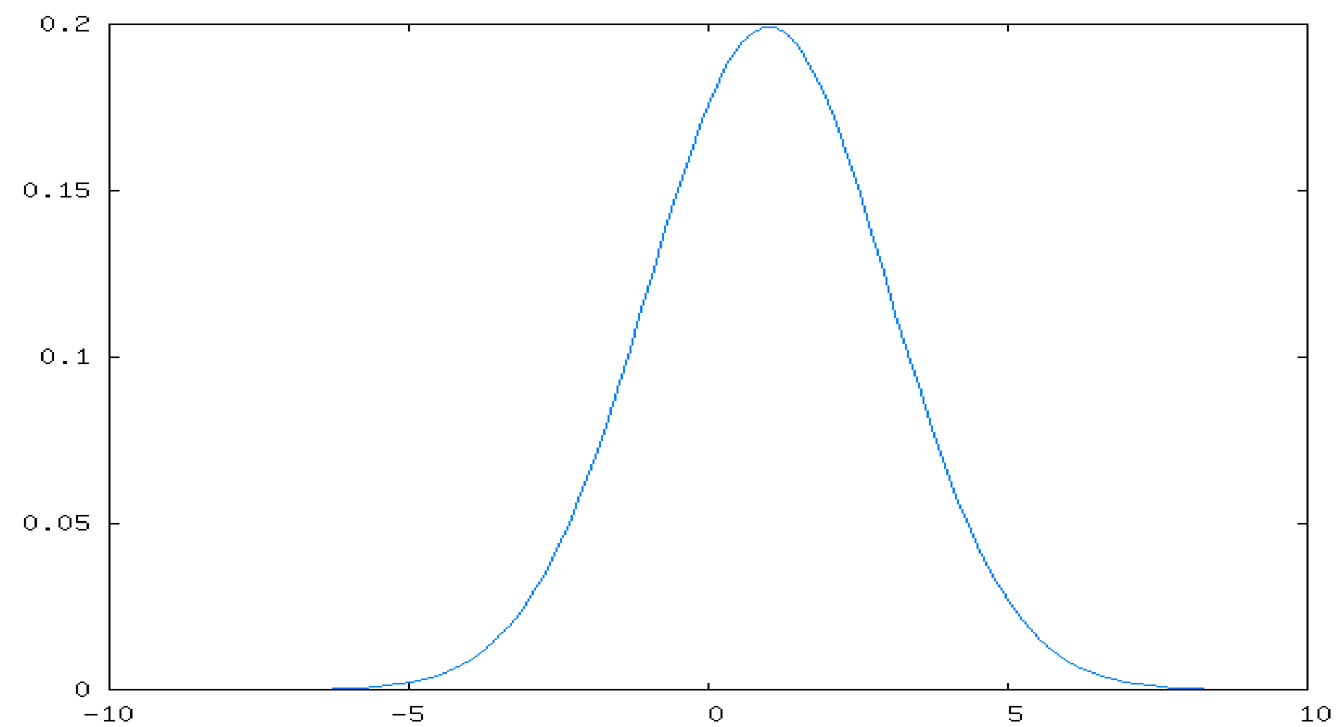
$$x \sim N(\mu, \sigma^2)$$

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

- 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

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

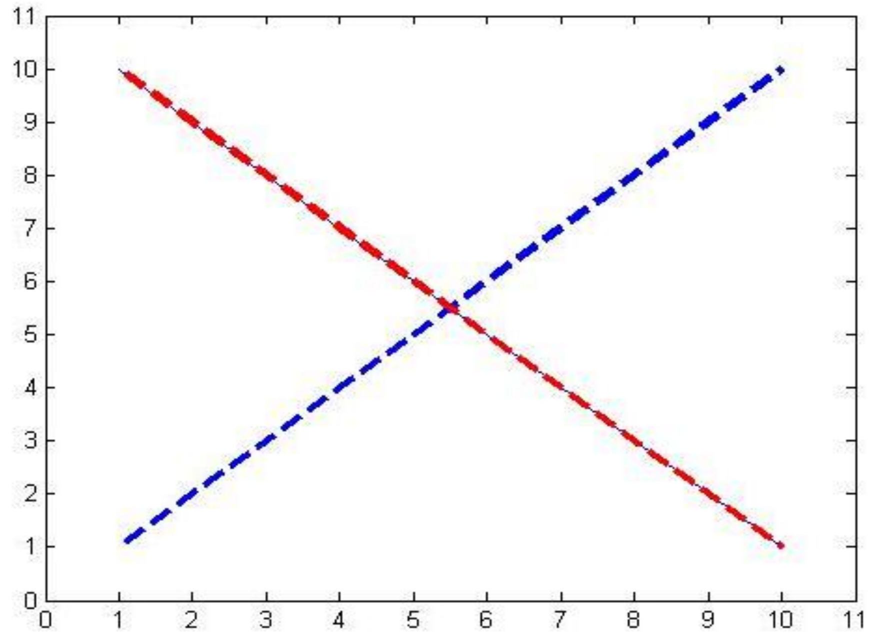
$$E[x] = \mu = (E[x_1], \ldots, E[x_N])^T$$

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

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

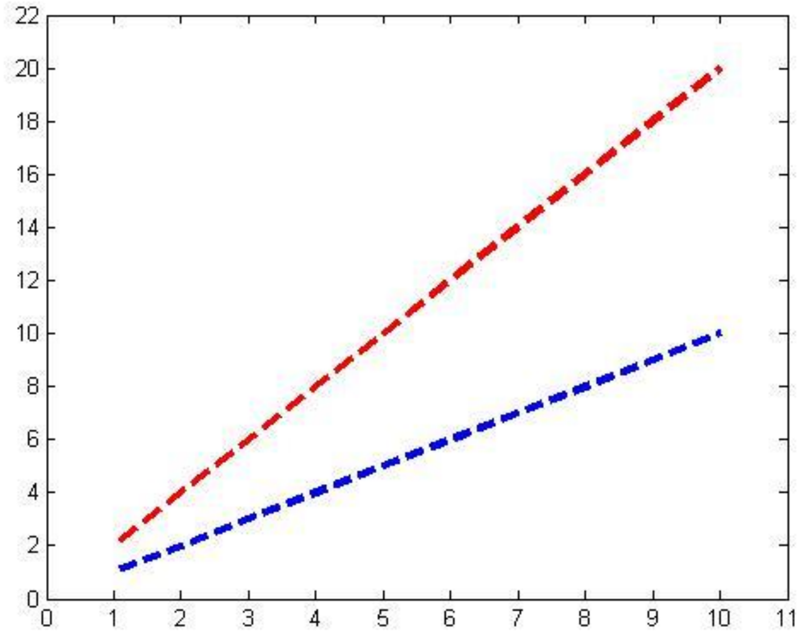
Covariance examples

Anti-correlated



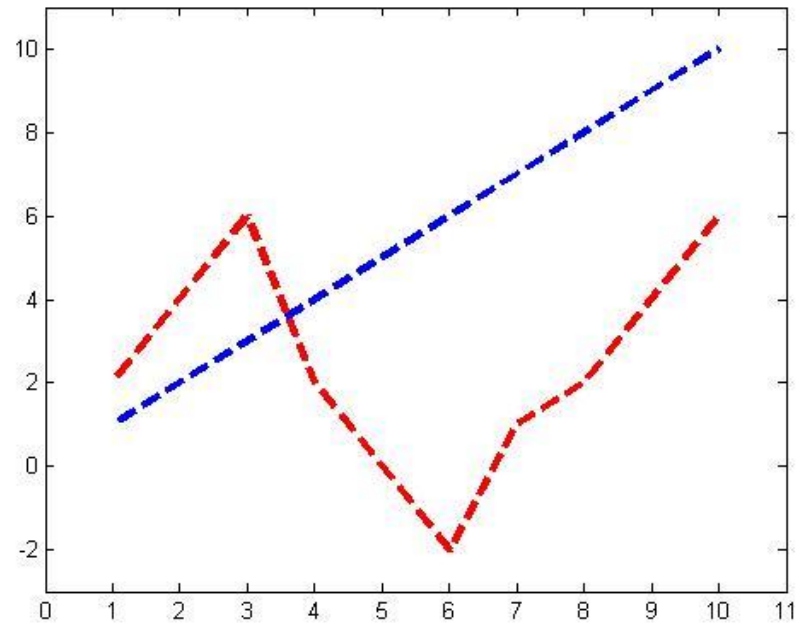
Covariance: -9.2

Correlated



Covariance: 18.33

Independent (almost)



Covariance: 0.6

Sum of Gaussians

- The sum of two Gaussians is a Gaussian:

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

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

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