Introduction to Deep Learning

2. Probability and Statistics

STAT 157, Spring 2019, UC Berkeley

Mu Li and Alex Smola

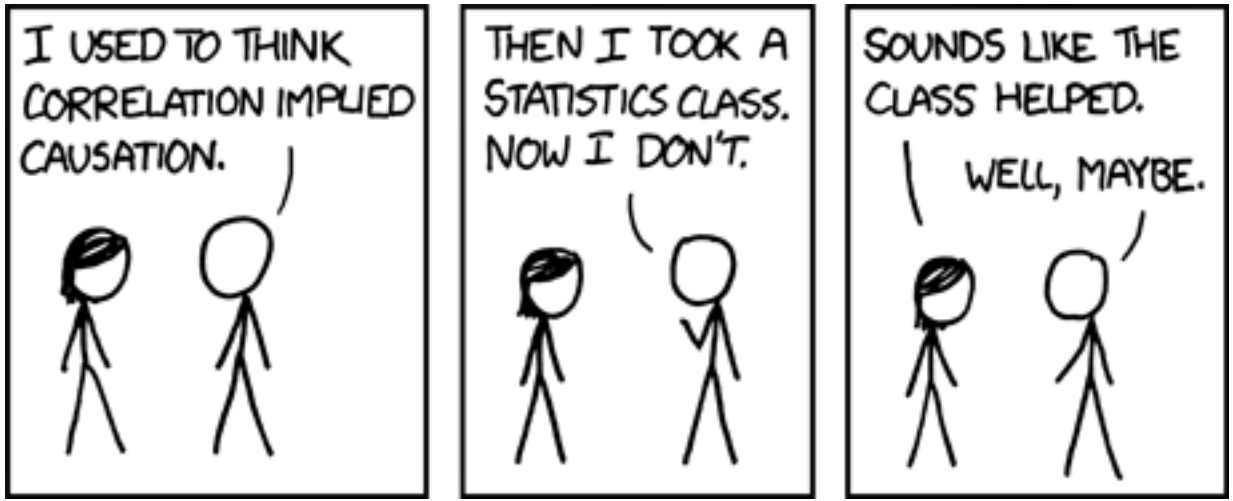
courses.d2l.ai/berkeley-stat-157

aws

Outline

- **Basic Probability**
  Random variables, conditional probabilities, Bayes rule
- **Naive Bayes**
  - Multiple tests
  - Examples - OCR
- **Sampling**
  - Distributions (categorial, normal, uniform)
  - Central limit theorem

Basic Probability



xkcd.com

aws

Probability

Space of events X

- server working; slow response; server broken
- income of the user (e.g. $95,000)
- query text for search (e.g. “statistics tutorial”)

Probability axioms (Kolmogorov)

$$\Pr(X) \in [0, 1], \Pr(\mathcal{X}) = 1$$
$$\Pr(\cup_i X_i) = \sum_i \Pr(X_i) \text{ if } X_i \cap X_j = \emptyset$$

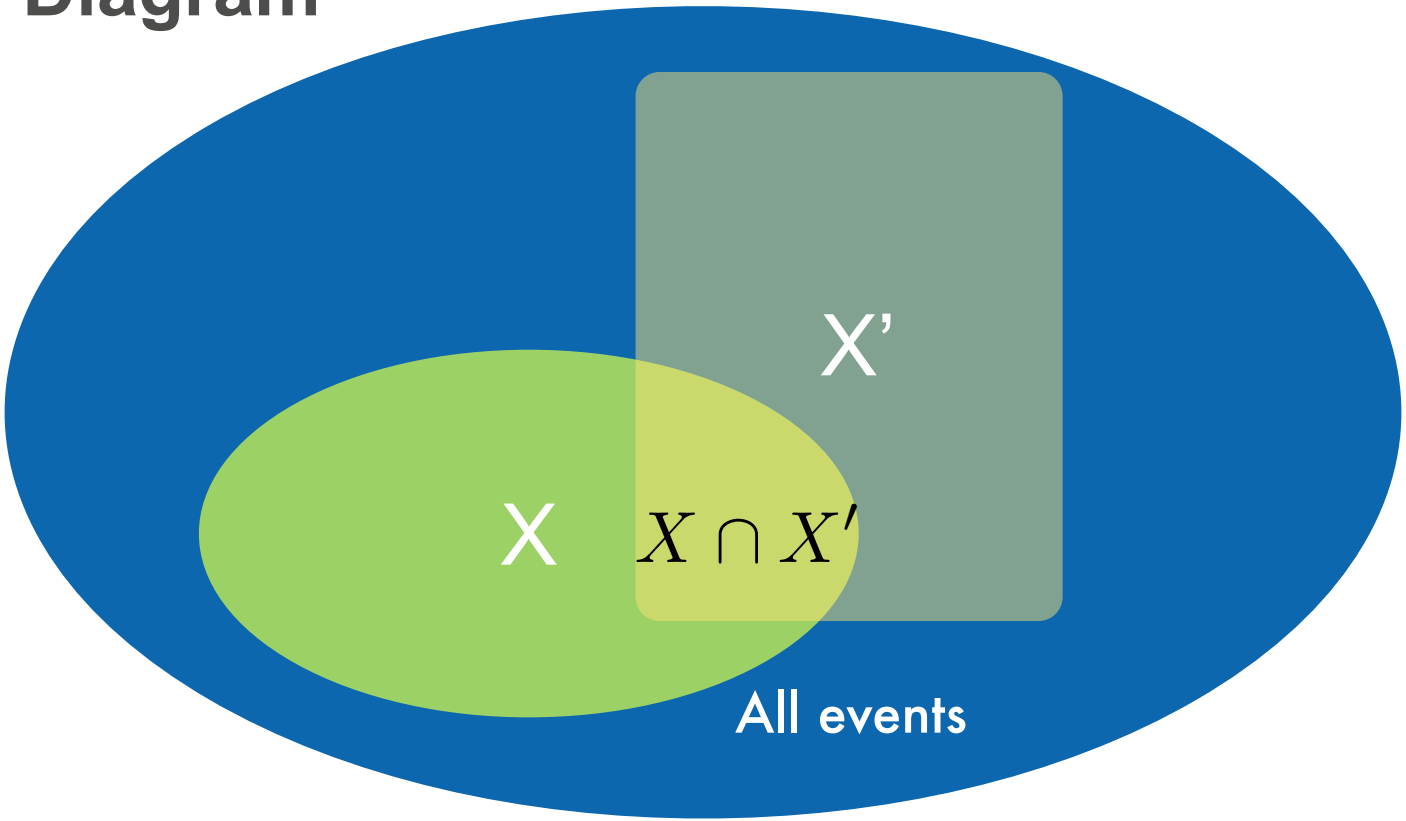
Example queries

- P(server working) = 0.999
- P(90,000 < income < 100,000) = 0.1

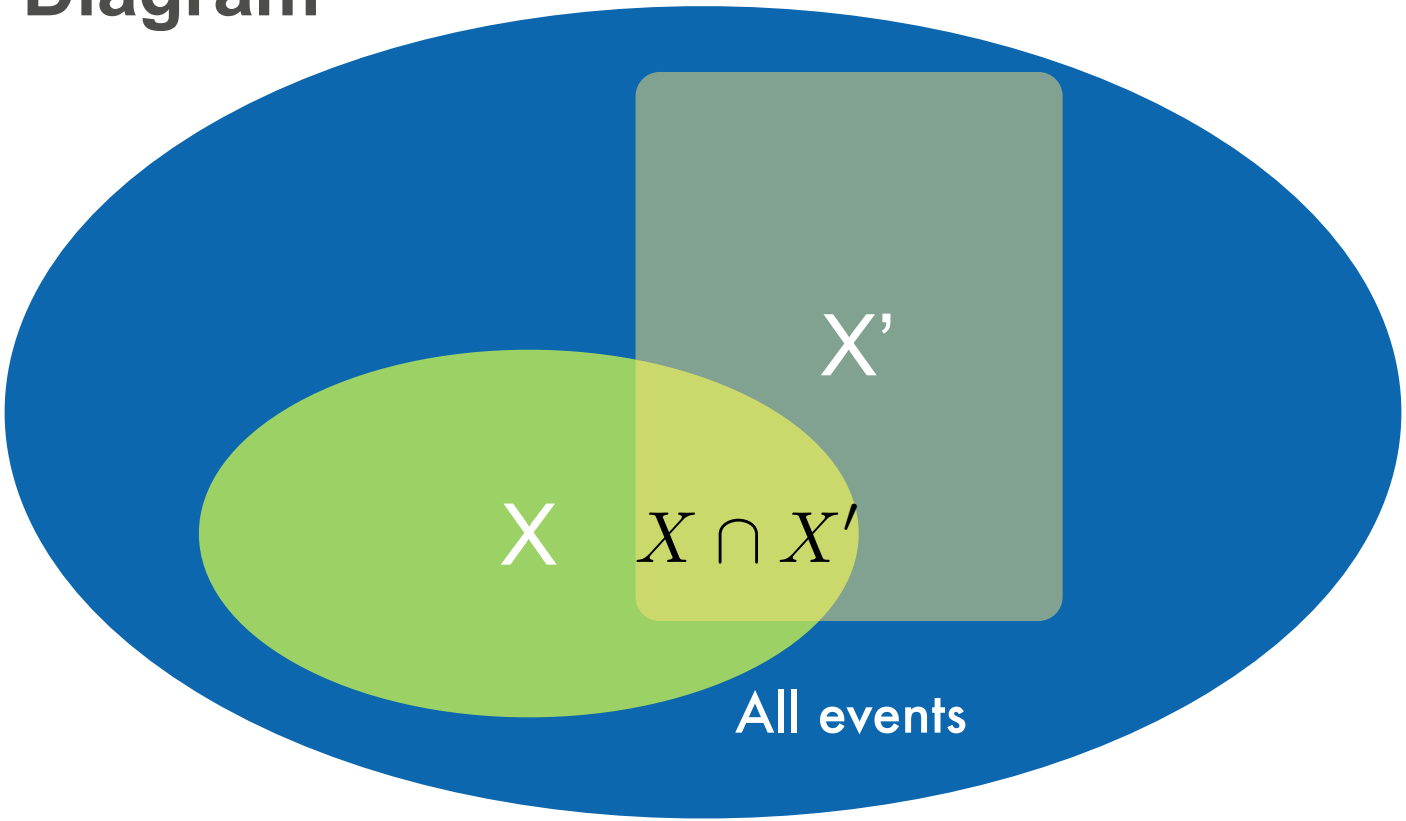
discrete

continuous

Venn Diagram



Venn Diagram



$$\Pr(X \cup X') = \Pr(X) + \Pr(X') - \Pr(X \cap X')$$

(In)dependence

Independence

- Login behavior of two users (approximately)
- Disk crash in different colos (approximately)

(In)dependence

Independence

- Login behavior of two users (approximately)
- Disk crash in different colos (approximately)

$$\Pr(x, y) = \Pr(x) \cdot \Pr(y)$$

(In)dependence

Independence

- Login behavior of two users (approximately)
- Disk crash in different colos (approximately)

$$\Pr(x, y) = \Pr(x) \cdot \Pr(y)$$

Dependent events

- Emails
- Queries
- News stream / Buzz / Tweets
- IM communication
- Russian Roulette

(In)dependence

Independence

- Login behavior of two users (approximately)
- Disk crash in different colos (approximately)

$$\Pr(x, y) = \Pr(x) \cdot \Pr(y)$$

Dependent events

- Emails
- Queries
- News stream / Buzz / Tweets
- IM communication
- Russian Roulette

$$\Pr(x, y) \neq \Pr(x) \cdot \Pr(y)$$

(In)dependence

Independence

- Login behavior of two users (approximately)
- Disk crash in different colos (approximately)

$$\Pr(x, y) = \Pr(x) \cdot \Pr(y)$$

Dependent events

- Emails
- Queries
- News stream / Buzz / Tweets
- IM communication
- Russian Roulette

Everywhere

$$\Pr(x, y) \neq \Pr(x) \cdot \Pr(y)$$

Recognizing Cats and Dogs is easy …

Recognizing Cats and Dogs is easy …

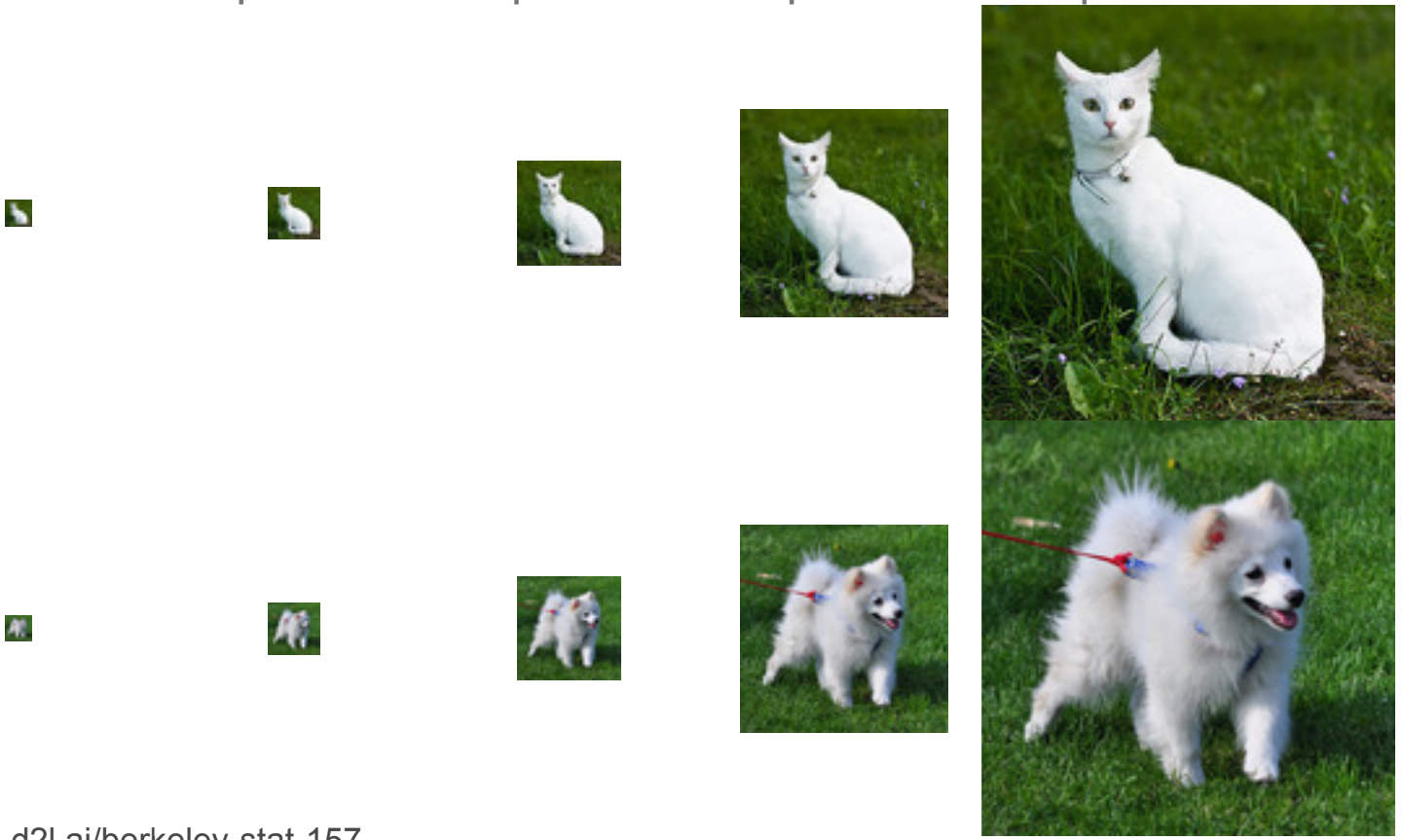
Recognizing Cats and Dogs is easy …

P(cat) = 0.4

P(cat) = 0.3

Recognizing Cats and Dogs is easy …

10px 20px 40px 80px 160px



What happened?

Uncertainty and Conditioning

- **Uncertainty**
  - Coin flip (head, tail, edge)
  - Lottery
- **Conditioning**
  - More information makes things more certain (if the information is related). p(y|x) rather than p(y)
  - We can build classifiers, regressors etc. (the point of this course)

Bayes Rule

- Joint Probability

$$\Pr(X, Y) = \Pr(X|Y)\Pr(Y) = \Pr(Y|X)\Pr(X)$$

- Bayes Rule

$$\Pr(X|Y) = \frac{\Pr(Y|X) \cdot \Pr(X)}{\Pr(Y)}$$

- Hypothesis testing
- Reverse conditioning

AIDS test (Bayes rule)

- Data
  - Approximately 0.1% are infected
  - Test detects all infections
  - Test reports positive for 1% healthy people
- Probability of having AIDS if test is positive

AIDS test (Bayes rule)

- Data
  - Approximately 0.1% are infected
  - Test detects all infections
  - Test reports positive for 1% healthy people
- Probability of having AIDS if test is positive

$$\begin{aligned}\Pr(a=1|t) &= \frac{\Pr(t|a=1)\cdot\Pr(a=1)}{\Pr(t)} \\ &= \frac{\Pr(t|a=1)\cdot\Pr(a=1)}{\Pr(t|a=1)\cdot\Pr(a=1)+\Pr(t|a=0)\cdot\Pr(a=0)} \\ &= \frac{1\cdot 0.001}{1\cdot 0.001+0.01\cdot 0.999} = 0.091\end{aligned}$$

Improving the diagnosis

Improving the diagnosis

- Use a follow-up test
  - Test 2 reports positive for 90% infections
  - Test 2 reports positive for 5% healthy people

$$\frac{0.01 \cdot 0.05 \cdot 0.999}{1 \cdot 0.9 \cdot 0.001 + 0.01 \cdot 0.05 \cdot 0.999} = 0.357$$

Improving the diagnosis

- Use a follow-up test
  - Test 2 reports positive for 90% infections
  - Test 2 reports positive for 5% healthy people

$$\frac{0.01 \cdot 0.05 \cdot 0.999}{1 \cdot 0.9 \cdot 0.001 + 0.01 \cdot 0.05 \cdot 0.999} = 0.357$$

- Why can't we use Test 1 twice?
  Outcomes are **not** independent but tests 1 and 2 are **conditionally independent**

Improving the diagnosis

- Use a follow-up test
  - Test 2 reports positive for 90% infections
  - Test 2 reports positive for 5% healthy people

$$\frac{0.01 \cdot 0.05 \cdot 0.999}{1 \cdot 0.9 \cdot 0.001 + 0.01 \cdot 0.05 \cdot 0.999} = 0.357$$

- Why can't we use Test 1 twice?
  Outcomes are **not** independent but tests 1 and 2 are **conditionally independent**

$$p(t_1, t_2|a) = p(t_1|a) \cdot p(t_2|a)$$

Naive Bayes

aws

Naive Bayes Spam Filter

Naive Bayes Spam Filter

- **Key assumption**
  Words occur independently of each other given the label of the document

$$p(w_1, \ldots, w_n|\text{spam}) = \prod_{i=1}^{n} p(w_i|\text{spam})$$

Naive Bayes Spam Filter

- **Key assumption**
  Words occur independently of each other given the label of the document

$$p(w_1, \ldots, w_n|\text{spam}) = \prod_{i=1}^{n} p(w_i|\text{spam})$$

- Spam classification via Bayes Rule

$$p(\text{spam}|w_1, \ldots, w_n) \propto p(\text{spam}) \prod_{i=1}^{n} p(w_i|\text{spam})$$

Naive Bayes Spam Filter

- **Key assumption**
  Words occur independently of each other given the label of the document

$$p(w_1, \ldots, w_n|\text{spam}) = \prod_{i=1}^{n} p(w_i|\text{spam})$$

- Spam classification via Bayes Rule

$$p(\text{spam}|w_1, \ldots, w_n) \propto p(\text{spam}) \prod_{i=1}^{n} p(w_i|\text{spam})$$

- **Parameter estimation**
  Compute spam probability and word distributions for spam and ham

Naive Bayes Spam Filter

Equally likely phrases

Get rich quick. Buy UCB stock.

Buy Viagra. Make your UCB experience last longer.

You deserve a PhD from UCB.
We recognize your expertise.

Naive Bayes Spam Filter

Equally likely phrases

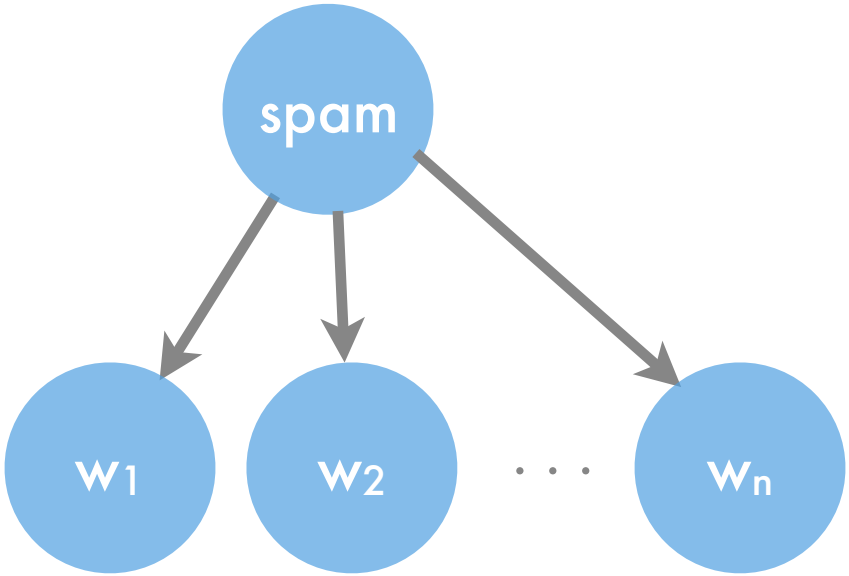
Get rich quick. Buy UCB stock.

Buy Viagra. Make your UCB experience last longer.

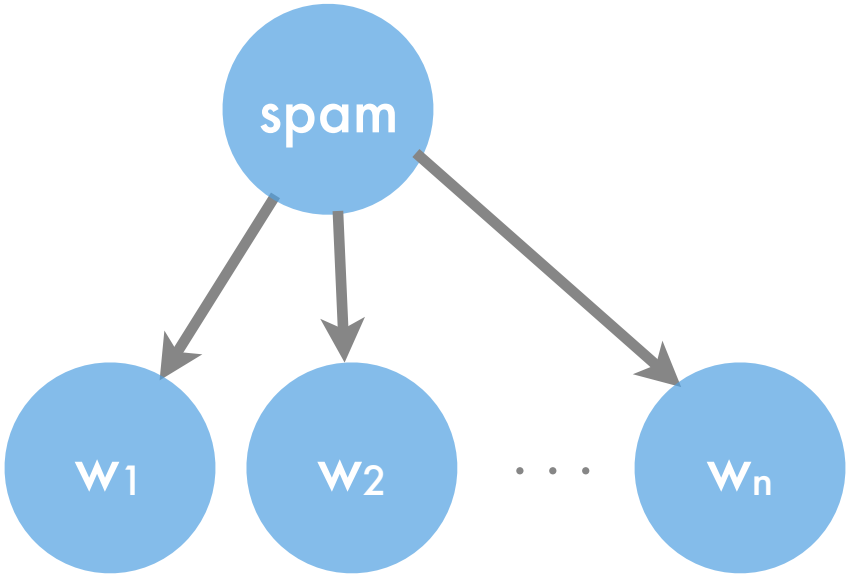
You deserve a PhD from UCB.
We recognize your expertise.

Make your rich UCB PhD experience last longer.

A Graphical Model

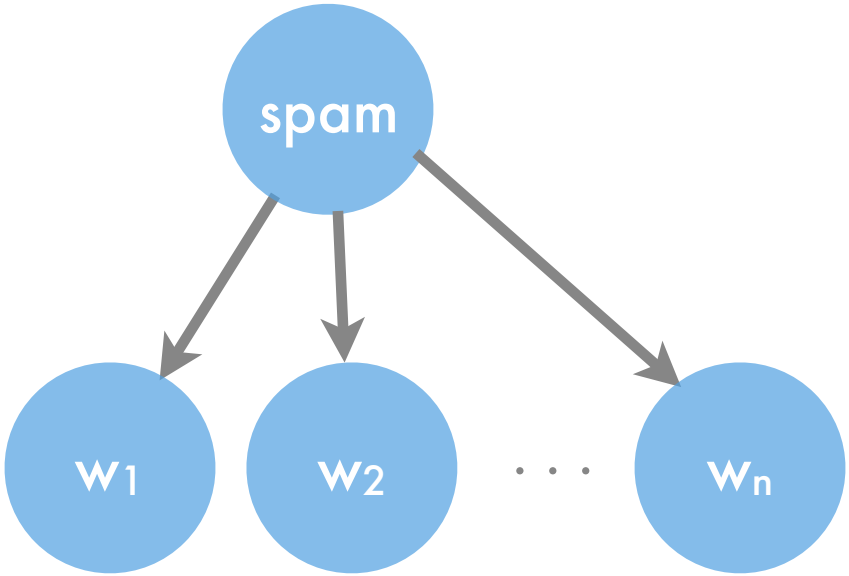


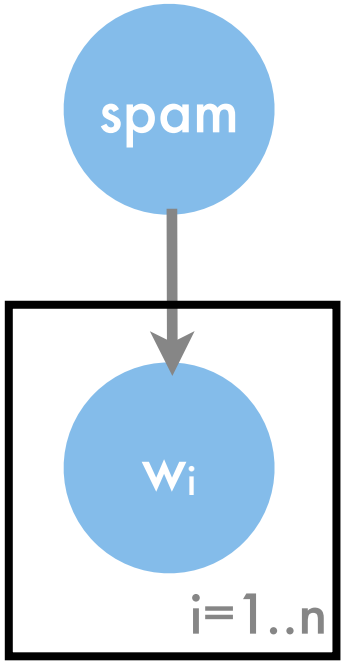
A Graphical Model



$$p(w_1, \ldots, w_n|\text{spam}) = \prod_{i=1}^{n} p(w_i|\text{spam})$$

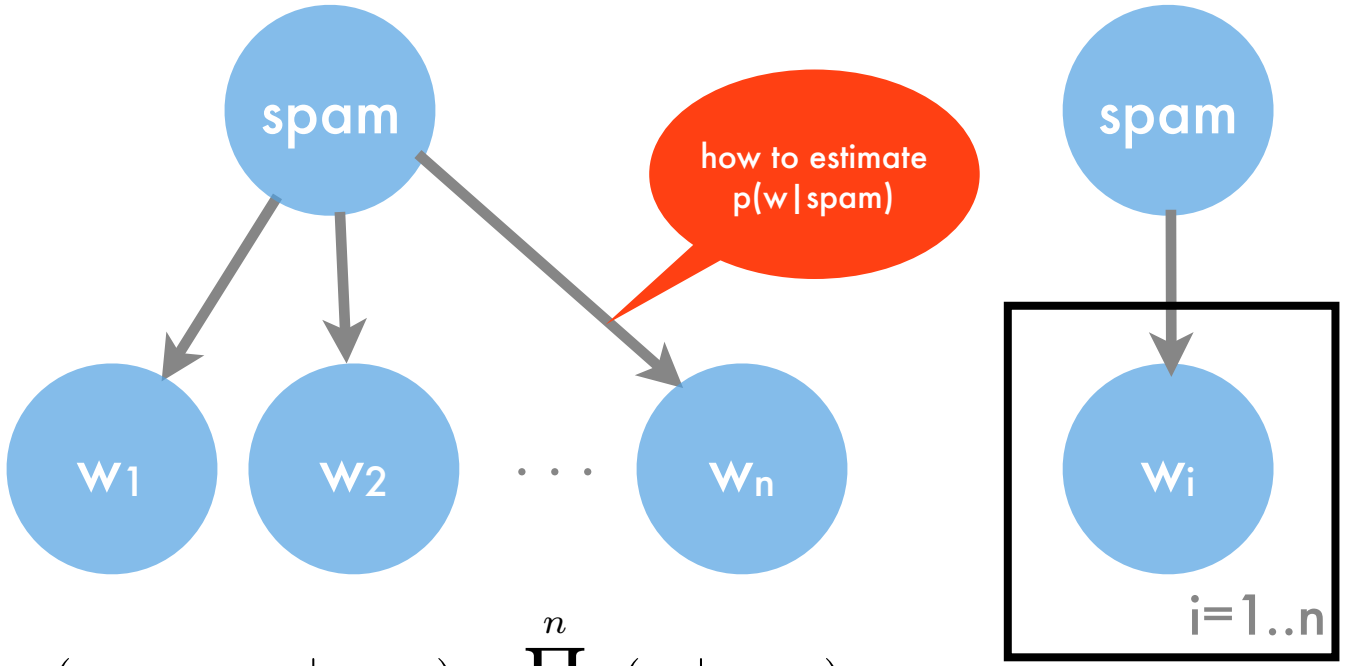
A Graphical Model





$$p(w_1, \ldots, w_n|\text{spam}) = \prod_{i=1}^{n} p(w_i|\text{spam})$$

A Graphical Model



$$p(w_1, \ldots, w_n|\text{spam}) = \prod_{i=1}^{n} p(w_i|\text{spam})$$

Naive Bayes Spam Filter

- Data
  - Emails (headers, body, metadata)
  - Labels (spam/ham)
    assume that users actually label all mails
  - Images and labels
- Need to estimate $p(y)$, $p(x_i|y)$
  - Compute distribution of x_i for every y
  - Compute distribution of y

this is a gross simplification

- date
- time
- recipient path
- IP number
- sender
- encoding
- many more features

```
Delivered-To: alex.smola@gmail.com
Received: by 10.216.47.73 with SMTP id s51cs361171web;
        Tue, 3 Jan 2012 14:17:53 -0800 (PST)
Received: by 10.213.17.145 with SMTP id s17mr2519891eba.147.1325629071725;
        Tue, 03 Jan 2012 14:17:51 -0800 (PST)
Return-Path: <alex+caf_=alex.smola=gmail.com@smola.org>
Received: from mail-ey0-f175.google.com (mail-ey0-f175.google.com [209.85.215.175])
        by mx.google.com with ESMTPS id n4si29264232eef.57.2012.01.03.14.17.51
        (version=TLSv1/SSLv3 cipher=OTHER);
        Tue, 03 Jan 2012 14:17:51 -0800 (PST)
Received-SPF: neutral (google.com: 209.85.215.175 is neither permitted nor denied by best guess
record for domain of alex+caf_=alex.smola=gmail.com@smola.org) client-ip=209.85.215.175;
Authentication-Results: mx.google.com; spf=neutral (google.com: 209.85.215.175 is neither
permitted nor denied by best guess record for domain of
alex+caf_=alex.smola=gmail.com@smola.org) smtp.mail=alex+caf_=alex.smola=gmail.com@smola.org;
dkim=pass (test mode) header.i=@googlemail.com
Received: by eaal1 with SMTP id l1so15092746eaa.6
        for <alex.smola@gmail.com>; Tue, 03 Jan 2012 14:17:51 -0800 (PST)
Received: by 10.205.135.18 with SMTP id ie18mr5325064bkc.72.1325629071362;
        Tue, 03 Jan 2012 14:17:51 -0800 (PST)
X-Forwarded-To: alex.smola@gmail.com
X-Forwarded-For: alex@smola.org alex.smola@gmail.com
Delivered-To: alex@smola.org
Received: by 10.204.65.198 with SMTP id k6cs206093bki;
        Tue, 3 Jan 2012 14:17:50 -0800 (PST)
Received: by 10.52.88.179 with SMTP id bh19mr10729402vdb.38.1325629068795;
        Tue, 03 Jan 2012 14:17:48 -0800 (PST)
Return-Path: <althoff.tim@googlemail.com>
Received: from mail-vx0-f179.google.com (mail-vx0-f179.google.com [209.85.220.179])
        by mx.google.com with ESMTPS id dt4si11767074vdb.93.2012.01.03.14.17.48
        (version=TLSv1/SSLv3 cipher=OTHER);
        Tue, 03 Jan 2012 14:17:48 -0800 (PST)
Received-SPF: pass (google.com: domain of althoff.tim@googlemail.com designates 209.85.220.179
as permitted sender) client-ip=209.85.220.179;
Received: by vcbf13 with SMTP id f13so11295098vcb.10
        for <alex@smola.org>; Tue, 03 Jan 2012 14:17:48 -0800 (PST)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
        d=googlemail.com; s=gamma;
        h=mime-version:sender:date:x-google-sender-auth:message-id:subject
         :from:to:content-type;
        bh=WCbdZ5sXac25dpH02XcRyDOdts993hKwsAVXpGrFh0w=;
        b=WK2B2+ExWnf/gvTkw6uUvKuP4XeoKnlJq3USYTm0RARK8dSFjyOQsIHeAP9Yssxp6O
         7ngGoTzYqd+ZsyJfvQcLAWp1PCJhG8AMcnqWkx0NMeoFvIp2HQooZwxSOCx5ZRgY+7qX
         uIbbdna4lUDXj6UFe16SpLDCkptd8OZ3gr7+o=
MIME-Version: 1.0
Received: by 10.220.108.81 with SMTP id e17mr24104004vcp.67.1325629067787;
 Tue, 03 Jan 2012 14:17:47 -0800 (PST)
Sender: althoff.tim@googlemail.com
Received: by 10.220.17.129 with HTTP; Tue, 3 Jan 2012 14:17:47 -0800 (PST)
Date: Tue, 3 Jan 2012 14:17:47 -0800
X-Google-Sender-Auth: 6bwi6D17HjZIkxOEol38NZzyeHs
Message-ID: <CAFJJHDGPBW+SdZg0MdAABiAKydDk9tpeMoDijYGjoGO-WC7osg@mail.gmail.com>
Subject: CS 281B. Advanced Topics in Learning and Decision Making
From: Tim Althoff <althoff@eecs.berkeley.edu>
To: alex@smola.org
Content-Type: multipart/alternative; boundary=f46d043c7af4b07e8d04b5a7113a

--f46d043c7af4b07e8d04b5a7113a
Content-Type: text/plain; charset=ISO-8859-1
```

Naive NaiveBayes Classifier

- Two classes (spam/ham)
- Binary features (e.g. presence of $$$, viagra)
- Simplistic Algorithm
  - Count occurrences of feature for spam/ham
  - Count number of spam/ham mails

feature probability

spam probability

$$p(x_i = \text{TRUE}|y) = \frac{n(i,y)}{n(y)} \text{ and } p(y) = \frac{n(y)}{n}$$

$$p(y|x) \propto \frac{n(y)}{n} \prod_{i:x_i=\text{TRUE}} \frac{n(i,y)}{n(y)} \prod_{i:x_i=\text{FALSE}} \frac{n(y)-n(i,y)}{n(y)}$$

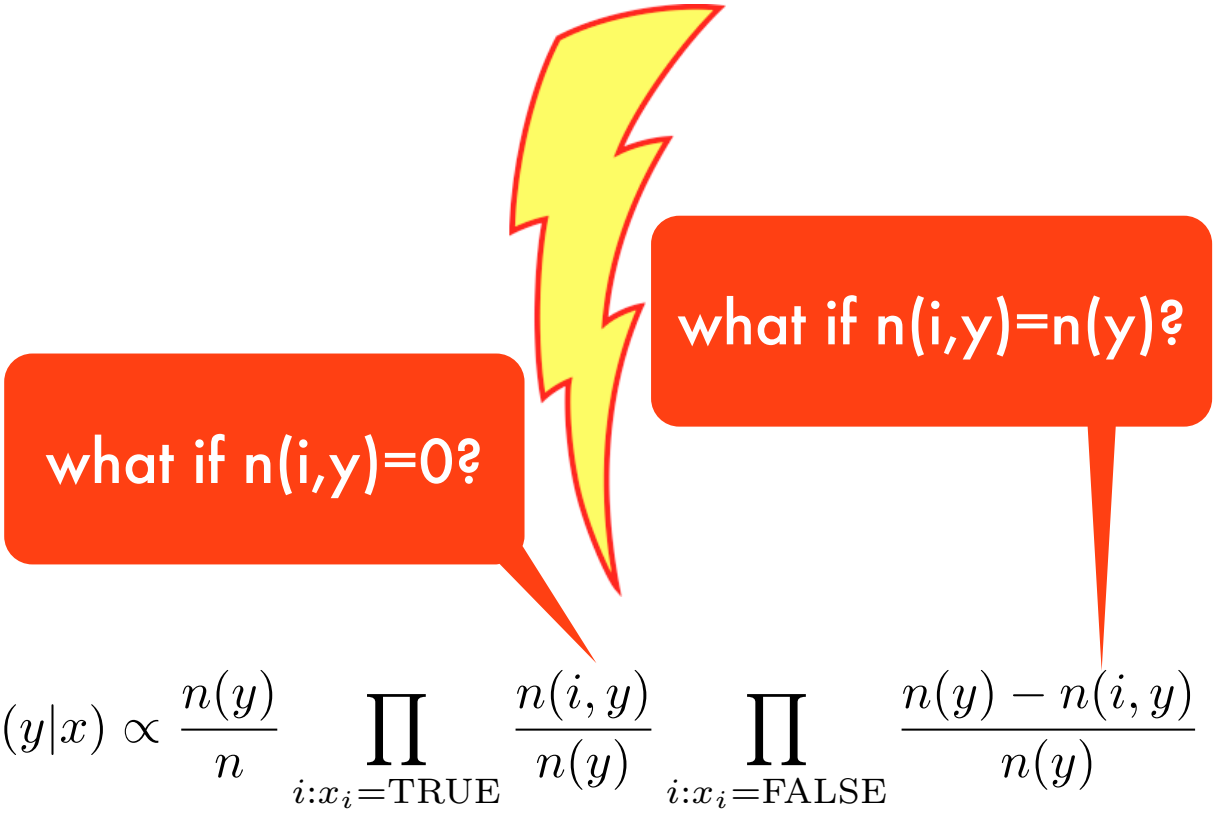
Naive NaiveBayes Classifier

what if n(i,y)=0?

what if n(i,y)=n(y)?

$$p(y|x) \propto \frac{n(y)}{n} \prod_{i:x_i=\text{TRUE}} \frac{n(i,y)}{n(y)} \prod_{i:x_i=\text{FALSE}} \frac{n(y)-n(i,y)}{n(y)}$$

Naive NaiveBayes Classifier



$$p(y|x) \propto \frac{n(y)}{n} \prod_{i:x_i=\text{TRUE}} \frac{n(i,y)}{n(y)} \prod_{i:x_i=\text{FALSE}} \frac{n(y)-n(i,y)}{n(y)}$$

Simple Algorithm

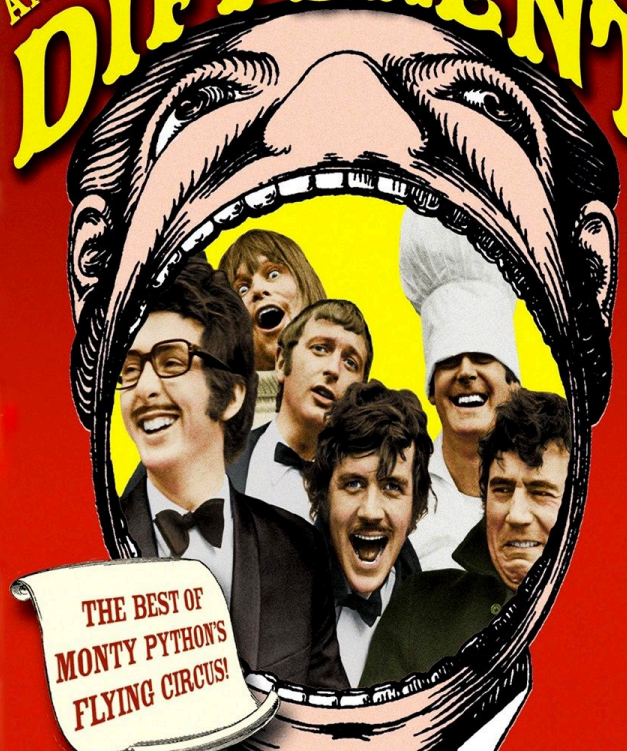
- For each document (x,y) do
  - Aggregate label counts given y
  - For each feature x_i in x do
    - Aggregate statistic for (x_i, y) for each y
- For y estimate distribution p(y)
- For each (x_i,y) pair do
  Estimate distribution $p(x_i|y)$, e.g. Parzen Windows, Exponential family (Gauss, Laplace, Poisson, ...), Mixture

- Given new instance compute $p(y|x) \propto p(y) \prod_j p(x_j|y)$

GRAHAM CHAPMAN JOHN CLEESE TERRY GILLIAM
ERIC IDLE TERRY JONES MICHAEL PALIN

MONTY PYTHON'S

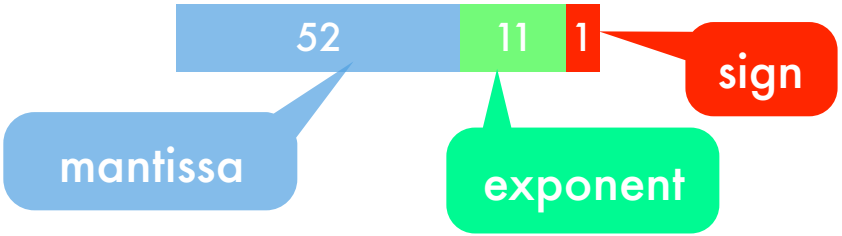
AND NOW FOR SOMETHING COMPLETELY DIFFERENT



THE BEST OF MONTY PYTHON'S FLYING CIRCUS!

Products of Probabilities in log-space

- Floating point numbers (FP64)



$$\pi = \log p$$

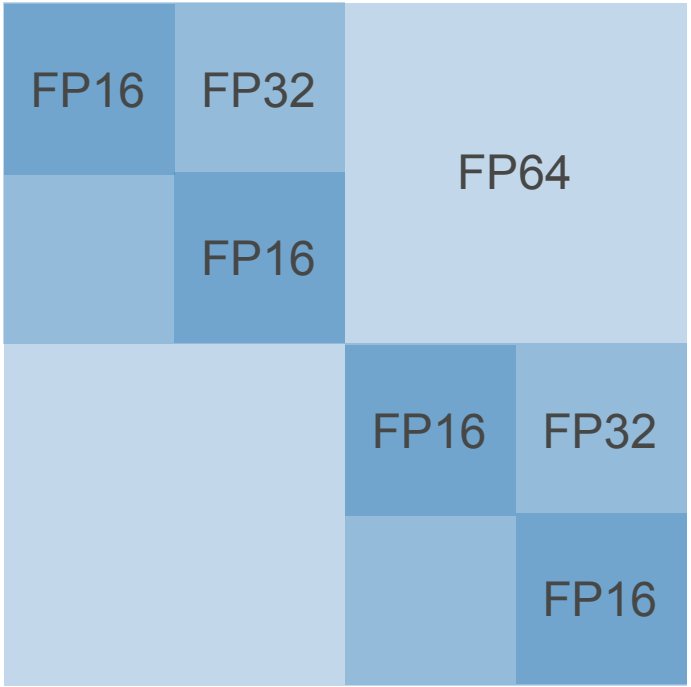
- Probabilities can be very small.
  In particular products of many probabilities. Underflow!
- Store data in mantissa, not exponent

$$\prod_i p_i \to \sum_i \pi_i \qquad \sum_i p_i \to \max \pi + \log \sum_i \exp\left[\pi_i - \max \pi\right]$$

Floating point numbers on GPUs

NVIDIA Titan RTX

- 400 GFlops FP64
- 13 TFlops FP32
- 27 TFlops FP16
- 107 TFlops FP16 Tensor cores
- 215 TFlops INT8 Tensor cores
- 430 TFlops INT4 Tensor cores



For fixed bandwidth, twice the number of operations

Overflow / underflow are dangerous

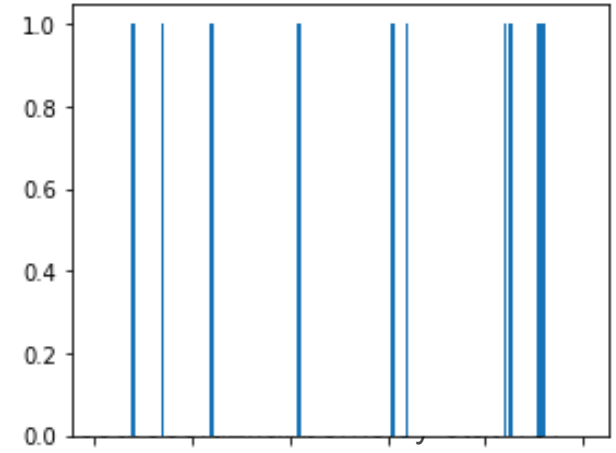
Sampling

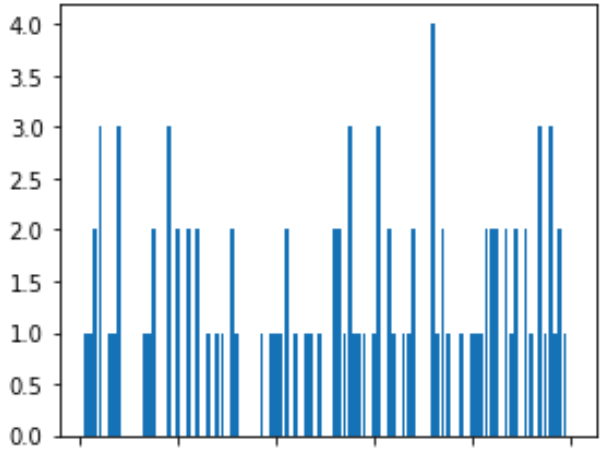
Uniform Distribution

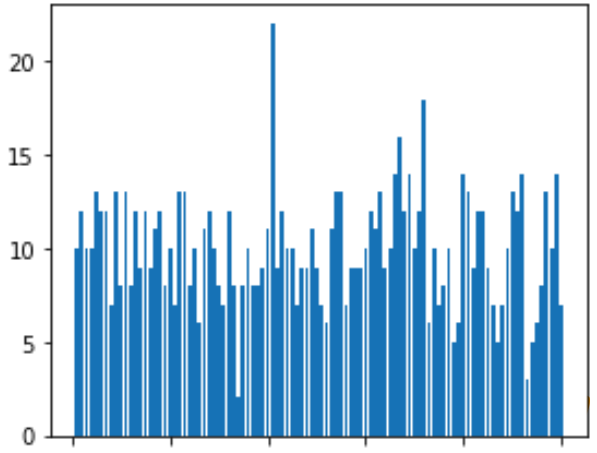
- Constant within an interval, zero outside

$$p(x) = \frac{1}{U - L} \text{ if } L \leq x \leq U$$

- Useful for initializing parameters or for load distribution







Discrete Distribution

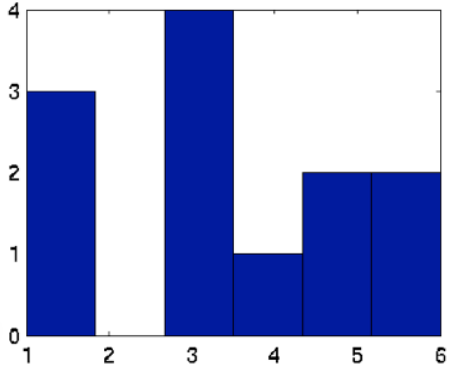
- **Discrete set of of outcomes**
  e.g. word distribution, distribution over classes, spam

the	and	house	see	is	white	a
0.1	0.05	0.05	0.1	0.2	0.2	0.3

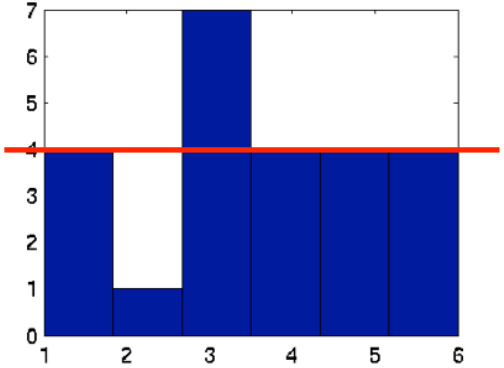
- **Sampling**
  - By brute force in O(n) time
  - Build a heap in O(log n) time with O(n) prep
- **Estimation** e.g. via softmax

Tossing a Dice

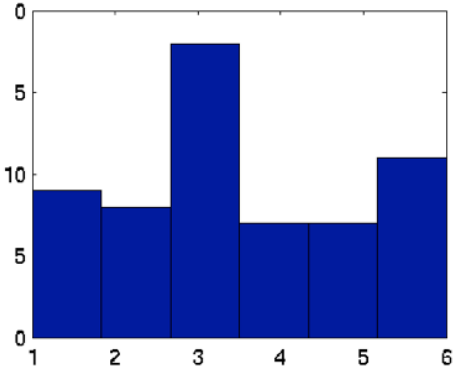
12



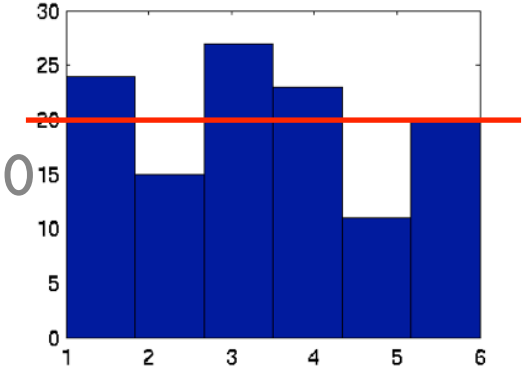
24



60

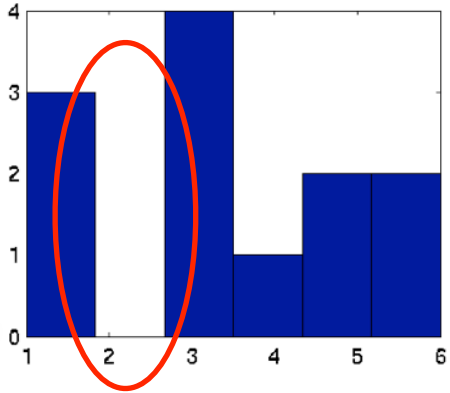


120

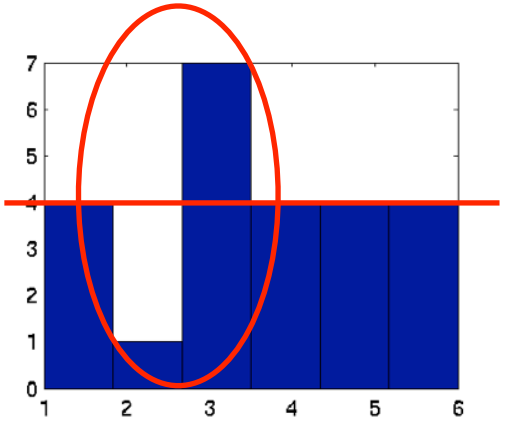


Tossing a Dice

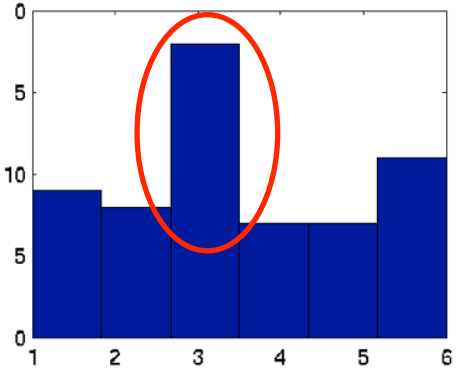
12



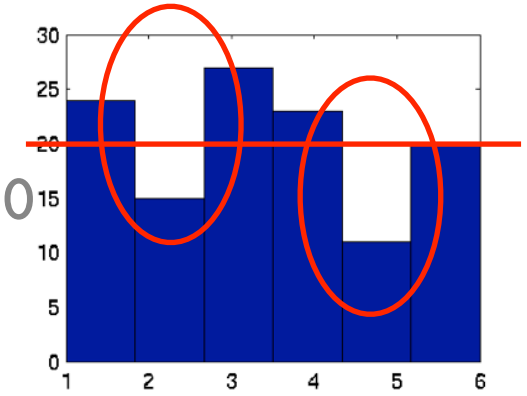
24



60

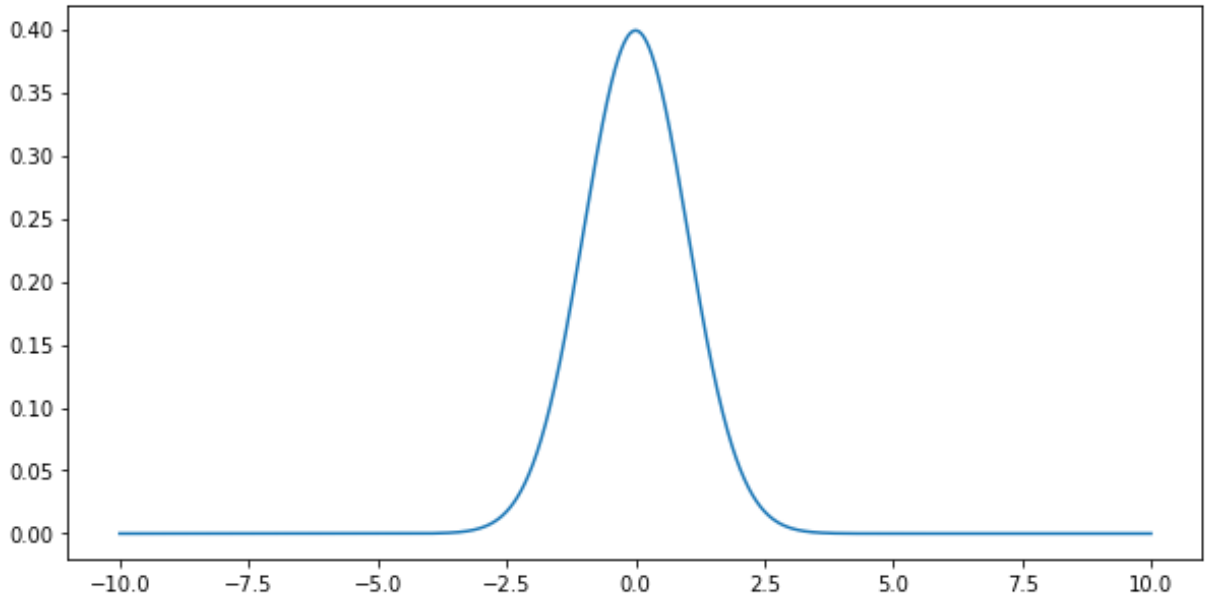


120



Normal Distribution

- Density $$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



Normal Distribution

- Density $$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$
- Mean $$\mathbf{E}[x] = \int x dp(x) = \mu$$
- Variance $$\mathbf{E}[(x-\mu)^2] = \int (x-\mu)^2 dp(x) = \mathbf{E}[x^2] - [\mathbf{E}[x]]^2 = \sigma^2$$

(prove this by using linearity of expectation)

Central Limit Theorem

The random variable below converges to a Gaussian.

$$z_n := \left[\sum_{i=1}^{n} \sigma_i^2\right]^{-\frac{1}{2}} \left[\sum_{i=1}^{n} x_i - \mu_i\right]$$

Practical application

Sums of independent random variables will converge to a normal distribution with joint variance.

Summary

- **Basic Probability**
  Random variables, conditional probabilities, Bayes rule
- **Naive Bayes**
  - Multiple tests
  - Examples - OCR
- **Sampling**
  - Distributions (categorial, normal, uniform)
  - Central limit theorem