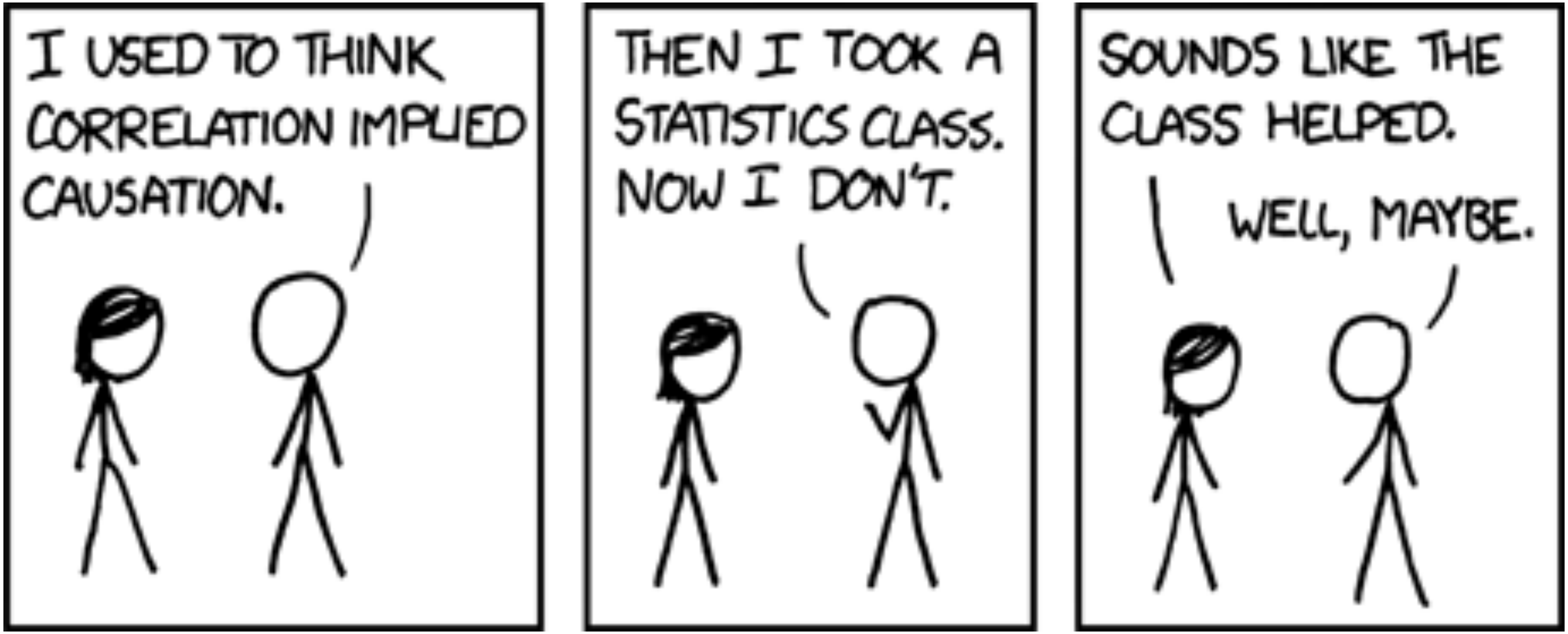
2.1 Probabilities

2 Statistics

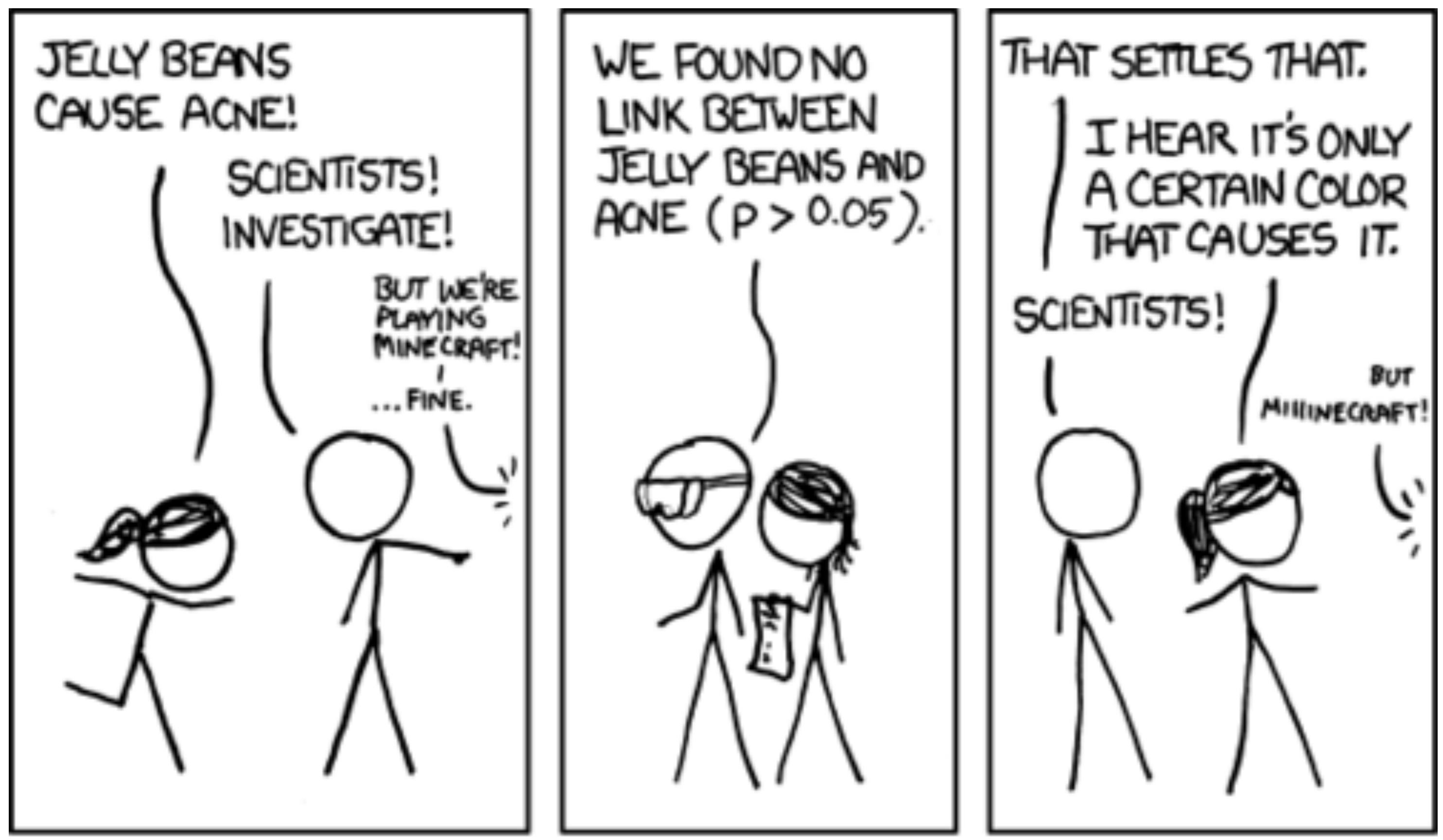
Alexander Smola
Introduction to Machine Learning 10-701
http://alex.smola.org/teaching/10-701-15

MACHINE LEARNING DEPARTMENT
Carnegie Mellon University

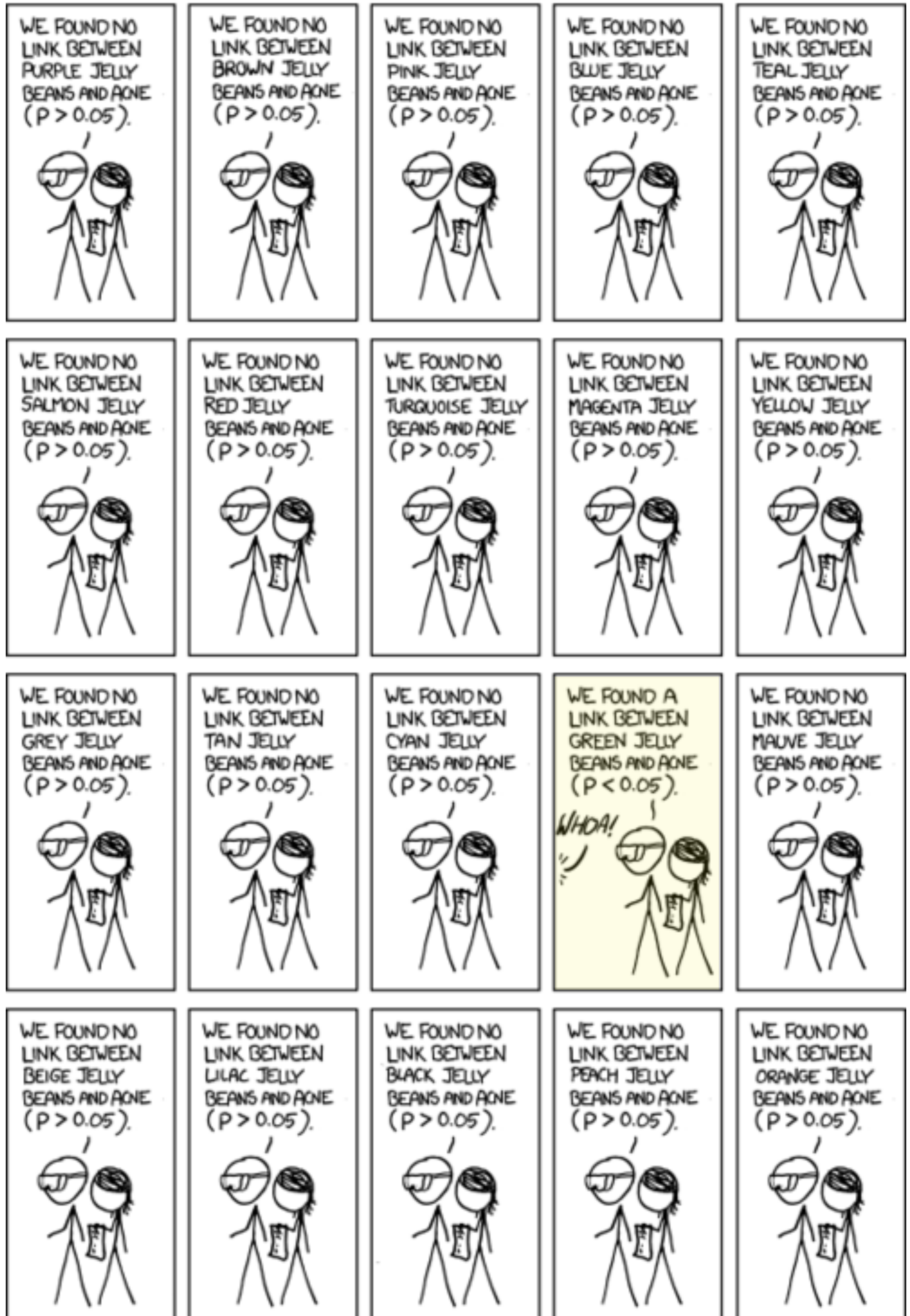


Basics

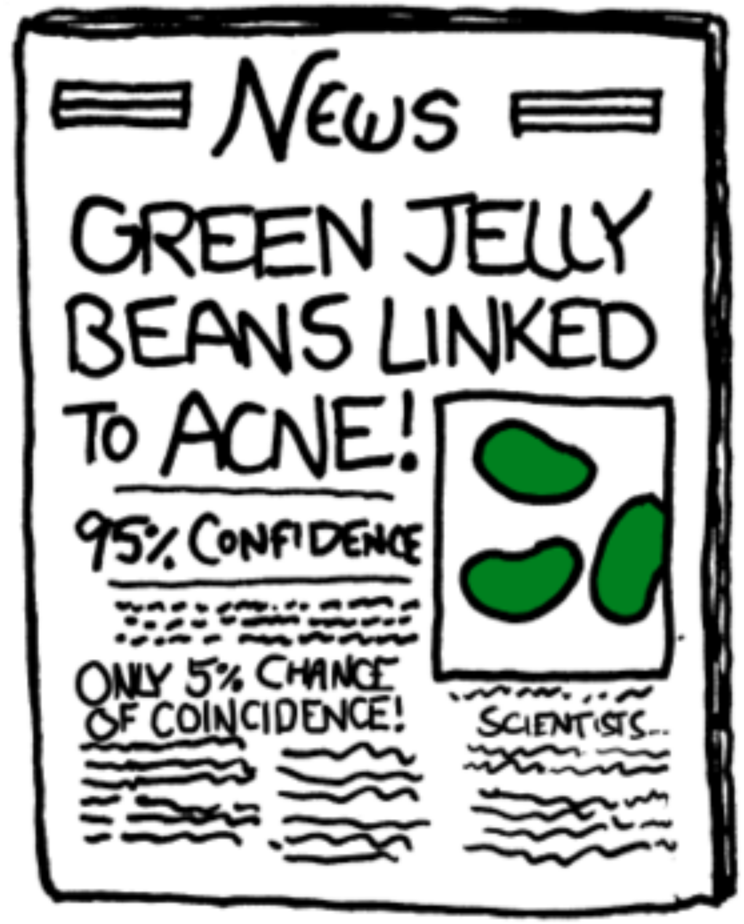
xkcd.com


JELLY BEANS CAUSE ACNE!
SCIENTISTS! INVESTIGATE!
BUT WE'RE PLAYING MINECRAFT!
...FINE.
WE FOUND NO LINK BETWEEN JELLY BEANS AND ACNE (P > 0.05).
THAT SETTLES THAT.
I HEAR IT'S ONLY A CERTAIN COLOR THAT CAUSES IT.
SCIENTISTS!
BUT MIIIINECRAFT!

xkcd.com


WE FOUND NO LINK BETWEEN PURPLE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN BROWN JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN PINK JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN BLUE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN TEAL JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN SALMON JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN RED JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN TURQUOISE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN MAGENTA JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN YELLOW JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN GREY JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN TAN JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN CYAN JELLY BEANS AND ACNE (P > 0.05).
WE FOUND A LINK BETWEEN GREEN JELLY BEANS AND ACNE (P < 0.05).
WHOA!
WE FOUND NO LINK BETWEEN MAUVE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN BEIGE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN LILAC JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN BLACK JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN PEACH JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN ORANGE JELLY BEANS AND ACNE (P > 0.05).

xkcd.com


NEWS
GREEN JELLY BEANS LINKED TO ACNE!
95% CONFIDENCE
ONLY 5% CHANCE OF COINCIDENCE!
SCIENTISTS...

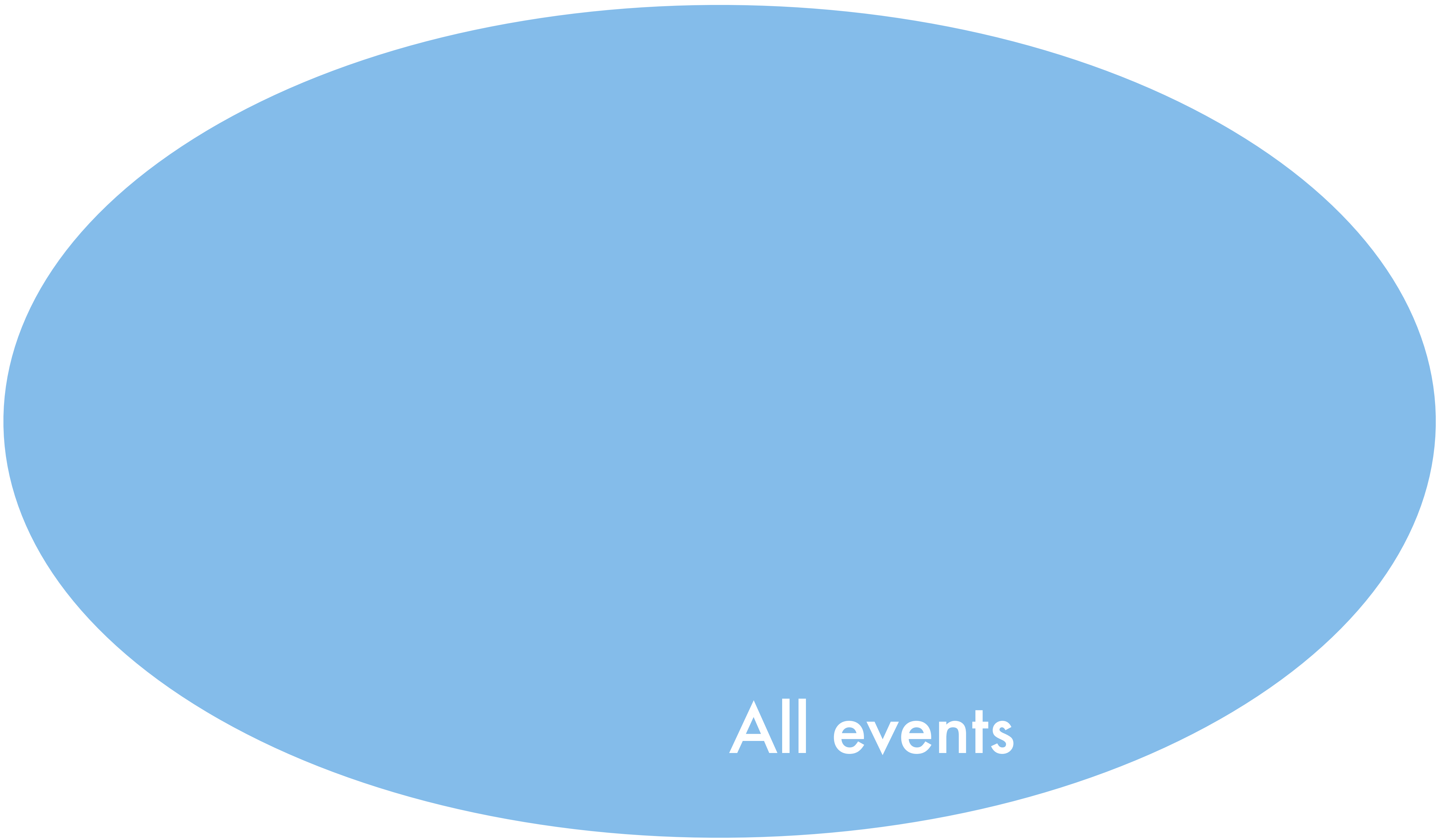
Probability

- 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)

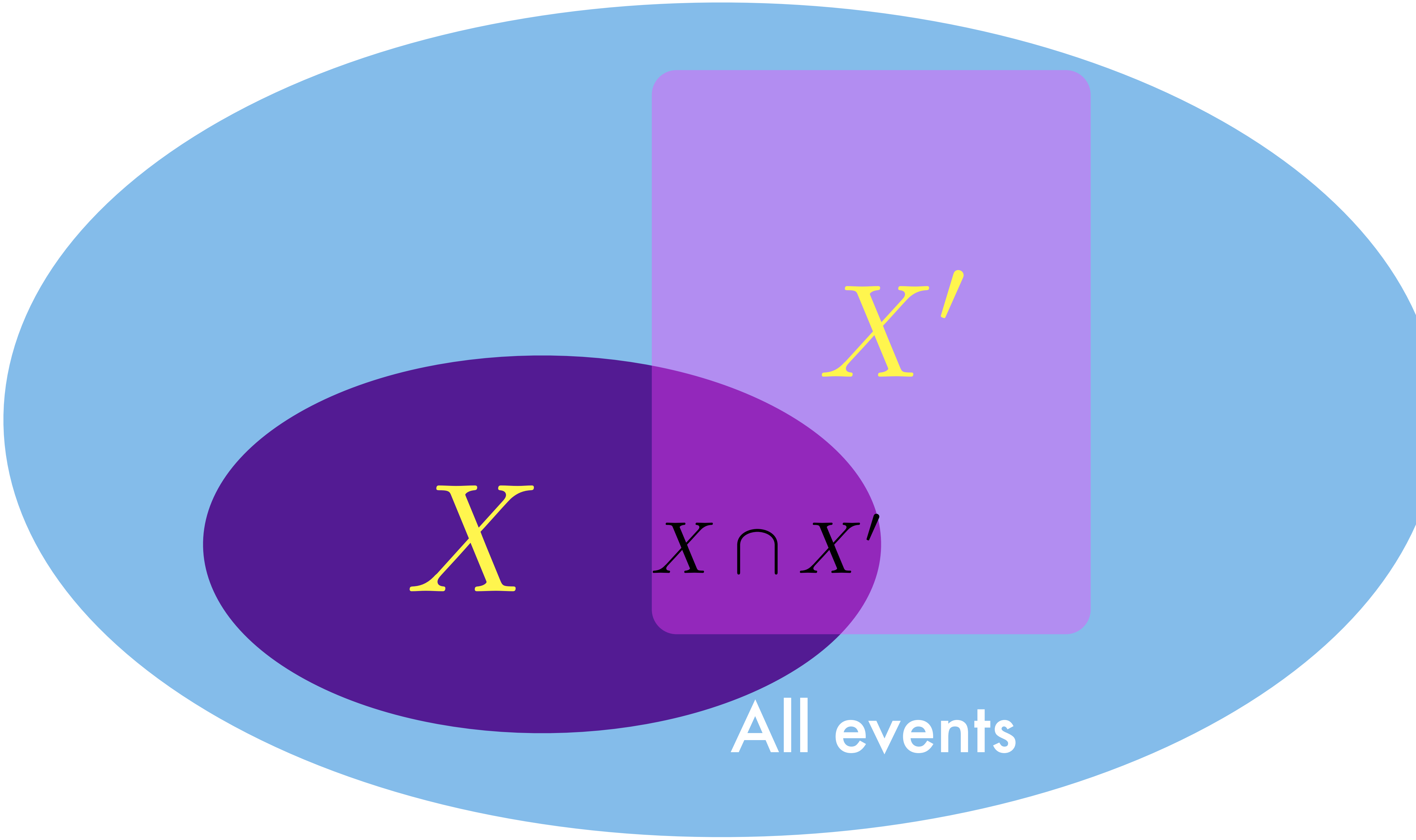
$$\Pr(X) \in [0,1],\ \Pr(\mathcal{X}) = 1$$
$$\Pr(\cup_i X_i) = \sum_i \Pr(X_i) \text{ if } X_i \cap X_j = \emptyset$$

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

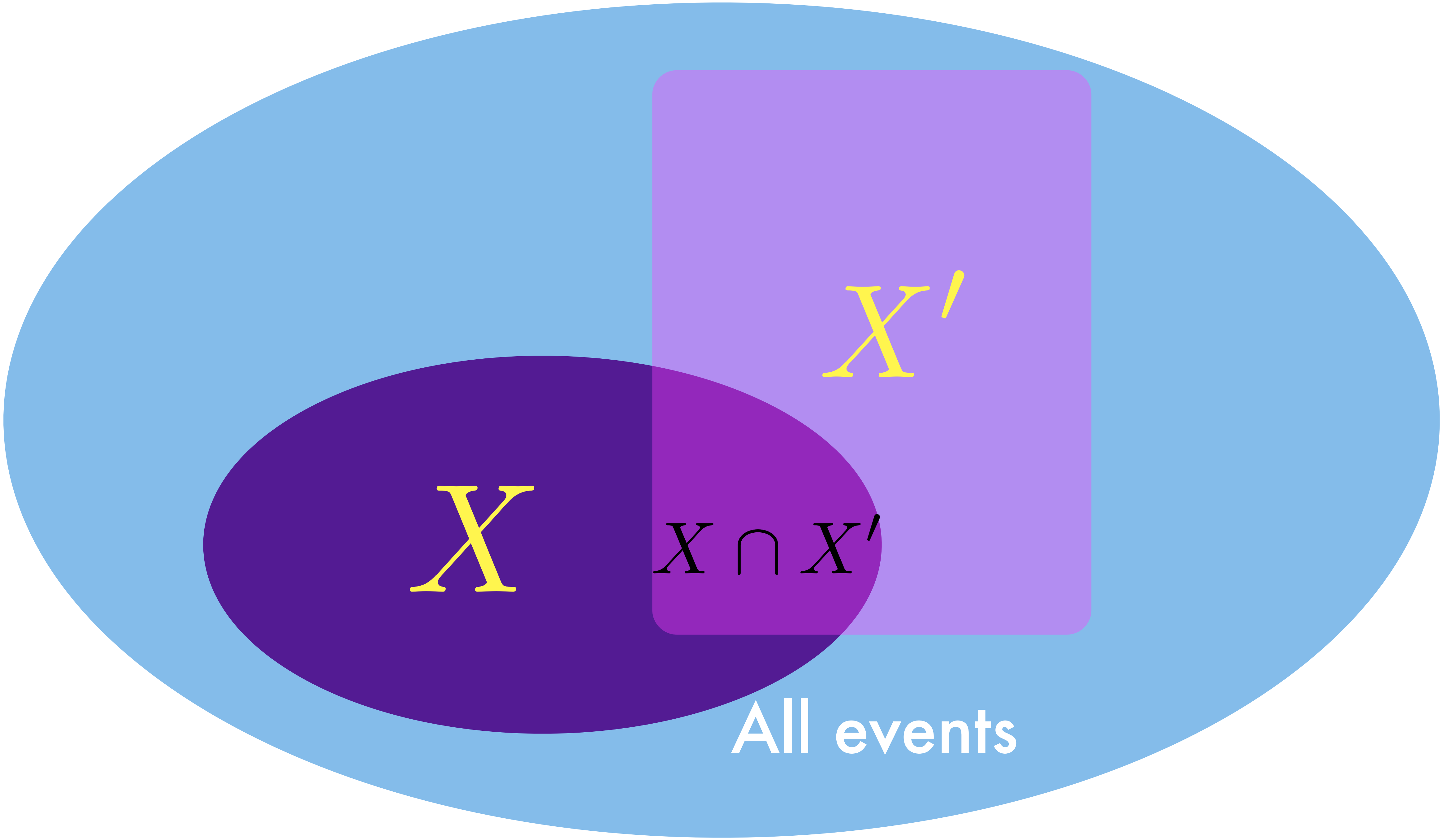
Venn Diagram



Venn Diagram



Venn Diagram



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

(In)dependence

- Independence $\Pr(x, y) = \Pr(x) \cdot \Pr(y)$
  - Login behavior of two users (approximately)
  - Disk crash in different colos (approximately)

(In)dependence

- Independence $$\Pr(x, y) = \Pr(x) \cdot \Pr(y)$$
  - Login behavior of two users (approximately)
  - Disk crash in different colos (approximately)
- Dependent events
  - Emails
  - Queries
  - News stream / Buzz / Tweets
  - IM communication
  - Russian Roulette

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

(In)dependence

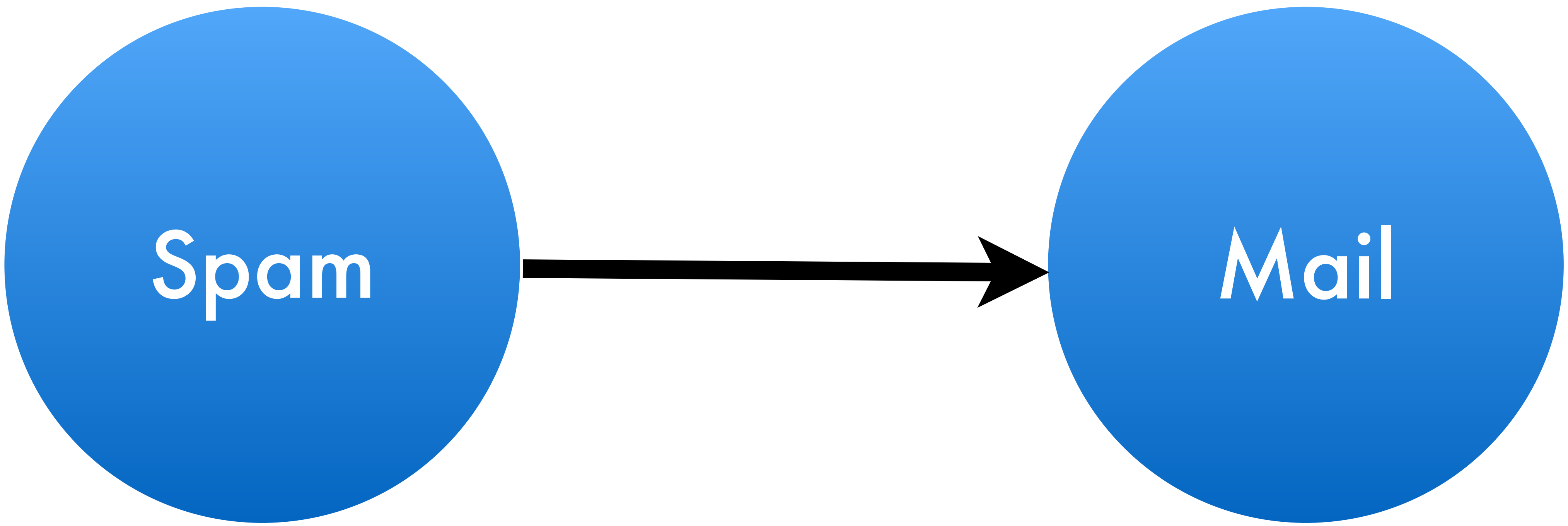
- Independence $\Pr(x, y) = \Pr(x) \cdot \Pr(y)$
  - Login behavior of two users (approximately)
  - Disk crash in different colos (approximately)
- Dependent events
  - Emails
  - Queries
  - News stream / Buzz / Tweets
  - IM communication
  - Russian Roulette

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

Everywhere!

A Graphical Model

$$p(\text{spam}, \text{mail}) = p(\text{spam})p(\text{mail}|\text{spam})$$



Bayes Rule

- Joint Probability

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

- Bayes Rule

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

- 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

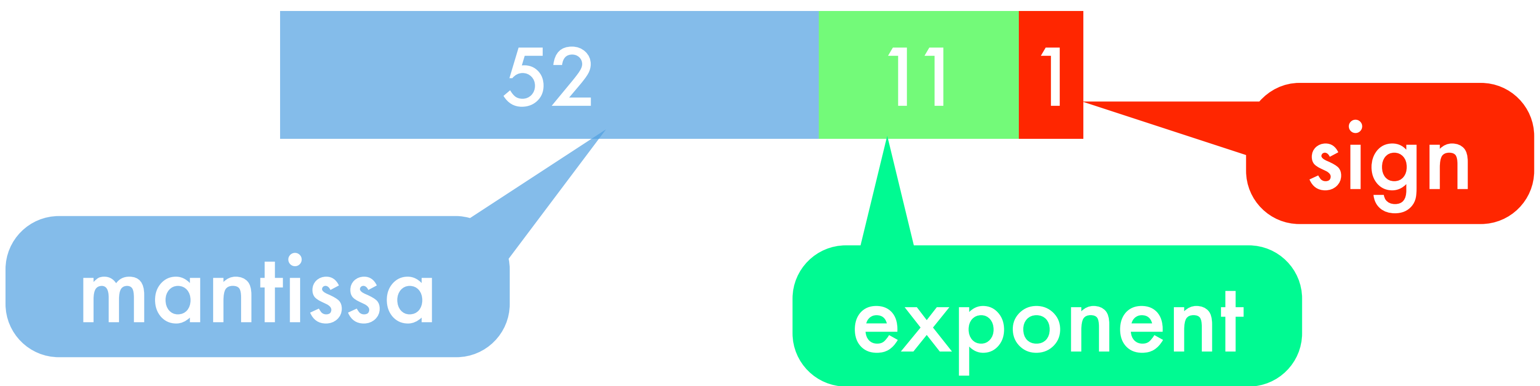
$$\frac{0.01 \cdot 0.05 \cdot 0.999}{1 \cdot 0.9 \cdot 0.001 + 0.01 \cdot 0.05 \cdot 0.999} = 0.357$$

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

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

Logarithms are good

- Floating point numbers



$$\pi = \log p$$

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

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


SPAM®
Classic
FILTERING
Ingredients: Pork with Ham, Salt, Water, Modified Potato Starch, Sugar, Sodium Nitrite
U.S. INSPECTED AND PASSED BY DEPARTMENT OF AGRICULTURE
Serving Suggestion
NET WT 12 OZ

Naive Bayes Spam Filter

Naive Bayes Spam Filter

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

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

- Spam classification via Bayes Rule

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 CMU stock.
- Buy Viagra. Make your CMU experience last longer.
- You deserve a PhD from CMU.
  We recognize your expertise.

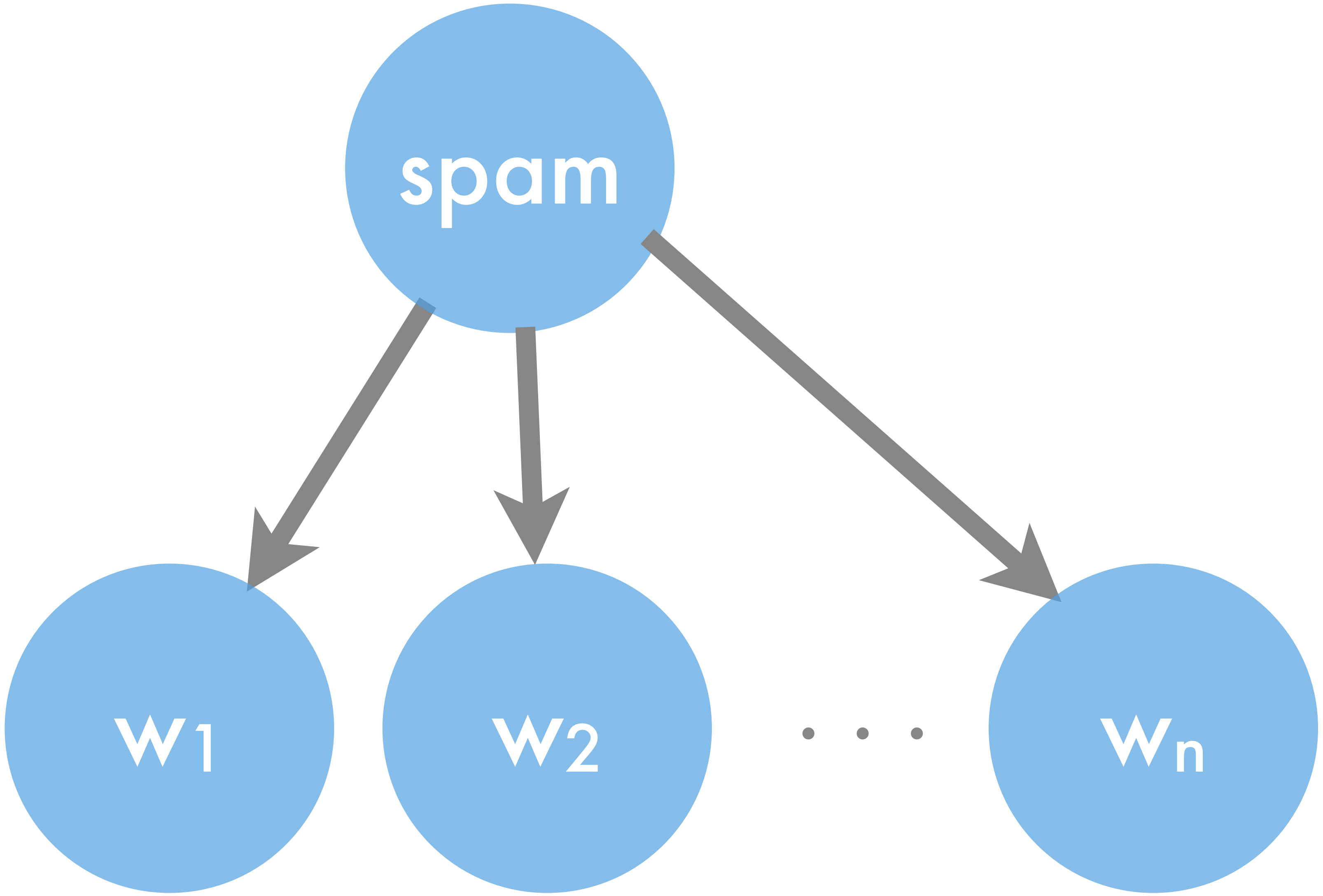
Naive Bayes Spam Filter

Equally likely phrases

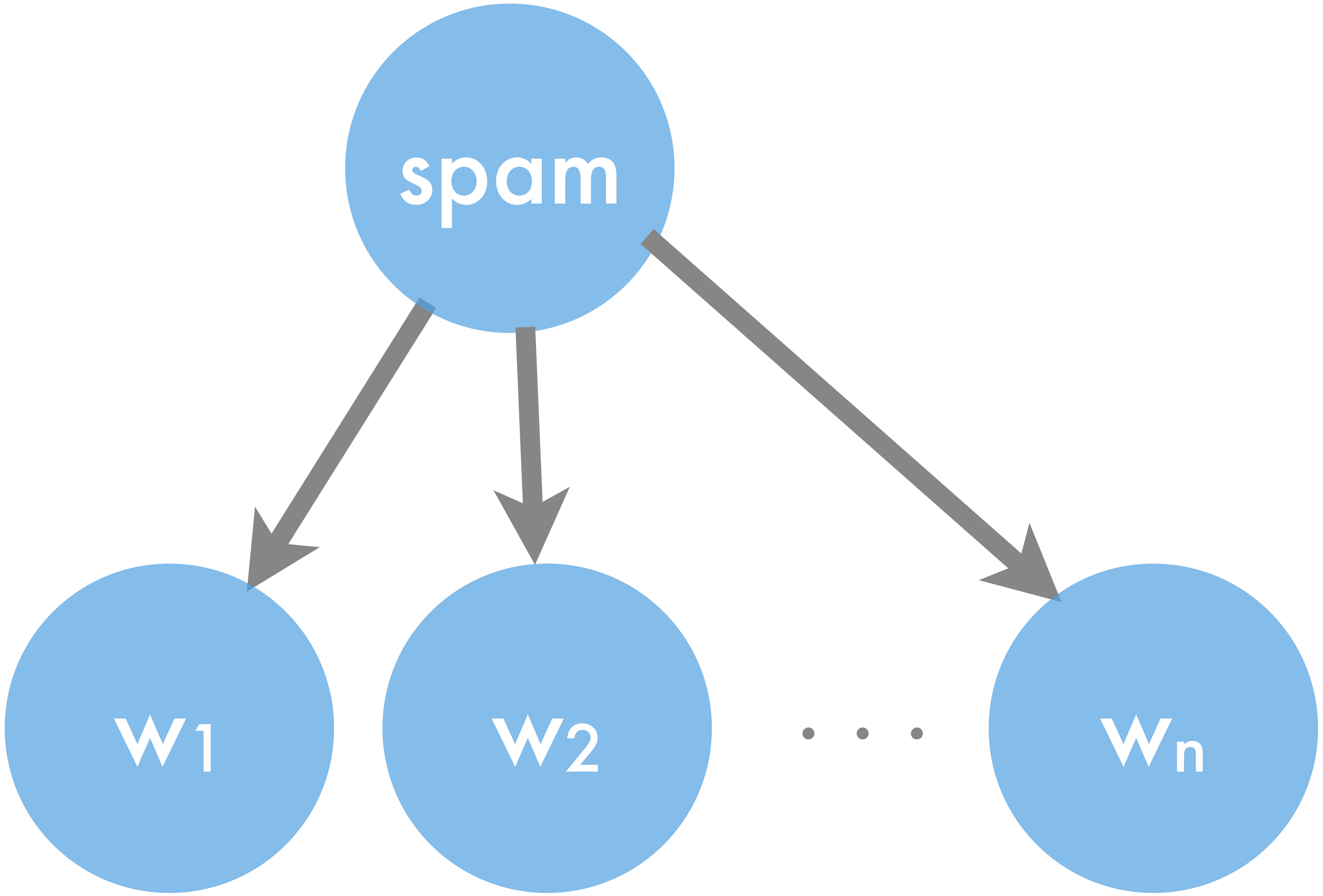
- Get rich quick. Buy CMU stock.
- Buy Viagra. Make your CMU experience last longer.
- You deserve a PhD from CMU.
  We recognize your expertise.

- Make your rich CMU PhD experience last longer.

A Graphical Model

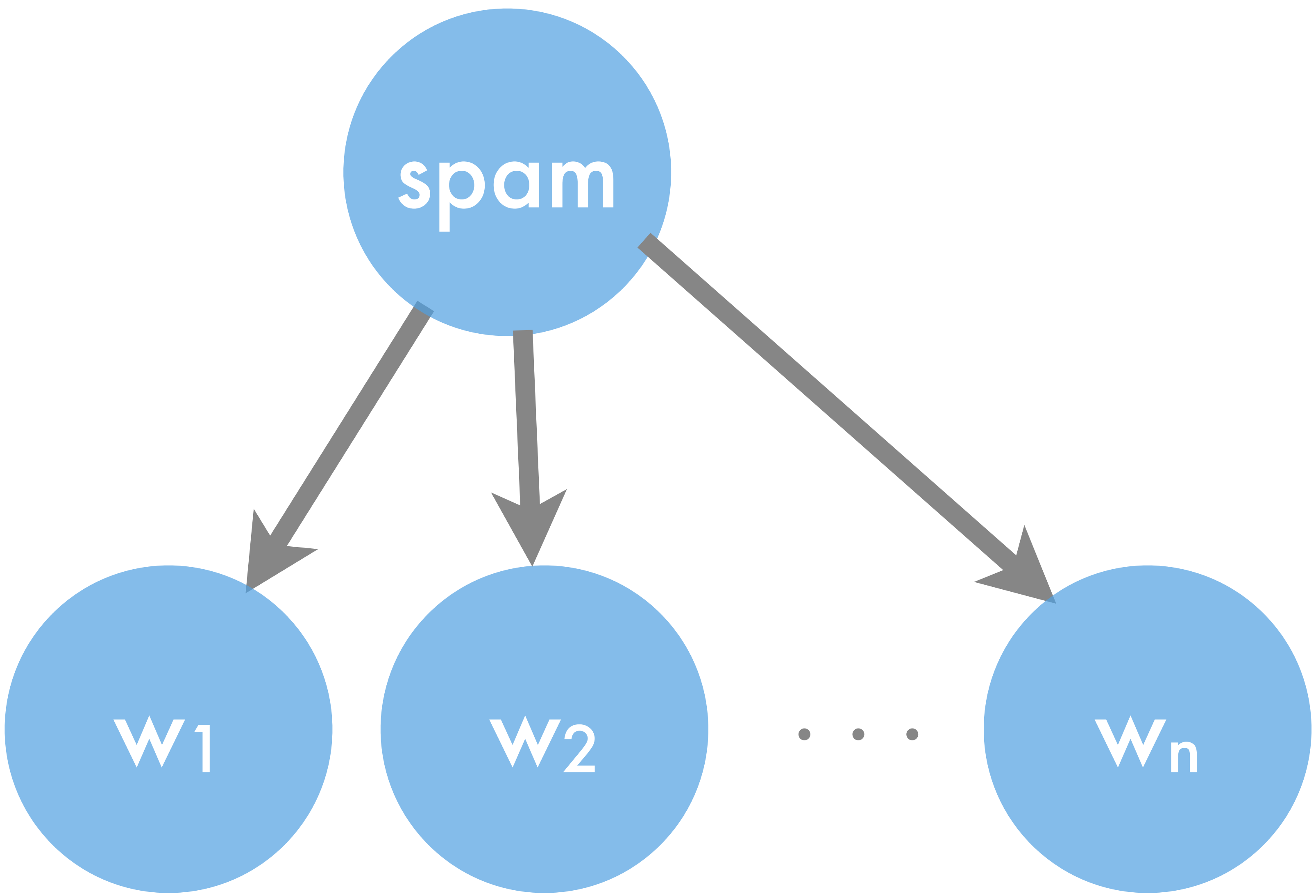


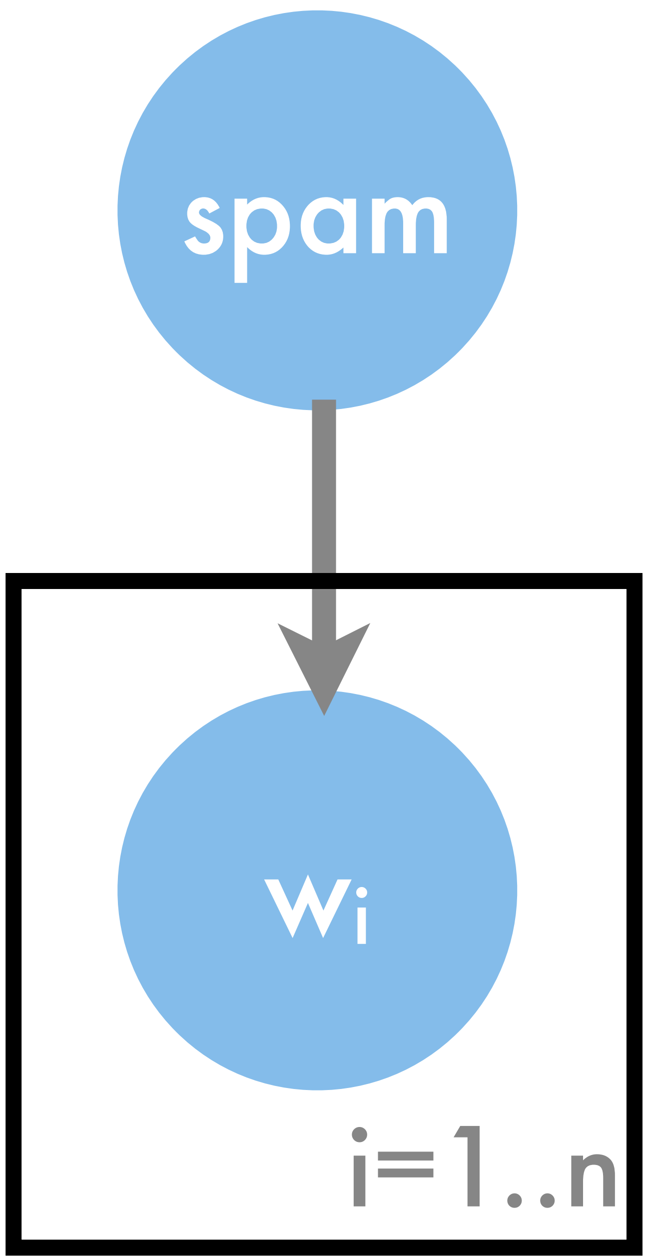
A Graphical Model



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

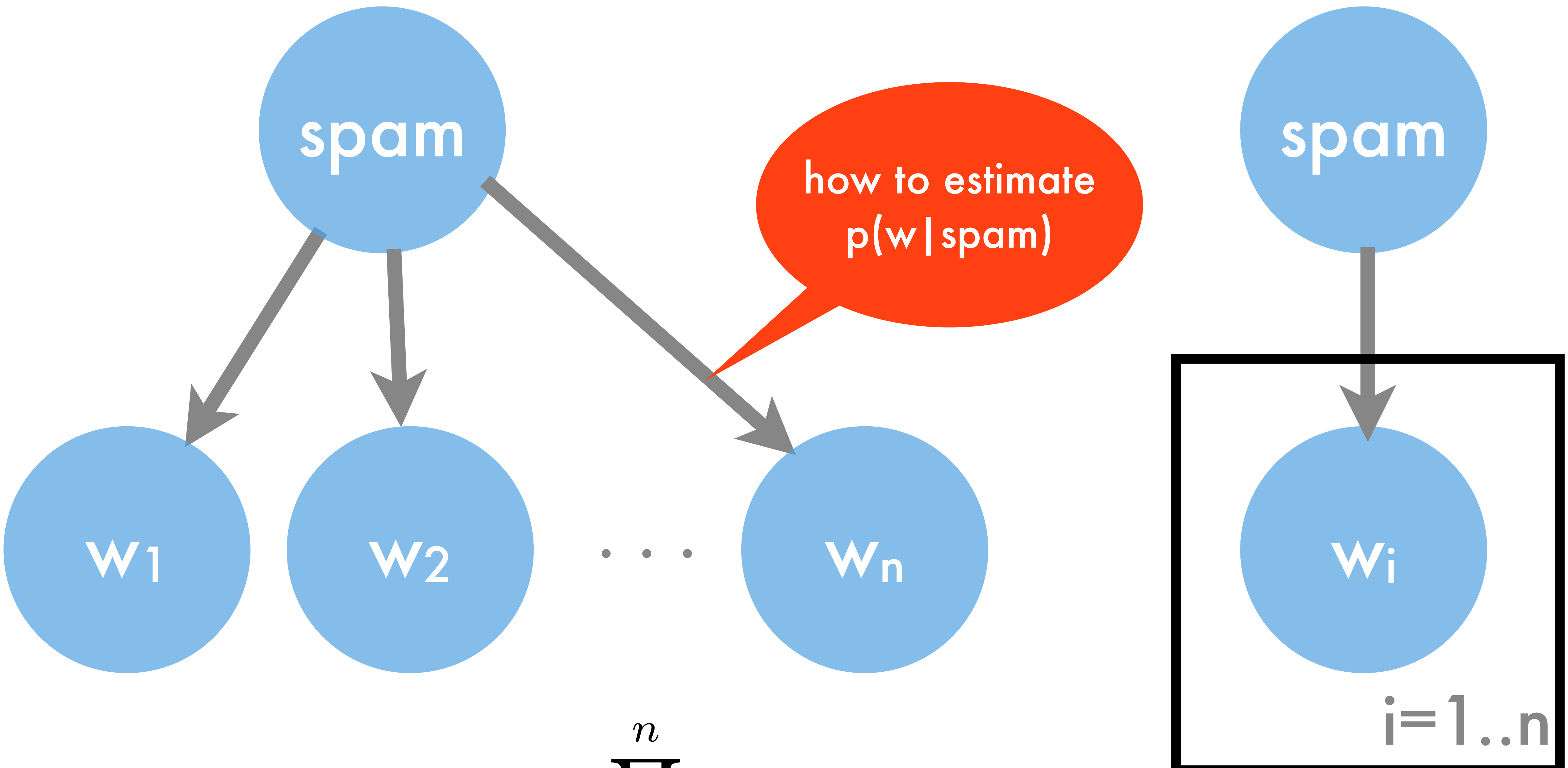
A Graphical Model





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

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
- Processing capability
  - Billions of e-mails
  - 1000s of servers
- Need to estimate p(y), $p(x_i|y)$
  - Compute distribution of x_i for every y
  - Compute distribution of y

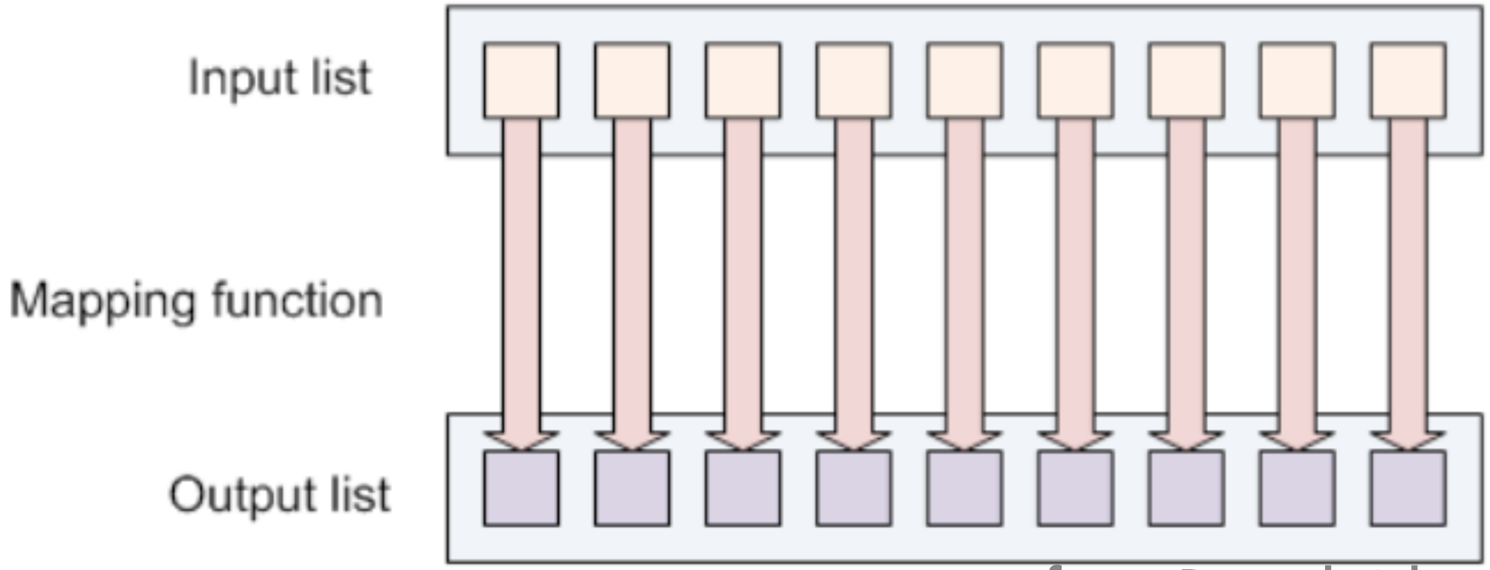
this is a gross simplification

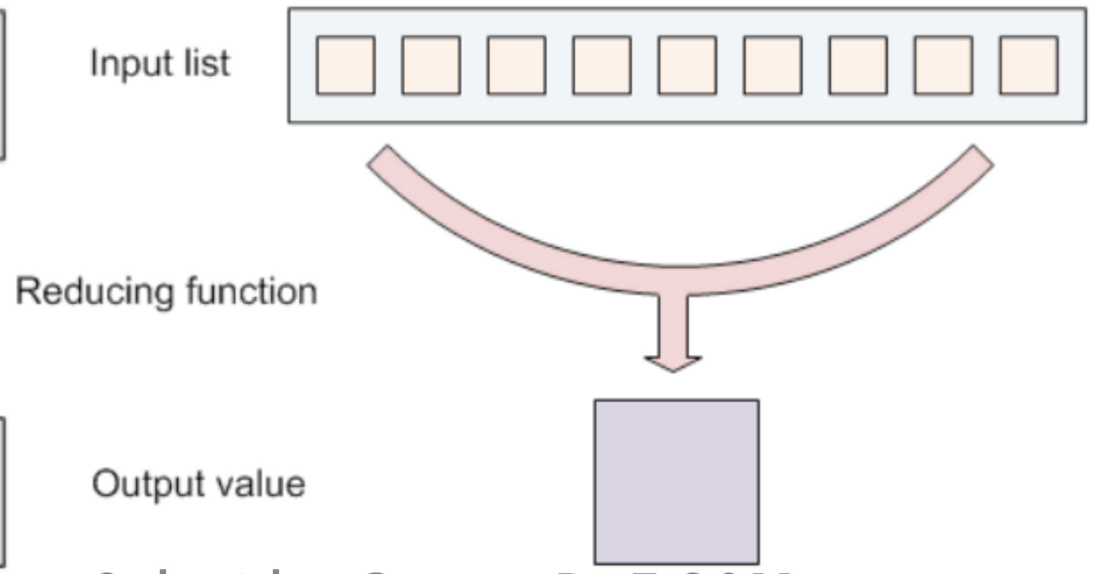
- date
- time
- recipient path
- IP number
- sender
- encoding
- many more features

```
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>
```

Preview - Map Reduce

- 1000s of (faulty) machines
- Lots of jobs are mostly embarrassingly parallel (except for a sorting/transpose phase)
- Functional programming origins
  - Map(key,value)
    processes each (key,value) pair and outputs a new (key,value) pair
  - Reduce(key,value)
    reduces all instances with same key to aggregate





from Ramakrishnan, Sakrejda, Canon, DoE 2011

Preview - Map Reduce

- 1000s of (faulty) machines
- Lots of jobs are mostly embarrassingly parallel
  (except for a sorting/transpose phase)
- Functional programming origins
  - Map(key,value)
    processes each (key,value) pair and outputs a new (key,value) pair
  - Reduce(key,value)
    reduces all instances with same key to aggregate
- Example - extremely naive wordcount
  - Map(docID, document)
    for each document emit many (wordID, count) pairs
  - Reduce(wordID, count)
    sum over all counts for given wordID and emit (wordID, aggregate)

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

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

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

Naive NaiveBayes

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

what if n(i,y)=0?

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

Naive NaiveBayes

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

what if n(i,y)=0?

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

Basic 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

$$p(y|x) \propto p(y) \prod_j p(x_j|y)$$

Basic 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

pass over all data

- 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

trivially parallel

$$p(y|x) \propto p(y) \prod_j p(x_j|y)$$

MapReduce Variant

- Map(document (x,y))
  - For each mapper for each feature x_i in x do
    - Aggregate statistic for (x_i, y) for each y
  - Send statistics (key = (x_i,y), value = counts) to reducer
- Reduce(x_i, y)
  - Aggregate over all messages from mappers
  - Estimate distribution $p(x_i|y)$, e.g. Parzen Windows, Exponential family (Gauss, Laplace, Poisson, ...), Mixture
  - Send coordinate-wise model to global storage
- Given new instance compute

$$p(y|x) \propto p(y) \prod_j p(x_j|y)$$

MapReduce Variant

- Map(document (x,y))
  - For each mapper for each feature x_i in x do
    - Aggregate statistic for (x_i, y) for each y
  - Send statistics (key = (x_i,y), value = counts) to reducer

local per mapper

- Reduce(x_i, y)
  - Aggregate over all messages from mappers
  - Estimate distribution $p(x_i|y)$, e.g. Parzen Windows, Exponential family (Gauss, Laplace, Poisson, ...), Mixture
  - Send coordinate-wise model to global storage
- Given new instance compute

only aggregates needed

$$p(y|x) \propto p(y) \prod_j p(x_j|y)$$

Estimating Probabilities

Binomial Distribution

- Two outcomes (head, tail); (0,1)
- Data likelihood

$$p(X;\pi) = \pi^{n_1}(1-\pi)^{n_0}$$

- Maximum Likelihood Estimation
  - Constrained optimization problem $\pi \in [0,1]$
  - Incorporate constraint via $p(x;\theta) = \frac{e^{x\theta}}{1+e^{\theta}}$
  - Taking derivatives yields

$$\theta = \log \frac{n_1}{n_0+n_1} \iff p(x=1) = \frac{n_1}{n_0+n_1}$$

... in detail ...

$$p(X;\theta) = \prod_{i=1}^{n} p(x_i;\theta) = \prod_{i=1}^{n} \frac{e^{\theta x_i}}{1+e^{\theta}}$$

$$\implies \log p(X;\theta) = \theta \sum_{i=1}^{n} x_i - n \log\left[1+e^{\theta}\right]$$

$$\implies \partial_\theta \log p(X;\theta) = \sum_{i=1}^{n} x_i - n \frac{e^{\theta}}{1+e^{\theta}}$$

$$\iff \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{e^{\theta}}{1+e^{\theta}} = p(x=1)$$

empirical probability of x=1

Discrete Distribution

- n outcomes (e.g. USA, Canada, India, UK, NZ)
- Data likelihood

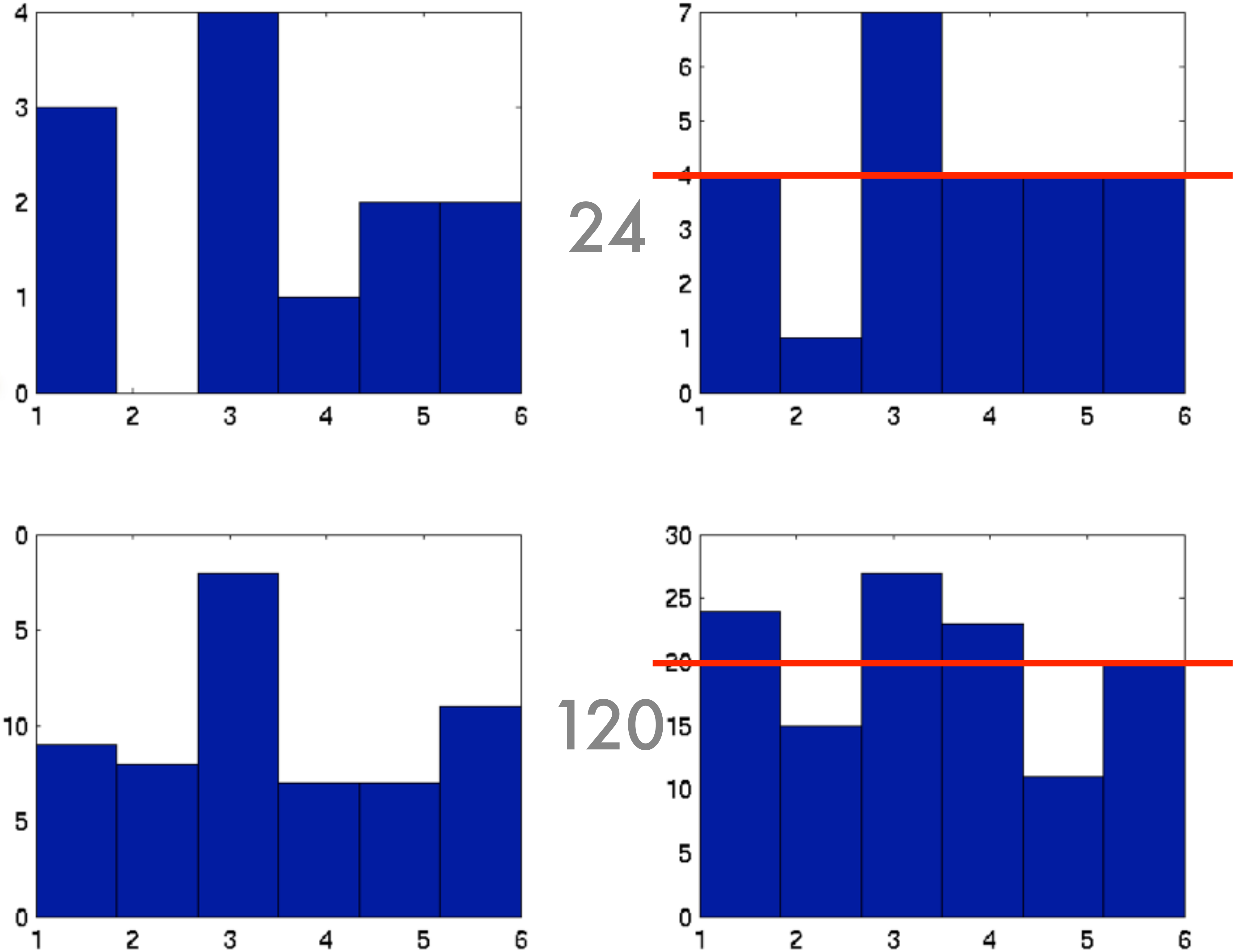
$$p(X;\pi) = \prod_i \pi_i^{n_i}$$

- Maximum Likelihood Estimation
  - Constrained optimization problem ... or ...
  - Incorporate constraint via
  - Taking derivatives yields

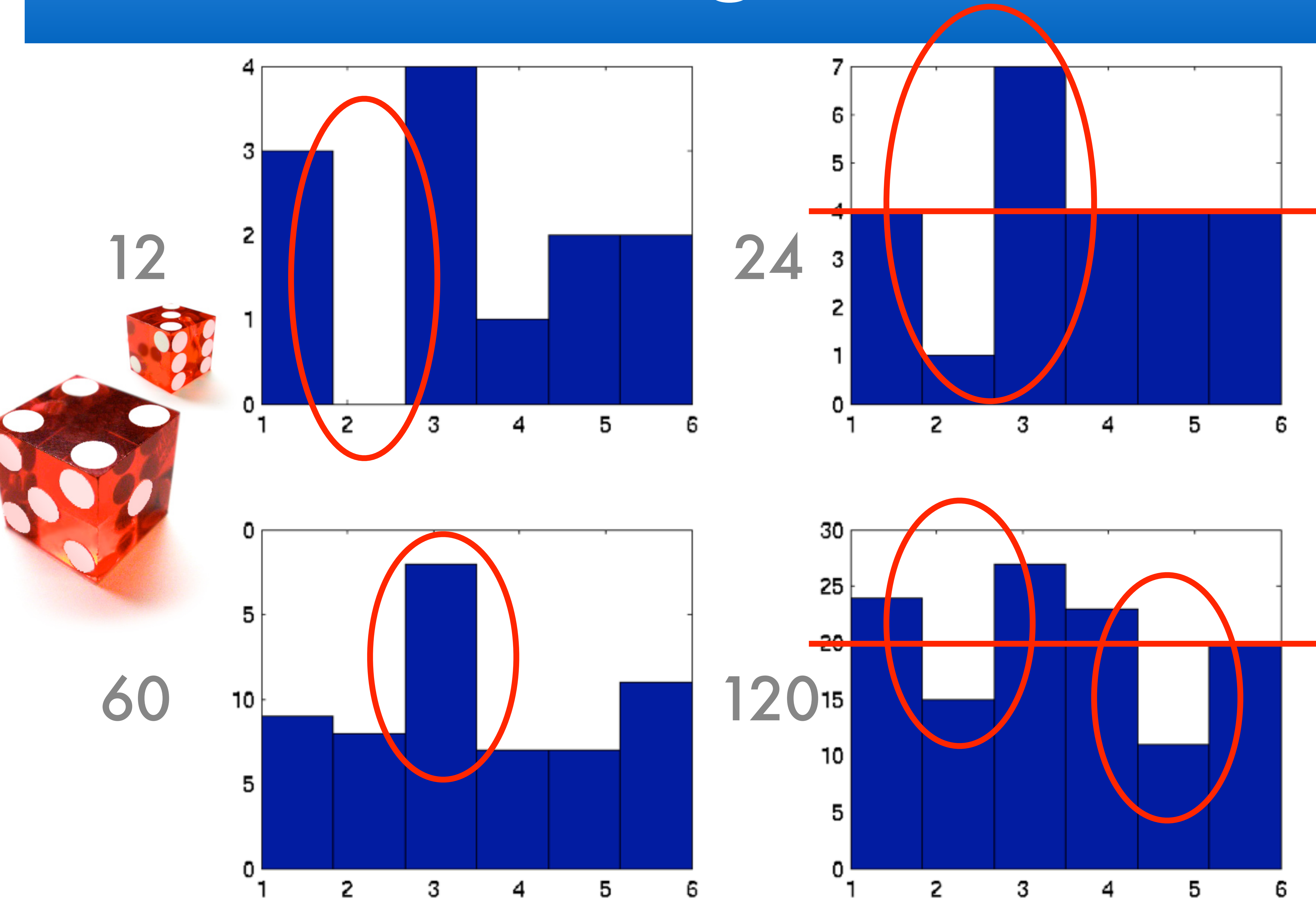
$$p(x;\theta) = \frac{\exp \theta_x}{\sum_{x'} \exp \theta_{x'}}$$

$$\theta_i = \log \frac{n_i}{\sum_j n_j} \iff p(x = i) = \frac{n_i}{\sum_j n_j}$$

Tossing a Dice



Tossing a Dice



Key Questions

- Do empirical averages converge?
  - Probabilities
  - Means / moments
- Rate of convergence and limit distribution
- Worst case guarantees
- Using prior knowledge

drug testing, semiconductor fabs
computational advertising
user interface design ...

2.2 Tail Bounds

2 Statistics

Alexander Smola
Introduction to Machine Learning 10-701
http://alex.smola.org/teaching/10-701-15

Convergence of Estimates

Expectations

- Random variable x with probability measure
- Expected value of f(x)

$$\mathbf{E}[f(x)] = \int f(x) dp(x)$$

- Special case - discrete probability mass

$$\Pr\{x = c\} = \mathbf{E}[\{x = c\}] = \int \{x = c\}\, dp(x)$$

(same trick works for intervals)

- Draw x_i identically and independently from p
- Empirical average

$$\mathbf{E}_{\mathrm{emp}}[f(x)] = \frac{1}{n}\sum_{i=1}^{n} f(x_i) \text{ and } \Pr_{\mathrm{emp}}\{x = c\} = \frac{1}{n}\sum_{i=1}^{n}\{x_i = c\}$$

Deviations

- Gambler rolls dice 100 times

$$\hat{P}(X = 6) = \frac{1}{n} \sum_{i=1}^{n} \{x_i = 6\}$$

- '6' only occurs 11 times. Fair number is16.7

IS THE DICE TAINTED?

- Probability of seeing '6' at most 11 times

$$\Pr(X \leq 11) = \sum_{i=0}^{11} p(i) = \sum_{i=0}^{11} \binom{100}{i} \left[\frac{1}{6}\right]^{i} \left[\frac{5}{6}\right]^{100-i} \approx 7.0\%$$

It's probably OK ... can we develop general theory?

Deviations

- Gambler rolls dice 100 times

$$\hat{P}(X = 6) = \frac{1}{n} \sum_{i=1}^{n} \{x_i = 6\}$$

- '6' only occurs 11 times. Fair number is16.7

IS THE DICE TAINTED?

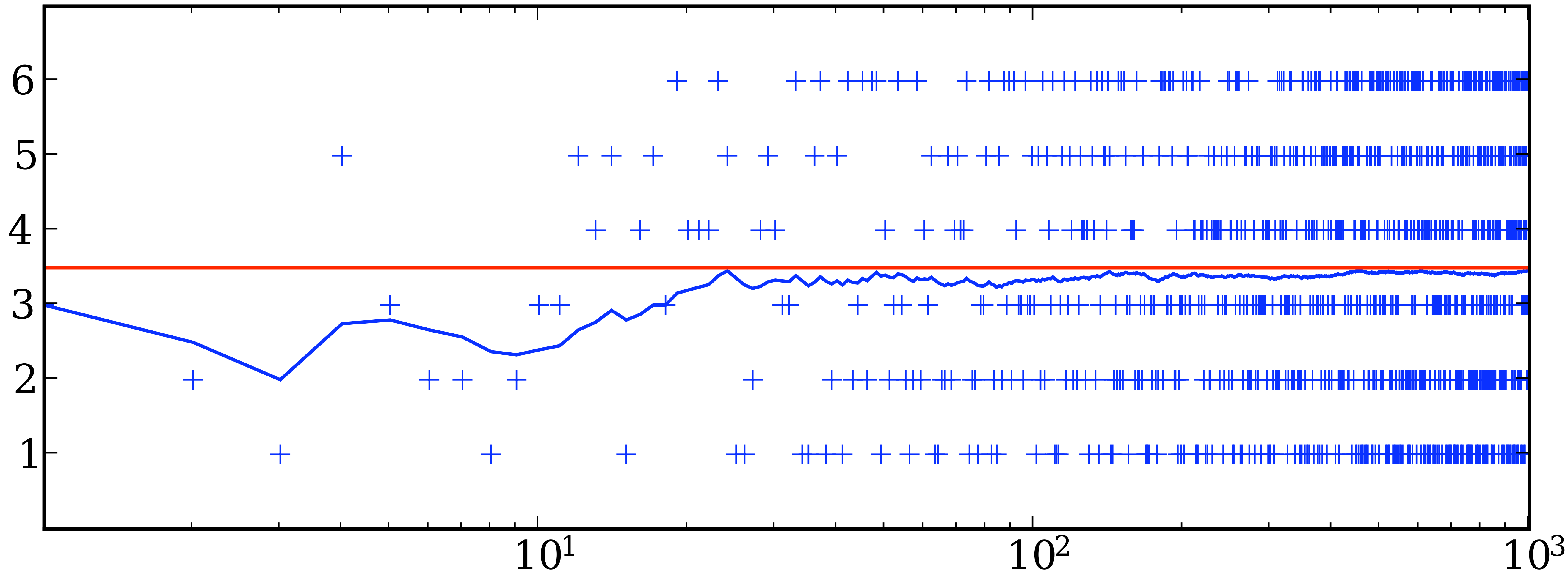
ad campaign working
new page layout better
drug working

- Probability of seeing '6' at most 11 times

$$\Pr(X \leq 11) = \sum_{i=0}^{11} p(i) = \sum_{i=0}^{11} \binom{100}{i} \left[\frac{1}{6}\right]^{i} \left[\frac{5}{6}\right]^{100-i} \approx 7.0\%$$

It's probably OK ... can we develop general theory?

Empirical average for a dice



how quickly does it converge?

Law of Large Numbers

- Random variables x_i with mean $\mu = \mathbf{E}[x_i]$
- Empirical average $\hat{\mu}_n := \frac{1}{n} \sum_{i=1}^{n} x_i$

- Weak Law of Large Numbers

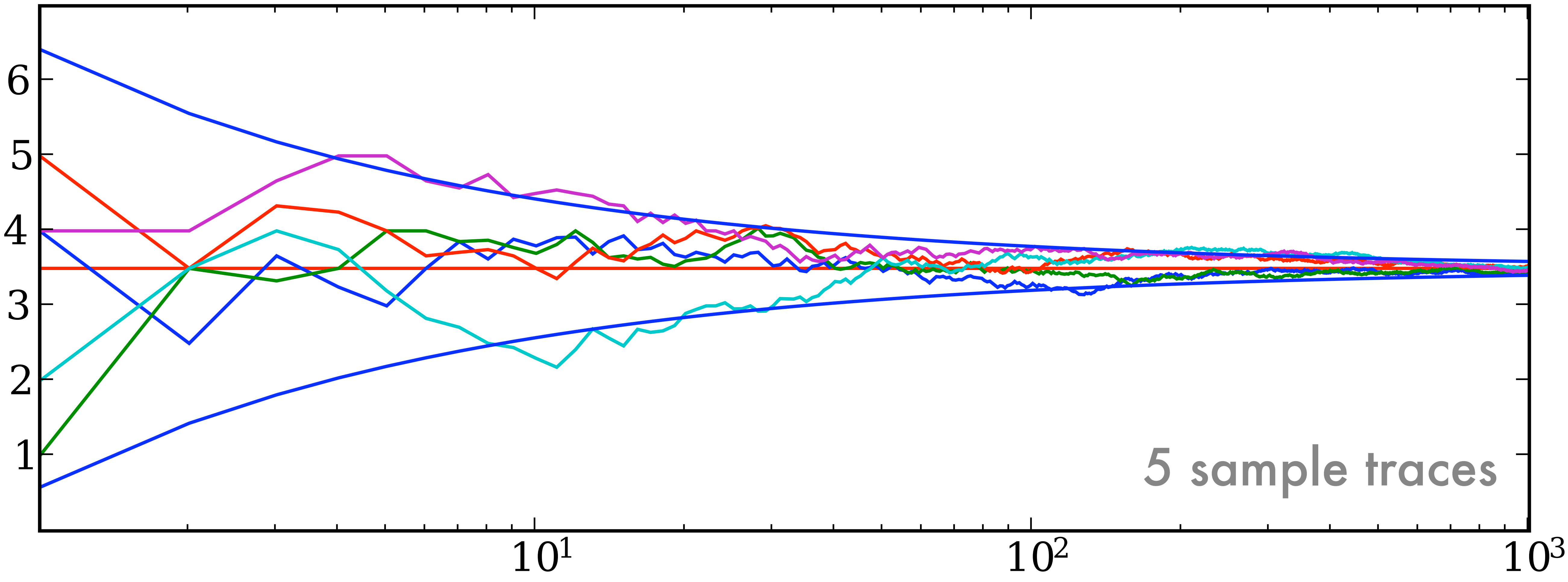
$$\lim_{n \to \infty} \Pr\left(|\hat{\mu}_n - \mu| \leq \epsilon\right) = 1 \text{ for any } \epsilon > 0$$

- Strong Law of Large Numbers

$$\Pr\left(\lim_{n \to \infty} \hat{\mu}_n = \mu\right) = 1$$

this means convergence in probability

Empirical average for a dice



- Upper and lower bounds are $\mu \pm \sqrt{\mathrm{Var}(x)/n}$
- This is an example of the central limit theorem

Central Limit Theorem

- Independent random variables x_i with mean μ_i and standard deviation σ_i
- The random variable
$$z_n := \left[\sum_{i=1}^n \sigma_i^2\right]^{-\frac{1}{2}} \left[\sum_{i=1}^n x_i - \mu_i\right]$$
converges to a Normal Distribution $\mathcal{N}(0,1)$
- Special case - IID random variables & average

$$\frac{\sqrt{n}}{\sigma}\left[\frac{1}{n}\sum_{i=1}^n x_i - \mu\right] \rightarrow \mathcal{N}(0,1)$$

$O\left(n^{-\frac{1}{2}}\right)$ convergence

Slutsky's Theorem

- Continuous mapping theorem
  - X_i and Y_i sequences of random variables
  - X_i has as its limit the random variable X
  - Y_i has as its limit the constant c
  - g(x,y) is continuous function for all g(x,c)

- $g(X_i, Y_i)$ converges in distribution to g(X,c)

Delta Method

- Random variable X_i convergent to b
$$a_n^{-2}(X_n - b) \to \mathcal{N}(0, \Sigma) \text{ with } a_n^2 \to 0 \text{ for } n \to \infty$$
- g is a continuously differentiable function for b
- Then $g(X_i)$ inherits convergence properties
$$a_n^{-2}\left(g(X_n) - g(b)\right) \to \mathcal{N}(0, [\nabla_x g(b)]\Sigma[\nabla_x g(b)]^\top)$$
- Proof: use Taylor expansion for $g(X_n)$ - g(b)
$$a_n^{-2}\left[g(X_n) - g(b)\right] = [\nabla_x g(\xi_n)]^\top a_n^{-2}(X_n - b)$$
  - $g(\xi_n)$ is on line segment $[X_n$, b]
  - By Slutsky's theorem it converges to g(b)
  - Hence $g(X_i)$ is asymptotically normal


since THE 1988
ONE HUNDRED PROOF
WHISKEY
57 PERCENT ALC/VOL
750 ML

Fourier Transform

- Fourier transform relations

$$F[f](\omega) := (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^n} f(x) \exp(-i \langle \omega, x \rangle) dx$$

$$F^{-1}[g](x) := (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^n} g(\omega) \exp(i \langle \omega, x \rangle) d\omega.$$

- Useful identities
  - Identity

$$F^{-1} \circ F = F \circ F^{-1} = \mathrm{Id}$$

  - Derivative

$$F[\partial_x f] = -i\omega F[f]$$

  - Convolution (also holds for inverse transform)

$$F[f \circ g] = (2\pi)^{\frac{d}{2}} F[f] \cdot F[g]$$

The Characteristic Function Method

- Characteristic function

$$\phi_X(\omega) := F^{-1}[p(x)] = \int \exp(i \langle \omega, x \rangle) dp(x)$$

- For X and Y independent we have
  - Joint distribution is convolution

$$p_{X+Y}(z) = \int p_X(z-y) p_Y(y) dy = p_X \circ p_Y$$

  - Characteristic function is product

$$\phi_{X+Y}(\omega) = \phi_X(\omega) \cdot \phi_Y(\omega)$$

  - Proof - plug in definition of Fourier transform
- Characteristic function is unique

Proof - Weak law of large numbers

- Require that expectation exists
- Taylor expansion of exponential

$$\exp(iwx) = 1 + i\langle w, x\rangle + o(|w|)$$
$$\text{and hence } \phi_X(\omega) = 1 + iw\mathbf{E}_X[x] + o(|w|).$$

(need to assume that we can bound the tail)

- Average of random variables

$$\phi_{\hat{\mu}_m}(\omega) = \left(1 + \frac{i}{m}w\mu + o(m^{-1}|w|)\right)^m$$

convolution

vanishing higher order terms

- Limit is constant distribution

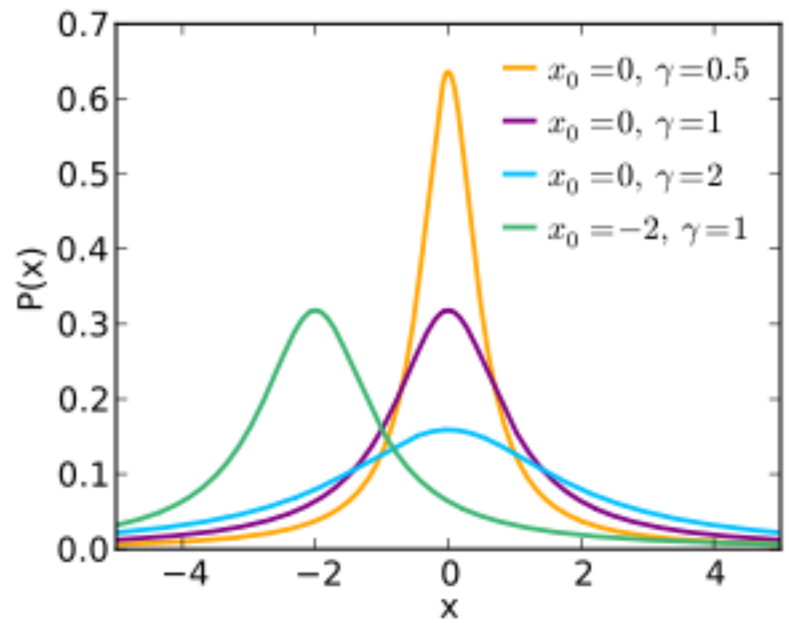
$$\phi_{\hat{\mu}_m}(\omega) \to \exp i\omega\mu = 1 + i\omega\mu + \ldots$$

mean

Warning

- Moments may not always exist
  - Cauchy distribution

$$p(x) = \frac{1}{\pi}\frac{1}{1+x^2}$$



  - For the mean to exist the following integral would have to converge

$$\int |x| dp(x) \geq \frac{2}{\pi}\int_1^\infty \frac{x}{1+x^2}dx \geq \frac{1}{\pi}\int_1^\infty \frac{1}{x}dx = \infty$$

Proof - Central limit theorem

- Require that second order moment exists (we assume they're all identical WLOG)
- Characteristic function

$$\exp(iwx) = 1 + iwx - \frac{1}{2}w^2x^2 + o(|w|^2)$$

$$\text{and hence } \phi_X(\omega) = 1 + iw\mathbf{E}_X[x] - \frac{1}{2}w^2\mathrm{var}_X[x] + o(|w|^2)$$

- Subtract out mean (centering) $z_n := \left[\sum_{i=1}^n \sigma_i^2\right]^{-\frac{1}{2}} \left[\sum_{i=1}^n x_i - \mu_i\right]$

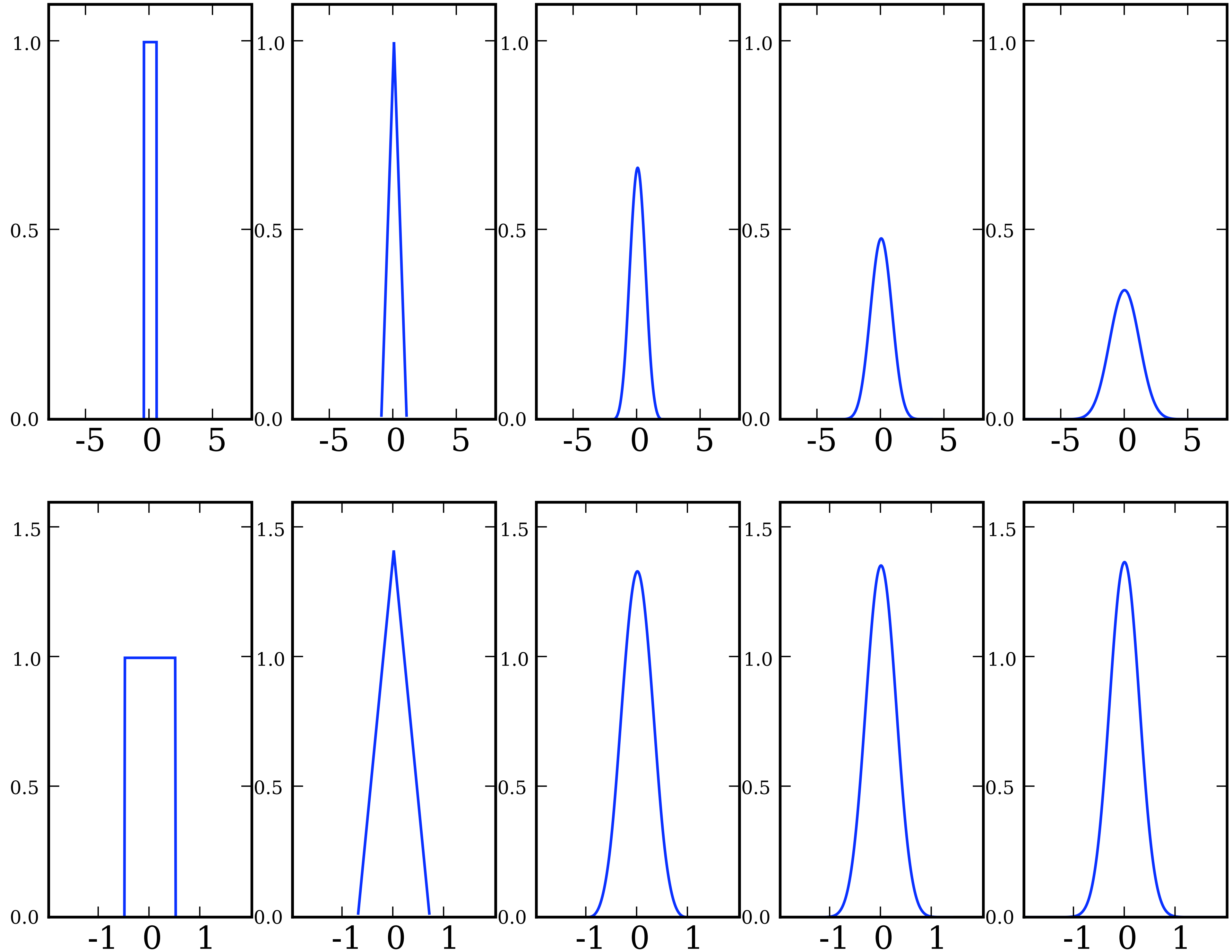
$$\phi_{Z_m}(\omega) = \left(1 - \frac{1}{2m}w^2 + o(m^{-1}|w|^2)\right)^m \to \exp\left(-\frac{1}{2}w^2\right) \text{ for } m \to \infty$$

This is the FT of a Normal Distribution

Central Limit Theorem in Practice

unscaled

scaled



Finite sample tail bounds

Simple tail bounds

- Gauss Markov inequality

Random variable X with mean μ

$$\Pr(X \geq \epsilon) \leq \mu/\epsilon$$

Proof - decompose expectation

$$\Pr(X \geq \epsilon) = \int_\epsilon^\infty dp(x) \leq \int_\epsilon^\infty \frac{x}{\epsilon} dp(x) \leq \epsilon^{-1} \int_0^\infty x dp(x) = \frac{\mu}{\epsilon}.$$

- Chebyshev inequality

Random variable X with mean μ and variance σ^2

$$\Pr(|\hat{\mu}_m - \mu\| > \epsilon) \leq \sigma^2 m^{-1} \epsilon^{-2} \text{ or equivalently } \epsilon \leq \sigma/\sqrt{m\delta}$$

Proof - applying Gauss-Markov to $Y = (X - \mu)^2$ with confidence ε^2 yields the result.

Scaling behavior

- Gauss-Markov
  $$\epsilon \leq \frac{\mu}{\delta}$$
  Scales properly in μ but expensive in δ
- Chebyshev
  $$\epsilon \leq \frac{\sigma}{\sqrt{m\delta}}$$
  Proper scaling in σ but still bad in δ

  Can we get logarithmic scaling in δ?

Chernoff bound

- KL-divergence variant of Chernoff bound

$$K(p,q) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$$

- n independent tosses from biased coin with p

$$\Pr\left\{\sum_i x_i \geq nq\right\} \leq \exp\left(-nK(q,p)\right) \leq \exp\left(-2n(p-q)^2\right)$$

Pinsker's inequality

- Proof

$$\text{w.l.o.g.} q > p \text{ and set } k \geq qn$$

$$\frac{\Pr\left\{\sum_i x_i = k|q\right\}}{\Pr\left\{\sum_i x_i = k|p\right\}} = \frac{q^k(1-q)^{n-k}}{p^k(1-p)^{n-k}} \geq \frac{q^{qn}(1-q)^{n-qn}}{p^{qn}(1-p)^{n-qn}} = \exp\left(nK(q,p)\right)$$

$$\sum_{k \geq nq} \Pr\left\{\sum_i x_i = k|p\right\} \leq \sum_{k \geq nq} \Pr\left\{\sum_i x_i = k|q\right\} \exp(-nK(q,p)) \leq \exp(-nK(q,p))$$

Carnegie Mellon University

McDiarmid Inequality

- Independent random variables X_i
- Function $f : \mathcal{X}^m \to \mathbb{R}$
- Deviation from expected value

$$\Pr\left(\left|f(x_1, \ldots, x_m) - \mathbf{E}_{X_1,\ldots,X_m}[f(x_1, \ldots, x_m)]\right| > \epsilon\right) \leq 2\exp\left(-2\epsilon^2 C^{-2}\right)$$

Here C is given by $C^2 = \sum_{i=1}^{m} c_i^2$ where

$$|f(x_1, \ldots, x_i, \ldots, x_m) - f(x_1, \ldots, x_i', \ldots, x_m)| \leq c_i$$

- Hoeffding's theorem
  f is average and X_i have bounded range c

$$\Pr\left(|\hat{\mu}_m - \mu| > \epsilon\right) \leq 2\exp\left(-\frac{2m\epsilon^2}{c^2}\right).$$

Carnegie Mellon University

Scaling behavior

- Hoeffding

$$\delta := \Pr\left(|\hat{\mu}_m - \mu| > \epsilon\right) \leq 2 \exp\left(-\frac{2m\epsilon^2}{c^2}\right)$$

$$\implies \log \delta/2 \leq -\frac{2m\epsilon^2}{c^2}$$

$$\implies \epsilon \leq c\sqrt{\frac{\log 2 - \log \delta}{2m}}$$

This helps when we need to combine several tail bounds since we only pay logarithmically in terms of their combination.

More tail bounds

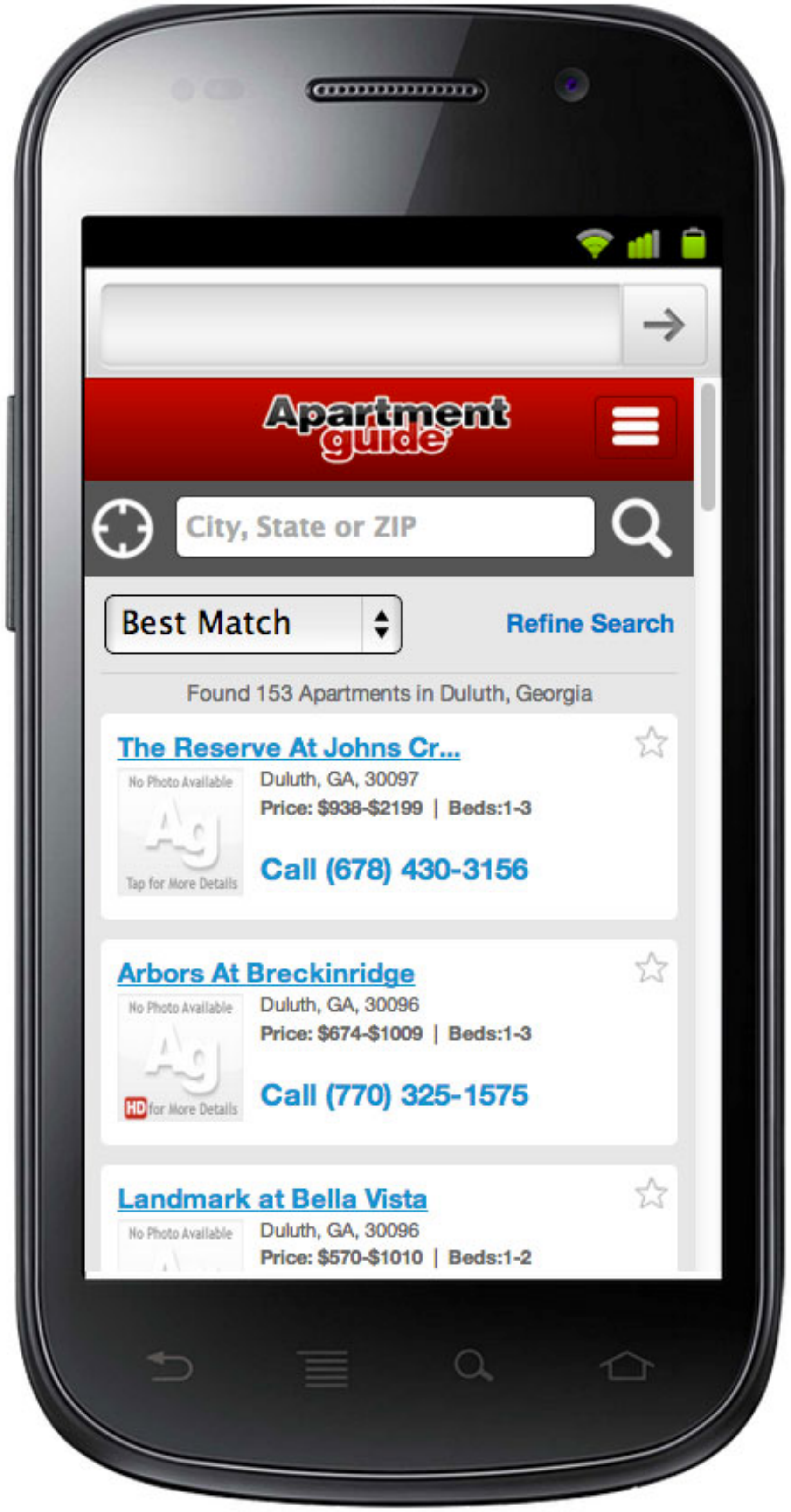
- Higher order moments
  - Bernstein inequality (needs variance bound)

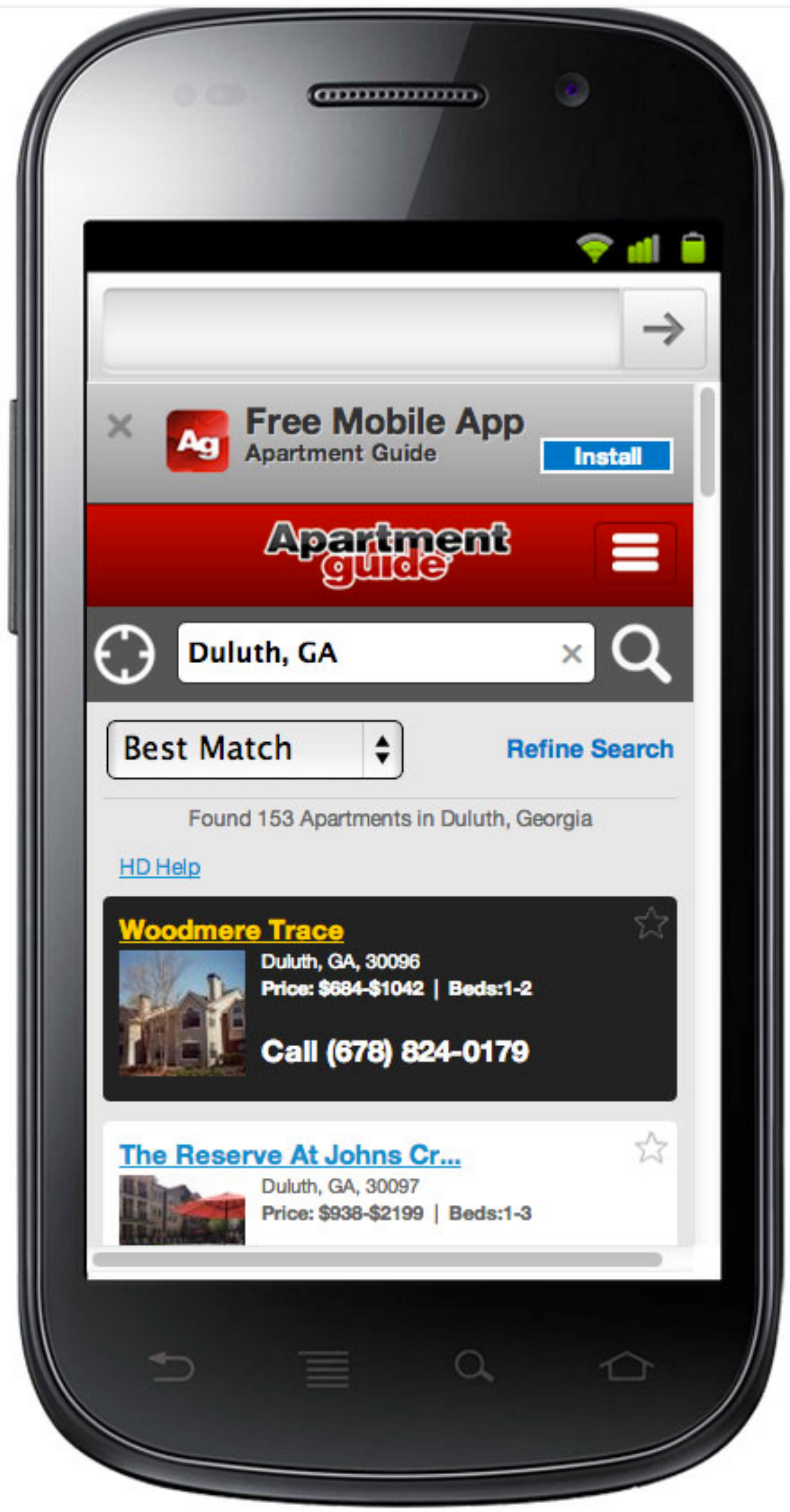
$$\Pr\left(\mu_m - \mu \geq \epsilon\right) \leq \exp\left(-\frac{t^2/2}{\sum_i \mathbf{E}[X_i^2] + Mt/3}\right)$$

  here M upper-bounds the random variables X_i

  - Proof via Gauss-Markov inequality applied to exponential sums (hence exp. inequality)
  - See also Azuma, Bennett, Chernoff, ...
- Absolute / relative error bounds
- Bounds for (weakly) dependent random variables

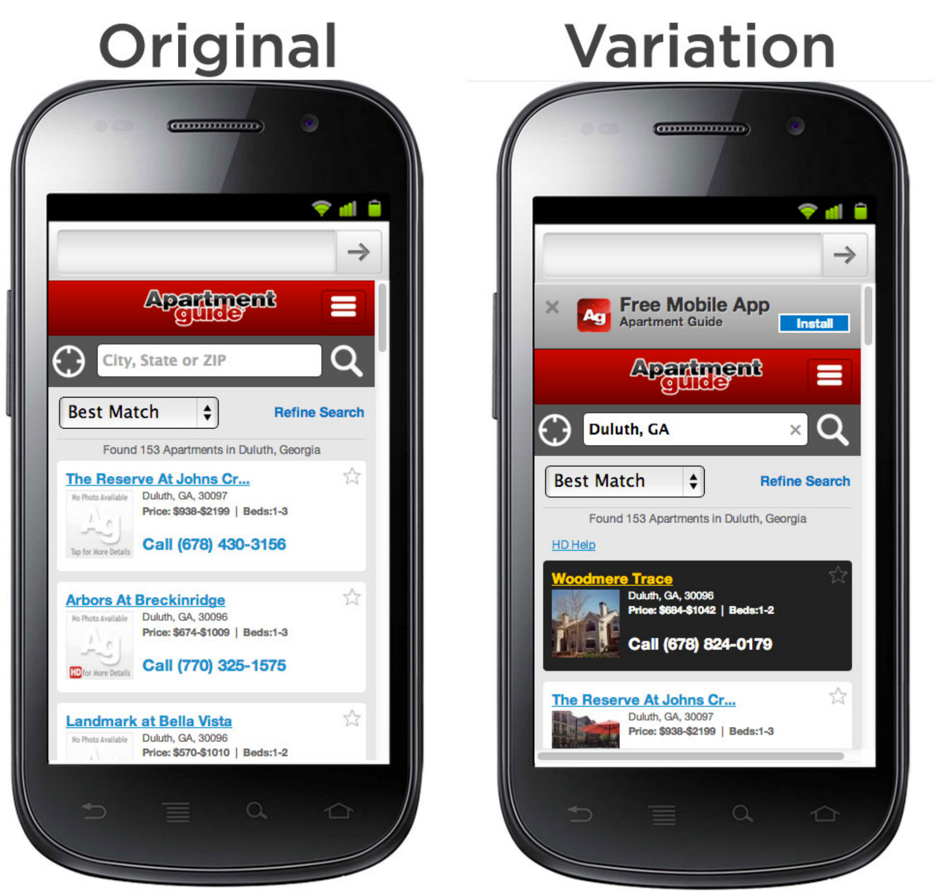
Tail bounds in practice





A/B testing

- Two possible webpage layouts
- Which layout is better?



- Experiment
  - Half of the users see A
  - The other half sees design B

- How many trials do we need to decide which is better

Assume that the probabilities are p(A) = 0.1 and p(B) = 0.11 respectively and that p(A) is known

Chebyshev Inequality

- Need to bound for a deviation of 0.01
- Mean is p(B) = 0.11 (we don't know this yet)
- Want failure probability of 5%

- If we have no prior knowledge, we can only bound the variance by $\sigma^2 = 0.25$

$$m \leq \frac{\sigma^2}{\epsilon^2 \delta} = \frac{0.25}{0.01^2 \cdot 0.05} = 50,000$$

- If we know that the click probability is at most 0.15 we can bound the variance at 0.15 * 0.85 = 0.1275. This requires at most 25,500 users.

Hoeffding's bound

- Random variable has bounded range [0, 1] (click or no click), hence c=1
- Solve Hoeffding's inequality for m

$$m \leq -\frac{c^2 \log \delta/2}{2\epsilon^2} = -\frac{1 \cdot \log 0.025}{2 \cdot 0.01^2} < 18,445$$

This is slightly better than Chebyshev.

Normal Approximation (Central Limit Theorem)

- Use asymptotic normality
- Gaussian interval containing 0.95 probability

$$\frac{1}{2\pi\sigma^2}\int_{\mu-\epsilon}^{\mu+\epsilon}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)dx = 0.95$$

  is given by ε = 2.96σ.
- Use variance bound of 0.1275 (see Chebyshev)

$$m \leq \frac{2.96^2\sigma^2}{\epsilon^2} = \frac{2.96^2 \cdot 0.1275}{0.01^2} \leq 11,172$$

  Same rate as Hoeffding bound!
  Better bounds by bounding the variance.

Beyond

- Many different layouts?
- Combinatorial strategy to generate them (aka the Thai Restaurant process)
- What if it depends on the user / time of day
- Stateful user (e.g. query keywords in search)
- What if we have a good prior of the response (rather than variance bound)?

- Explore/exploit/reinforcement learning/control

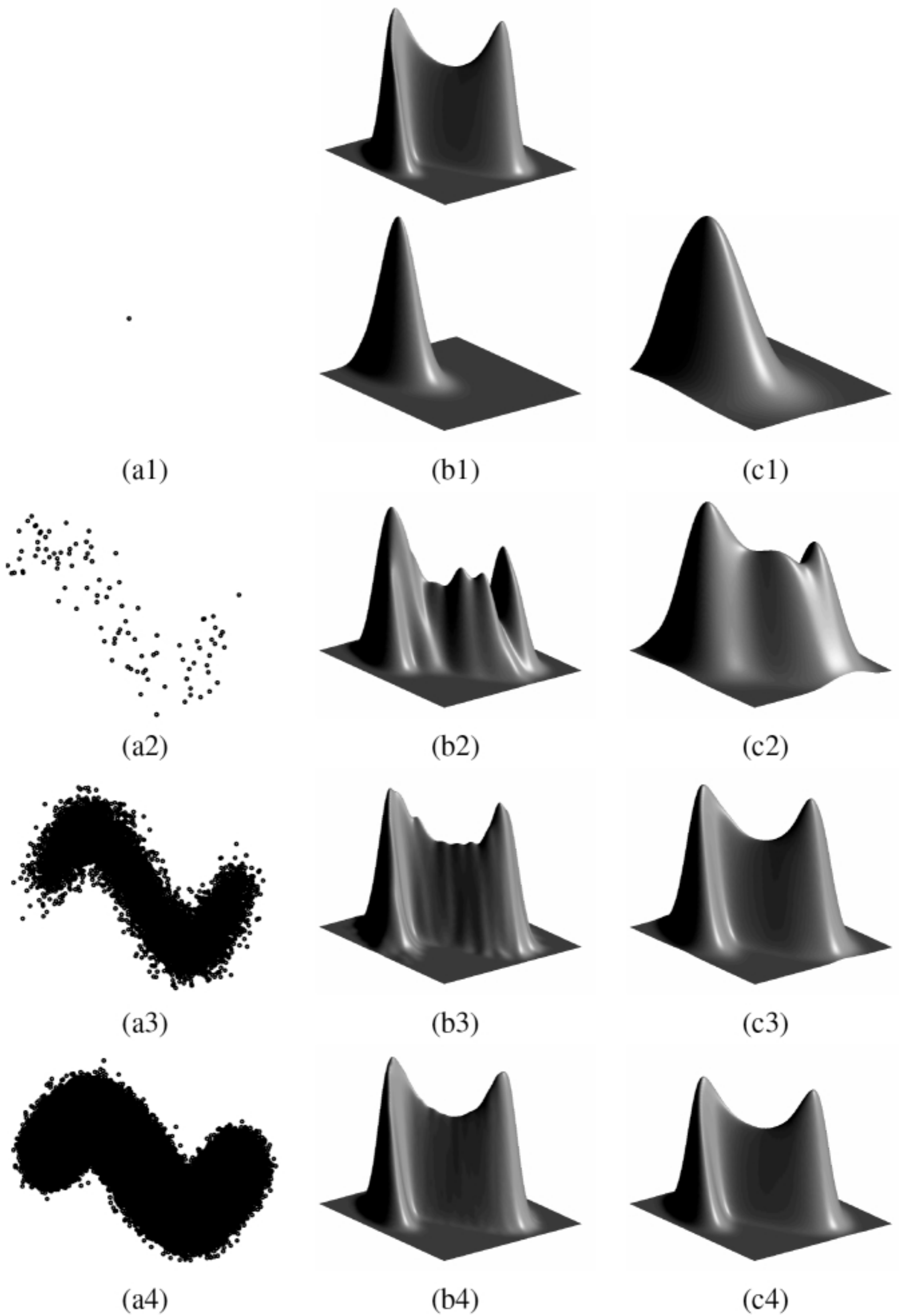
2.3 Kernel Density Estimation

2 Statistics

Alexander Smola

Introduction to Machine Learning 10-701

http://alex.smola.org/teaching/10-701-15



Parzen Windows

Density Estimation

- Observe some data x_i
- Want to estimate p(x)
  - Find unusual observations (e.g. security)
  - Find typical observations (e.g. prototypes)
  - Classifier via Bayes Rule

$$p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x|y)p(y)}{\sum_{y'} p(x|y')p(y')}$$

- Need tool for computing p(x) easily

Bin Counting

- Discrete random variables, e.g.
  - English, Chinese, German, French, ...
  - Male, Female
- Bin counting (record # of occurrences)

25	English	Chinese	German	French	Spanish
male	5	2	3	1	0
female	6	3	2	2	1

Bin Counting

- Discrete random variables, e.g.
  - English, Chinese, German, French, ...
  - Male, Female
- Bin counting (record # of occurrences)

25	English	Chinese	German	French	Spanish
male	0.2	0.08	0.12	0.04	0
female	0.24	0.12	0.08	0.08	0.04

Bin Counting

- Discrete random variables, e.g.
  - English, Chinese, German, French, ...
  - Male, Female
- Bin counting (record # of occurrences)

25	English	Chinese	German	French	Spanish
male	0.2	0.08	0.12	0.04	0
female	0.24	0.12	0.08	0.08	0.04

Bin Counting

- Discrete random variables, e.g.
  - English, Chinese, German, French, ...
  - Male, Female
- Bin counting (record # of occurrences)

not enough data

25	English	Chinese	German	French	Spanish
male	0.2	0.08	0.12	0.04	0
female	0.24	0.12	0.08	0.08	0.04

Curse of dimensionality (lite)

- Discrete random variables, e.g.
  - English, Chinese, German, French, ...
  - Male, Female
  - ZIP code
  - Day of the week
  - Operating system
  - ...

#bins grows exponentially

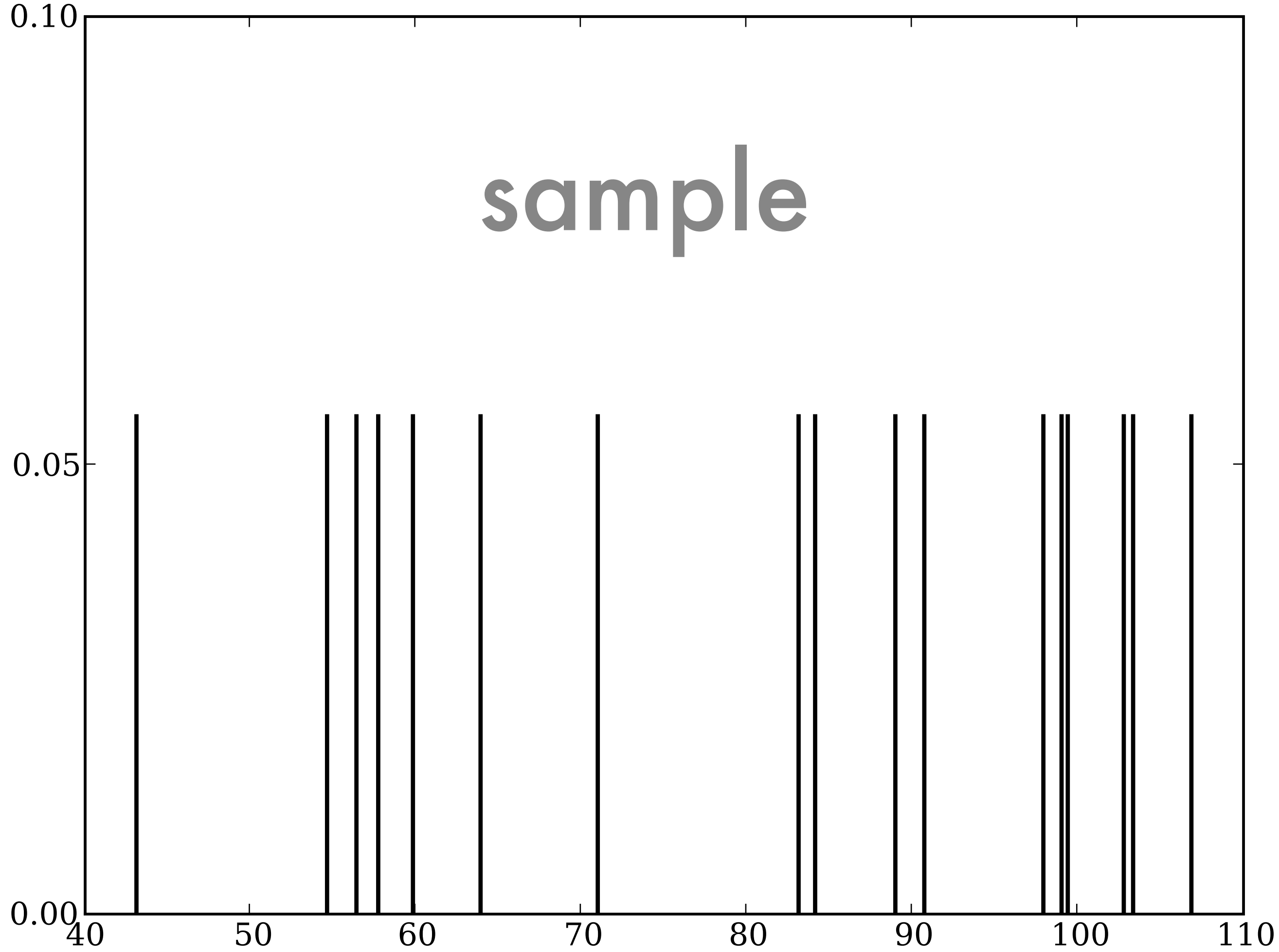
Curse of dimensionality (lite)

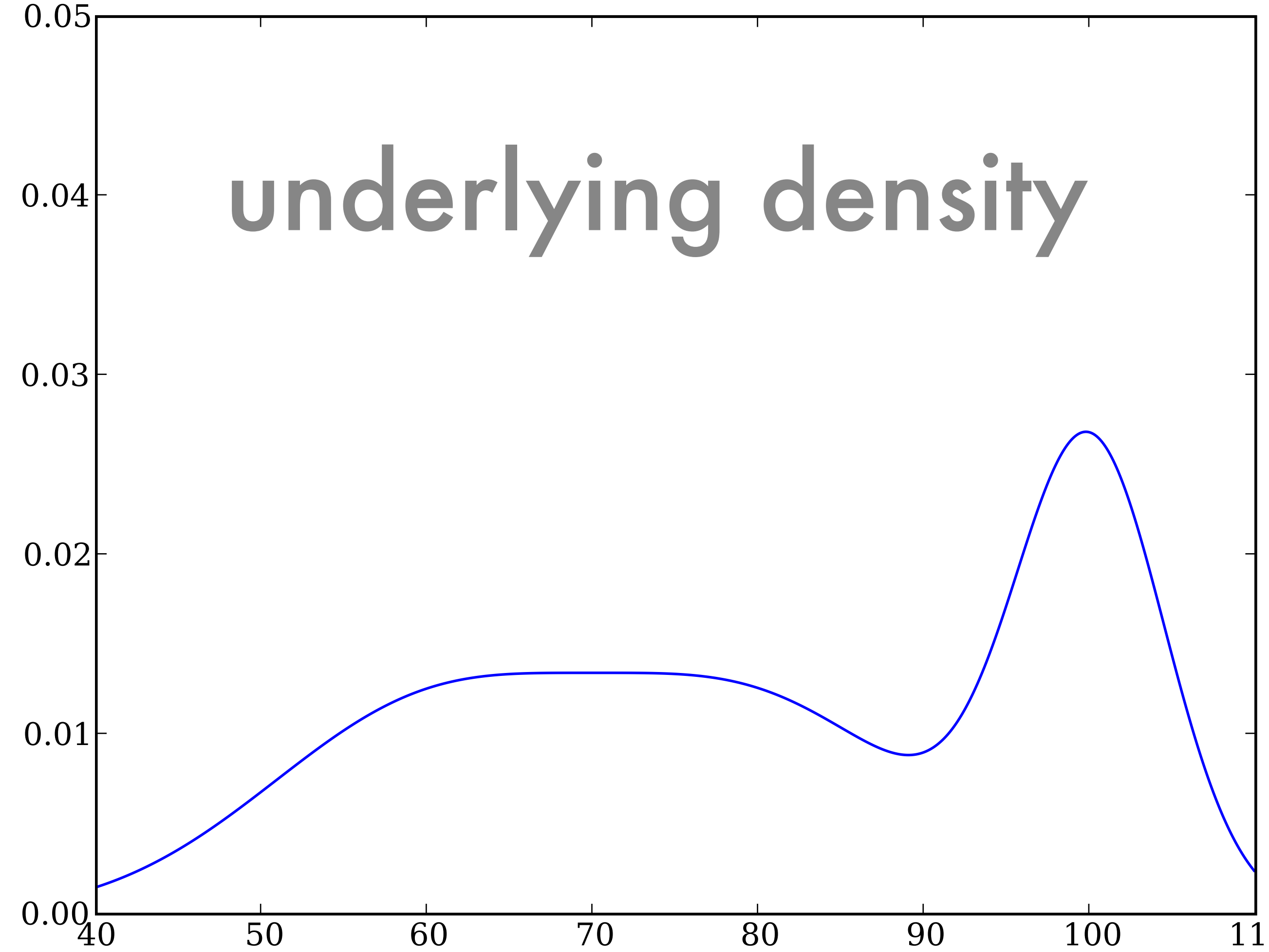
- Discrete random variables, e.g.
  - English, Chinese, German, French, ...
  - Male, Female
  - ZIP code
  - Day of the week
  - Operating system
  - ...
- Continuous random variables
  - Income
  - Bandwidth
  - Time

#bins grows exponentially

need many bins per dimension

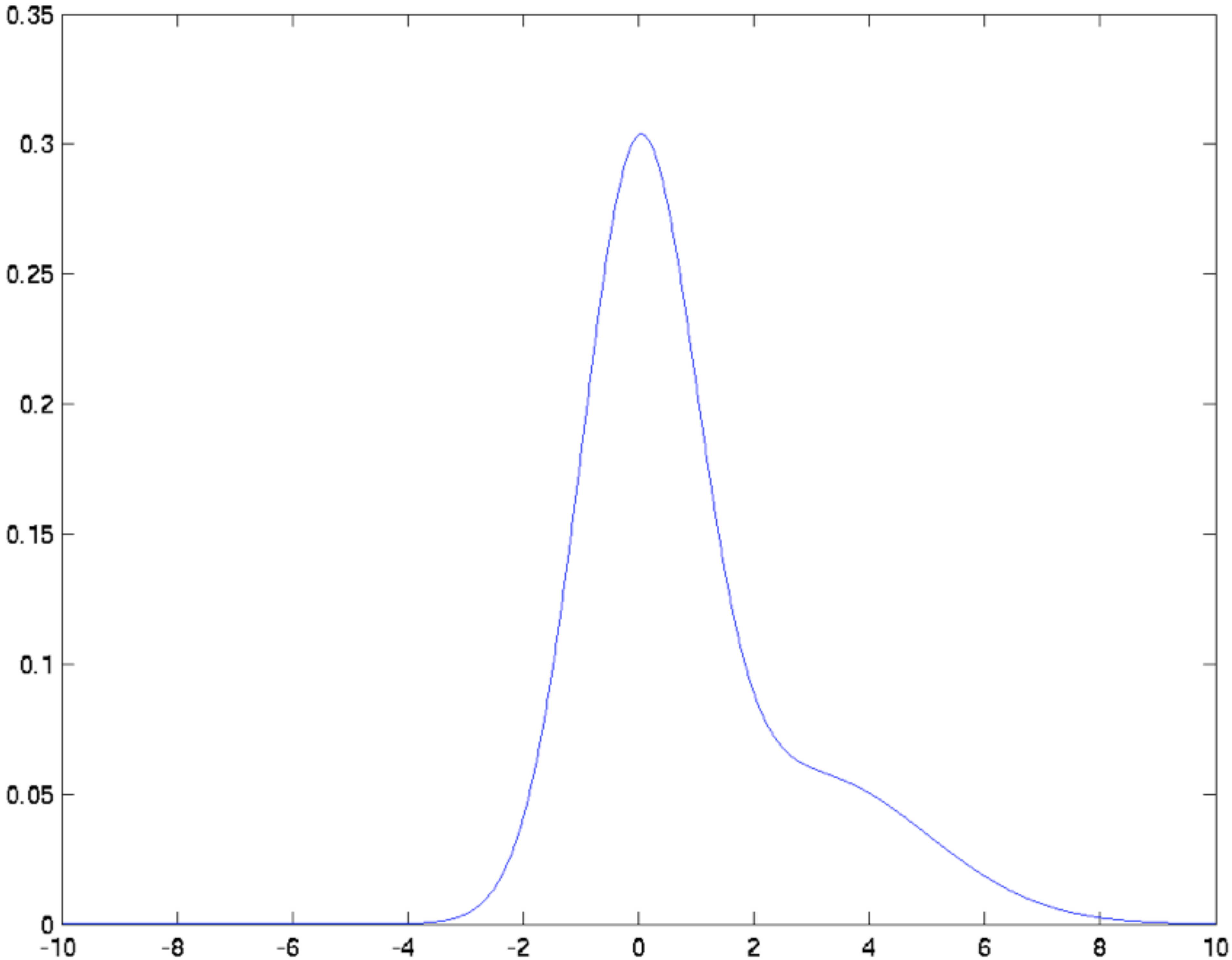
Density Estimation



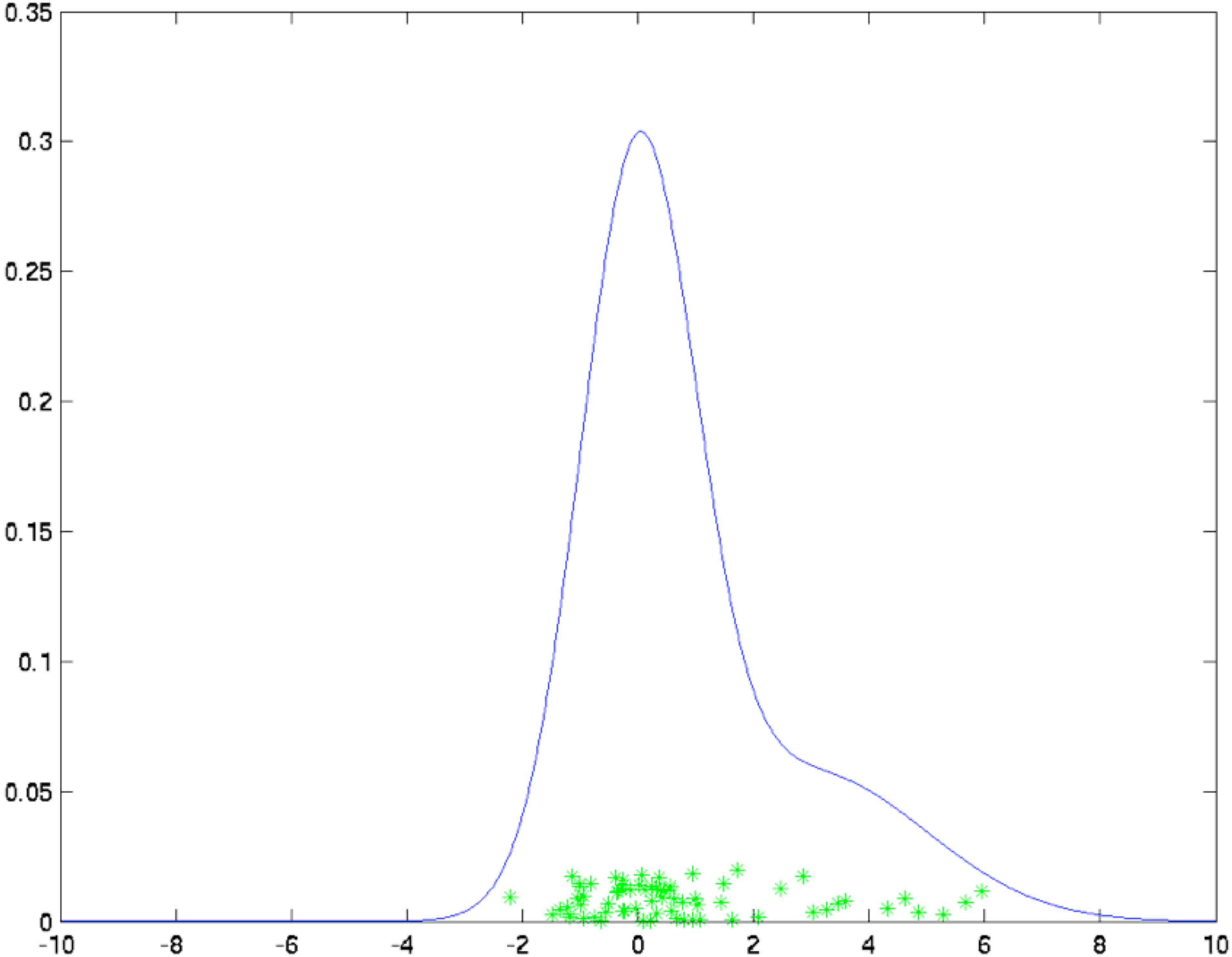


- Continuous domain = infinite number of bins
- Curse of dimensionality
  - 10 bins on [0, 1] is probably good
  - 10^{10} bins on $[0, 1]^{10}$ requires high accuracy in estimate: probability mass per cell also decreases by 10^{10}.

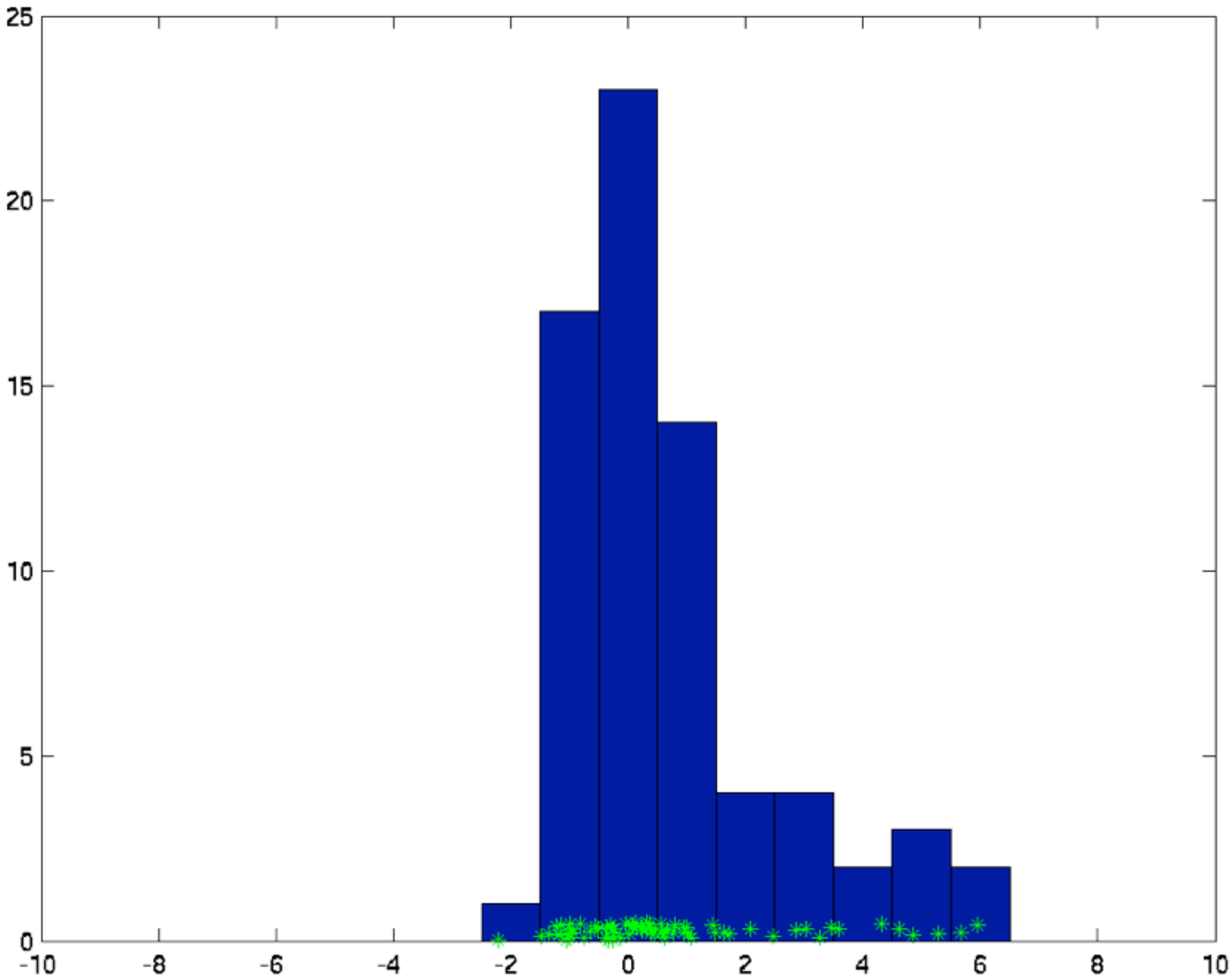
Bin Counting



Bin Counting

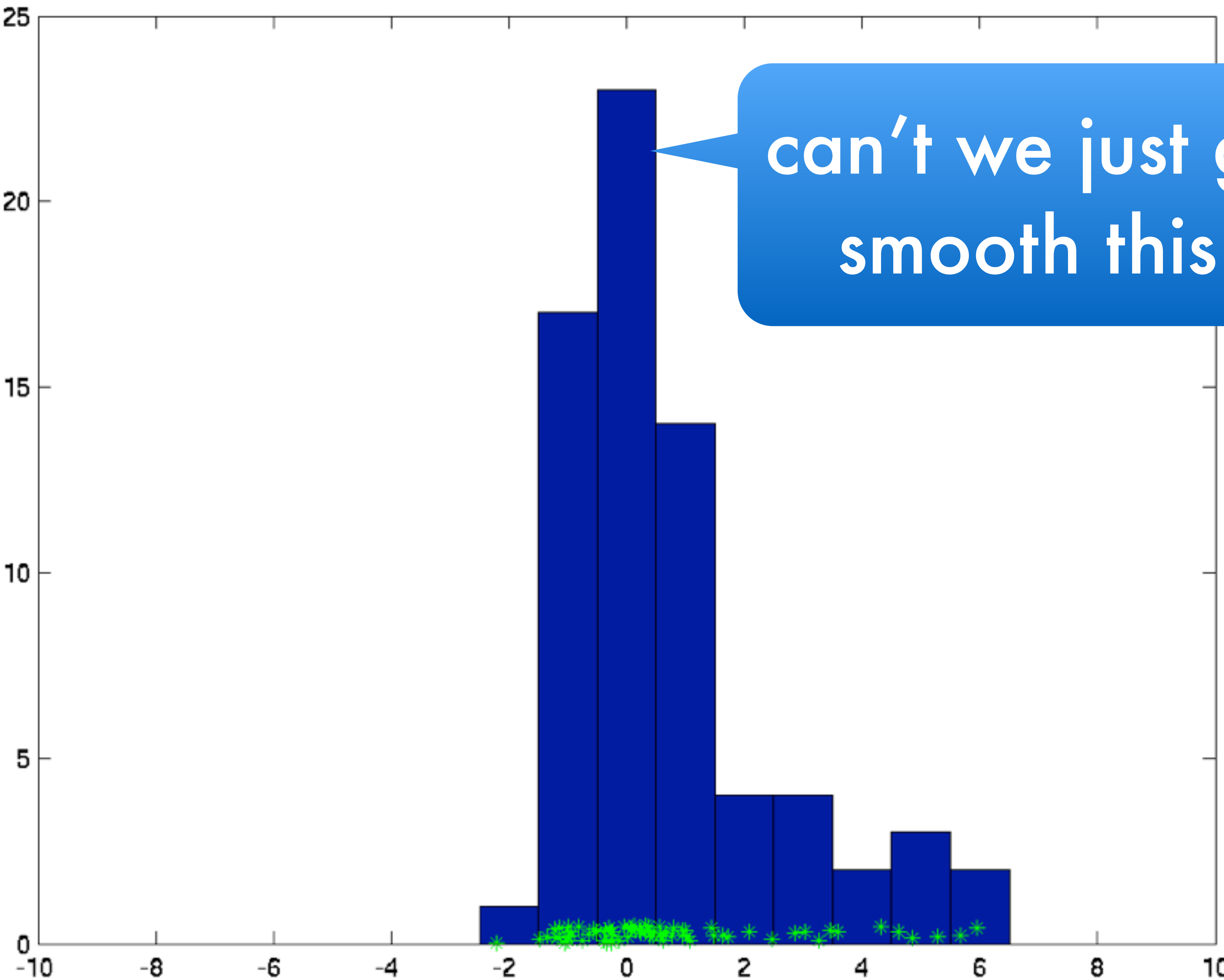


Bin Counting



iversity

Bin Counting



can't we just go and smooth this out?

What is happening?

- Hoeffding's theorem

$$\Pr\left\{\left|\mathbf{E}[x] - \frac{1}{m}\sum_{i=1}^{m} x_i\right| > \epsilon\right\} \leq 2e^{-2m\epsilon^2}$$

  For any average of [0,1] iid random variables.

- Bin counting
  - Random variables x_i are events in bins
  - Apply Hoeffding's theorem to each bin
  - Take the union bound over all bins to guarantee that all estimates converge

Density Estimation

- Hoeffding's theorem

$$\Pr\left\{\left|\mathbf{E}[x] - \frac{1}{m}\sum_{i=1}^{m} x_i\right| > \epsilon\right\} \leq 2e^{-2m\epsilon^2}$$

- Applying the union bound and Hoeffding

$$\Pr\left(\sup_{a\in A} |\hat{p}(a) - p(a)| \geq \epsilon\right) \leq \sum_{a\in A} \Pr\left(|\hat{p}(a) - p(a)| \geq \epsilon\right)$$

$$\leq 2|A| \exp\left(-2m\epsilon^2\right)$$

- Solving for error probability

good news

$$\frac{\delta}{2|A|} \leq \exp(-m\epsilon^2) \implies \epsilon \leq \sqrt{\frac{\boxed{\log 2|A|} - \log\delta}{2m}}$$

Density Estimation

- Hoeffding's theorem

$$\Pr\left\{\left|\mathbf{E}[x] - \frac{1}{m}\sum_{i=1}^{m} x_i\right| > \epsilon\right\} \leq 2e^{-2m\epsilon^2}$$

- Applying the union bound and Hoeffding

$$\Pr\left(\sup_{a \in A} |\hat{p}(a) - p(a)| \geq \epsilon\right) \leq \sum_{a \in A} \Pr\left(|\hat{p}(a) - p(a)| \geq \epsilon\right)$$

$$\leq 2|A| \exp\left(-2m\right.$$

- Solving for error probability

but not good enough

$$\frac{\delta}{2|A|} \leq \exp(-m\epsilon^2) \Longrightarrow \epsilon \leq \sqrt{\frac{\log 2|A| - \log \delta}{2m}}$$

Density Estimation

- Hoeffding's theorem

$$\Pr\left\{\left|\mathbf{E}[x] - \frac{1}{m}\sum_{i=1}^{m} x_i\right| > \epsilon\right\} \leq 2e^{-2m\epsilon^2}$$

bins not independent

- Applying the union bound and Ho

$$\Pr\left(\sup_{a \in A} |\hat{p}(a) - p(a)| \geq \epsilon\right) \leq \sum_{a \in A} \Pr\left(|\hat{p}(a) - p(a)| \geq \epsilon\right)$$

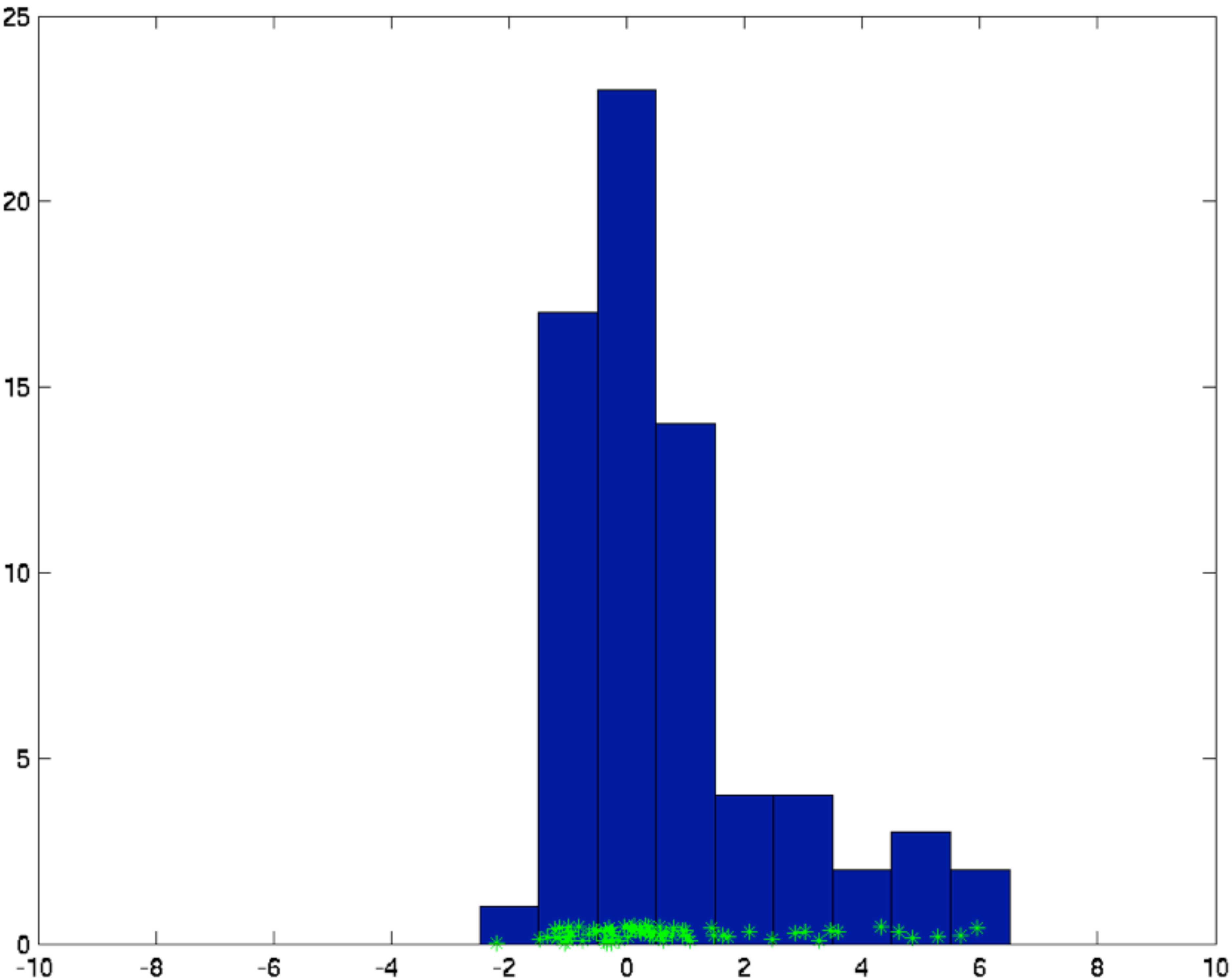
$$\leq 2|A| \exp\left(-2r\right.$$

but not good enough

- Solving for error probability

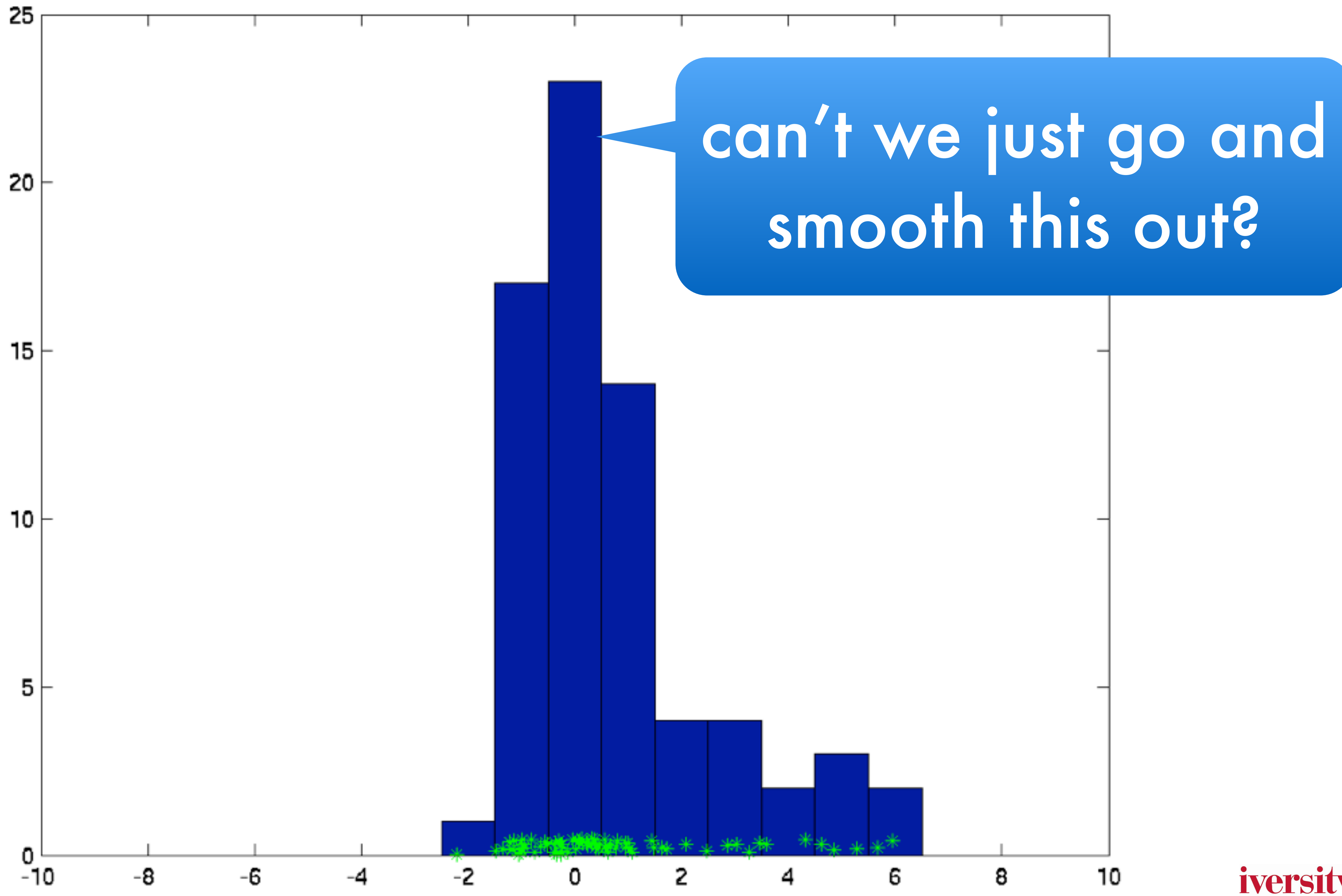
$$\frac{\delta}{2|A|} \leq \exp(-m\epsilon^2) \Longrightarrow \epsilon \leq \sqrt{\frac{\log 2|A| - \log \delta}{2m}}$$

Bin Counting



iversity

Bin Counting



Parzen Windows

- Naive approach
  Use empirical density (delta distributions)

$$p_{\mathrm{emp}}(x) = \frac{1}{m} \sum_{i=1}^{m} \delta_{x_i}(x)$$

- This breaks if we see slightly different instances
- Kernel density estimate
  Smear out empirical density with a nonnegative smoothing kernel $k_x(x')$ satisfying

$$\int_{\mathcal{X}} k_x(x') dx' = 1 \text{ for all } x$$

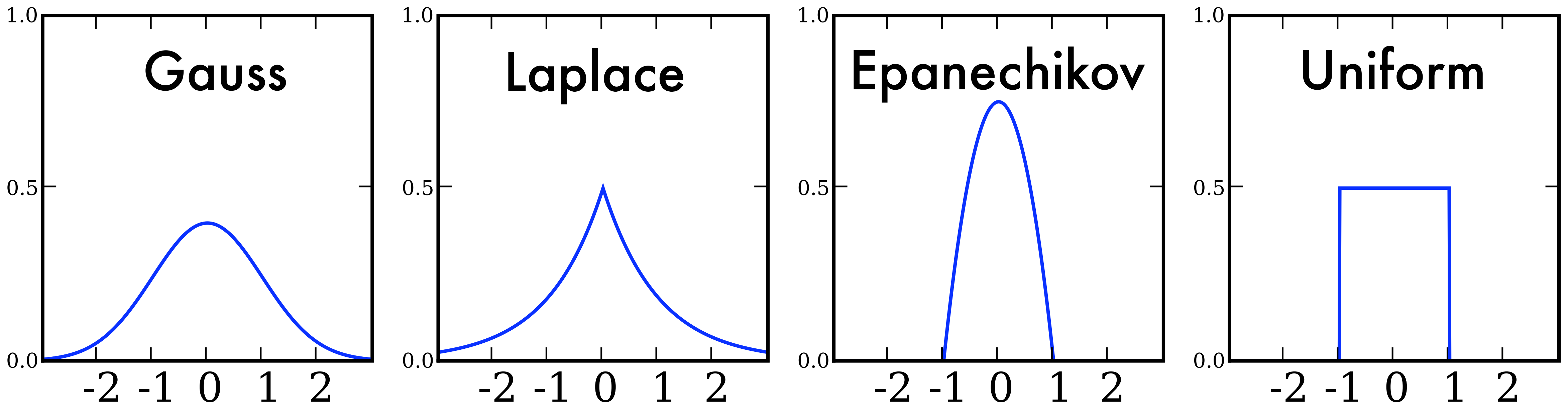
Parzen Windows

- Density estimate

$$p_{\text{emp}}(x) = \frac{1}{m}\sum_{i=1}^{m} \delta_{x_i}(x)$$

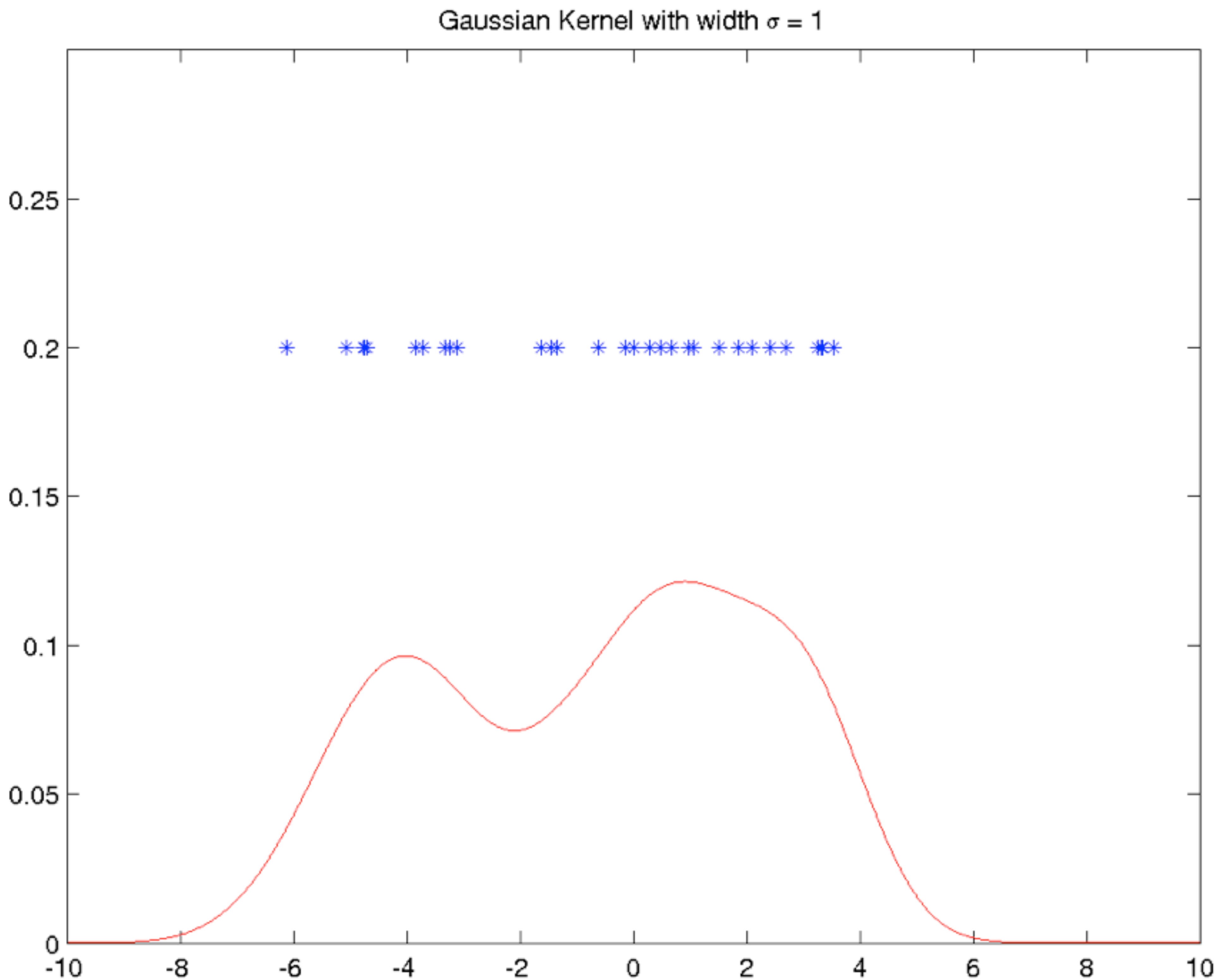
- Smoothing kernels

$$\hat{p}(x) = \frac{1}{m}\sum_{i=1}^{m} k_{x_i}(x)$$



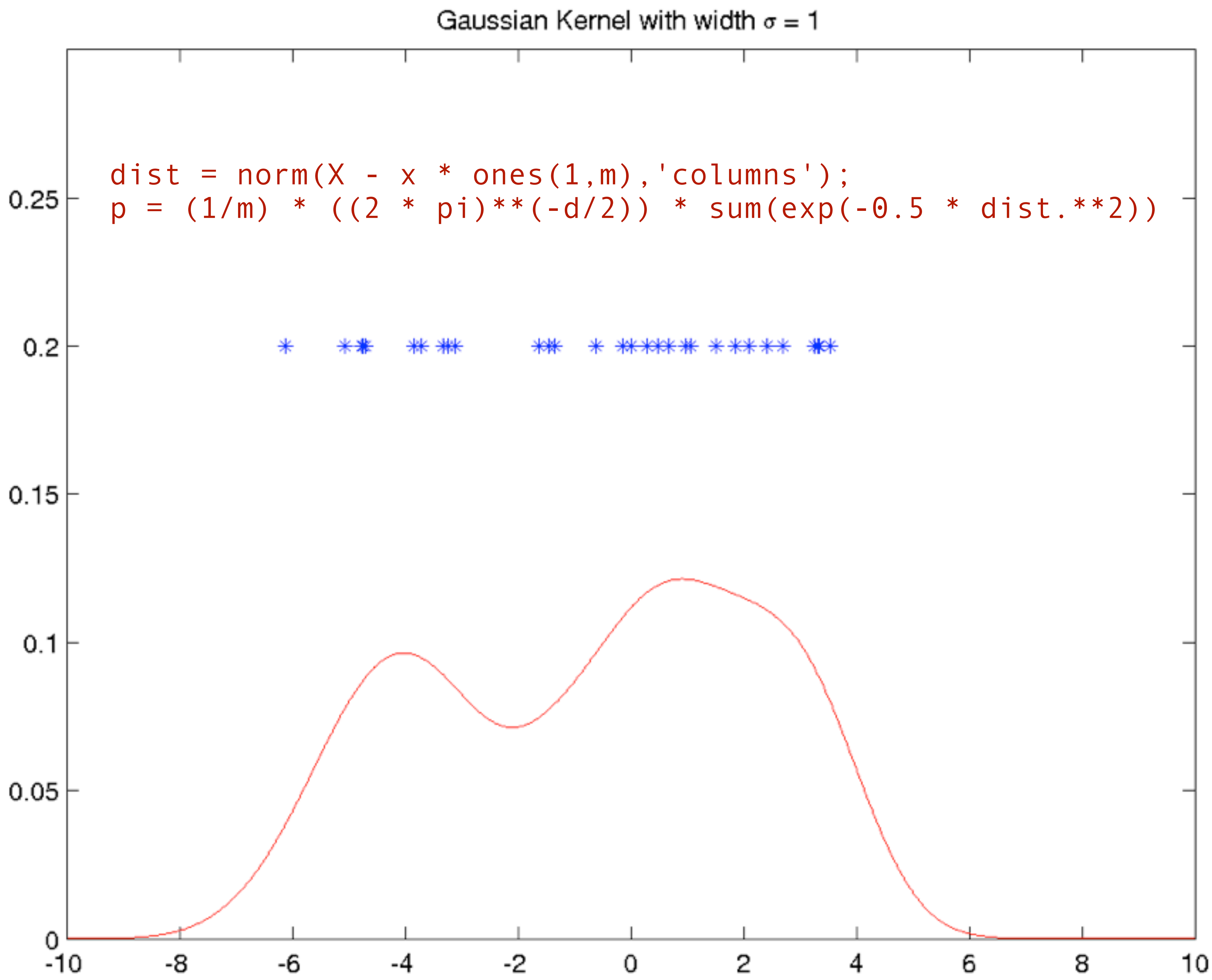
$$(2\pi)^{-\frac{1}{2}} e^{-\frac{1}{2}x^2} \qquad \frac{1}{2}e^{-|x|} \qquad \frac{3}{4}\max(0, 1 - x^2) \qquad \frac{1}{2}\chi_{[-1,1]}(x)$$

Smoothing

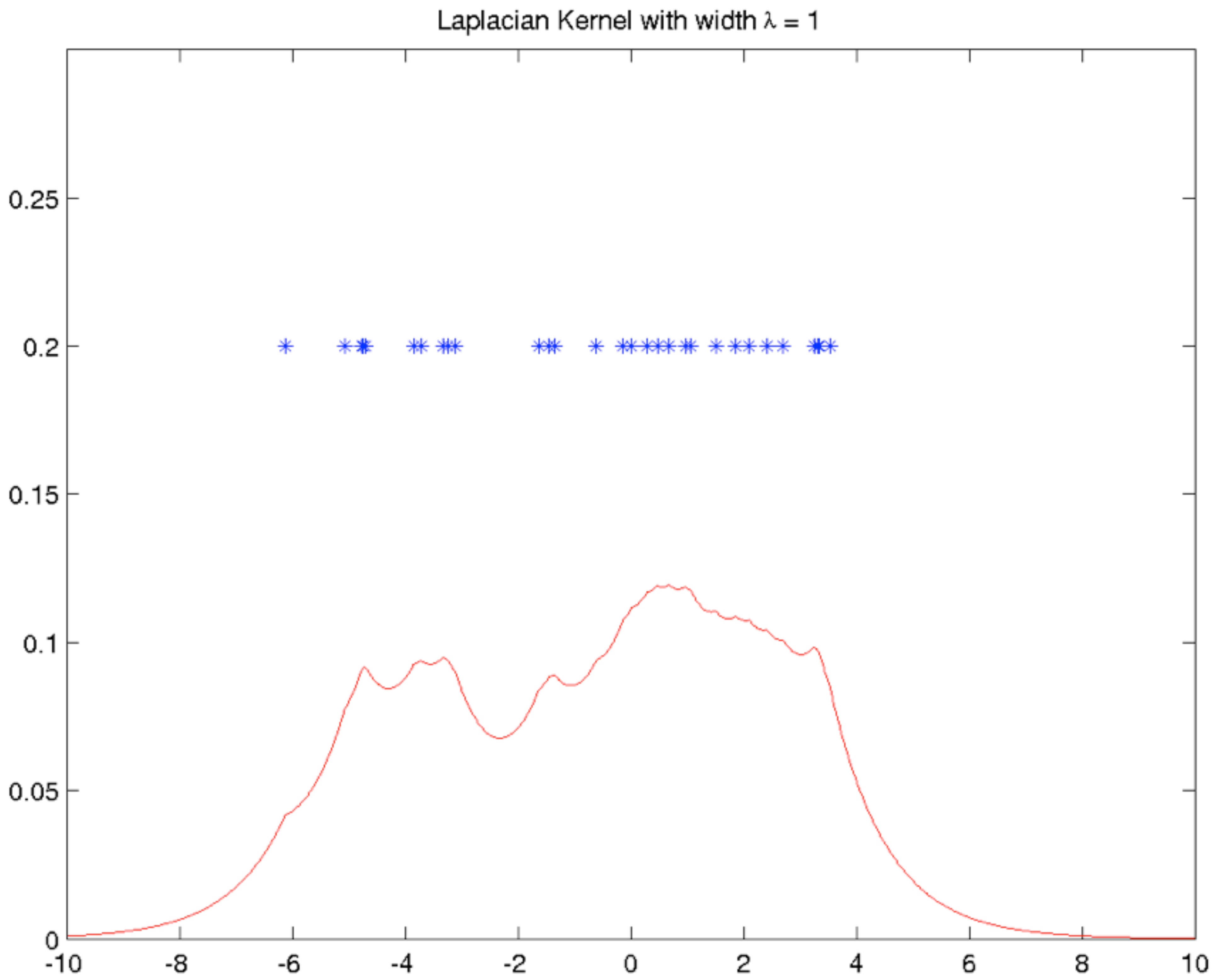


iversity

Smoothing

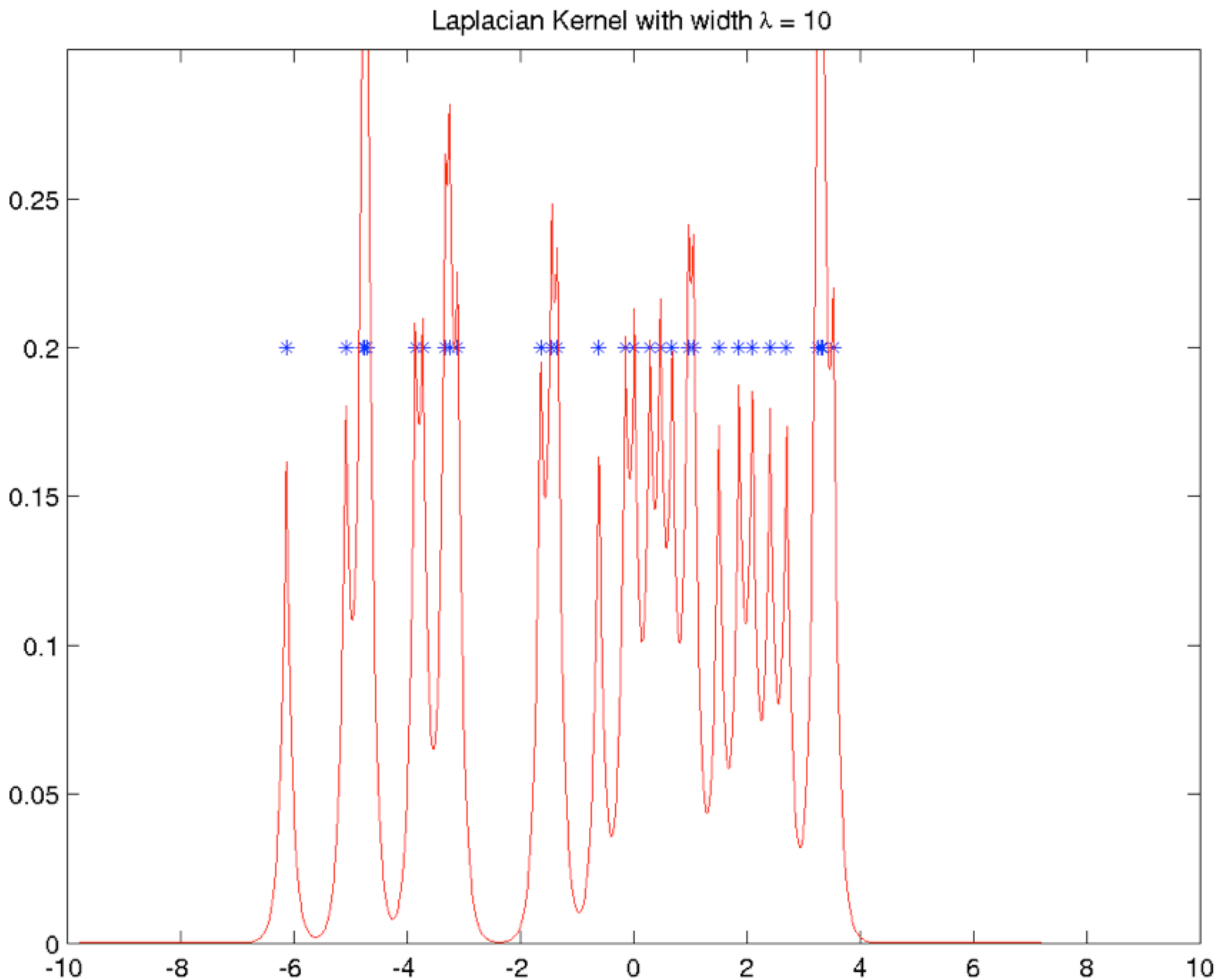

Gaussian Kernel with width σ = 1
dist = norm(X - x * ones(1,m),'columns');
p = (1/m) * ((2 * pi)**(-d/2)) * sum(exp(-0.5 * dist.**2))
0.25
0.2
0.15
0.1
0.05
0
-10
-8
-6
-4
-2
0
2
4
6
8
10

Smoothing

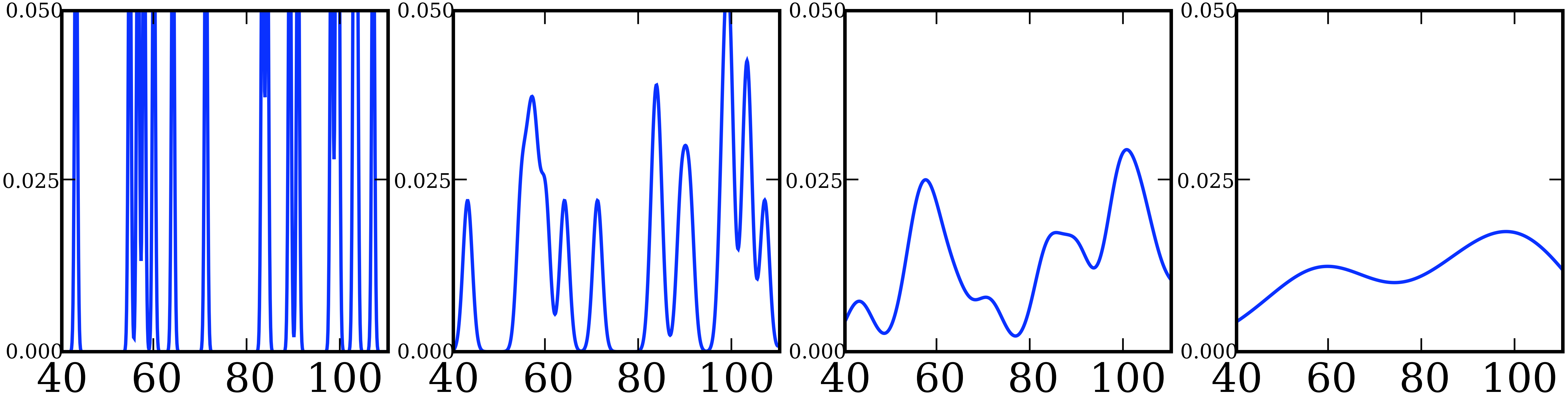


iversity

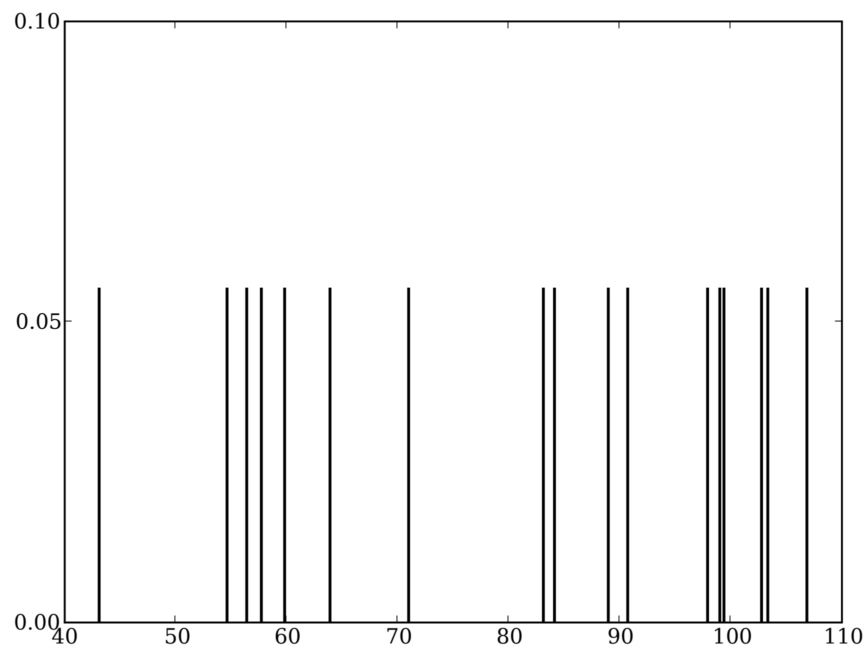
Smoothing

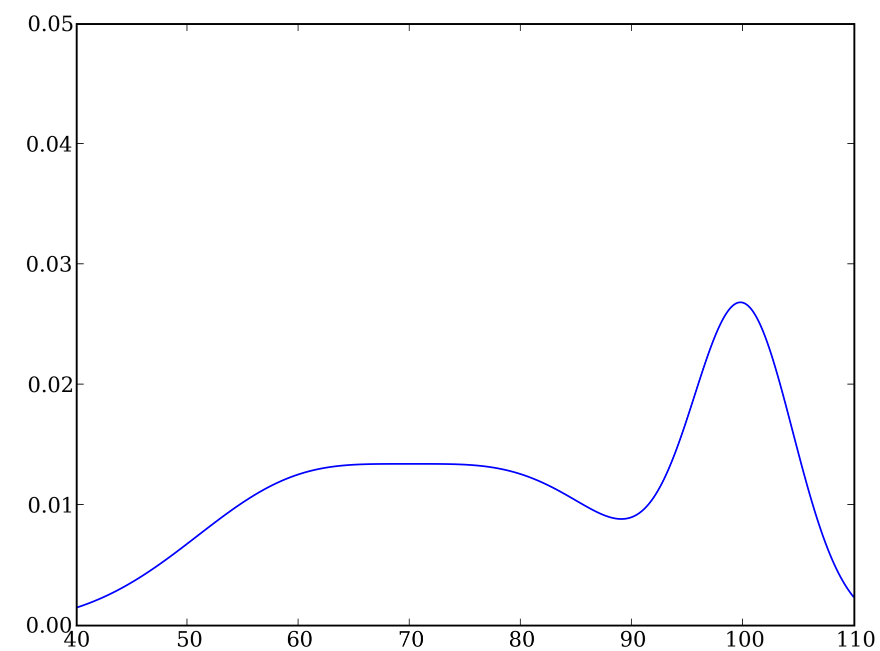

Laplacian Kernel with width λ = 10
0.25
0.2
0.15
0.1
0.05
0
-10
-8
-6
-4
-2
0
2
4
6
8
10

Size matters

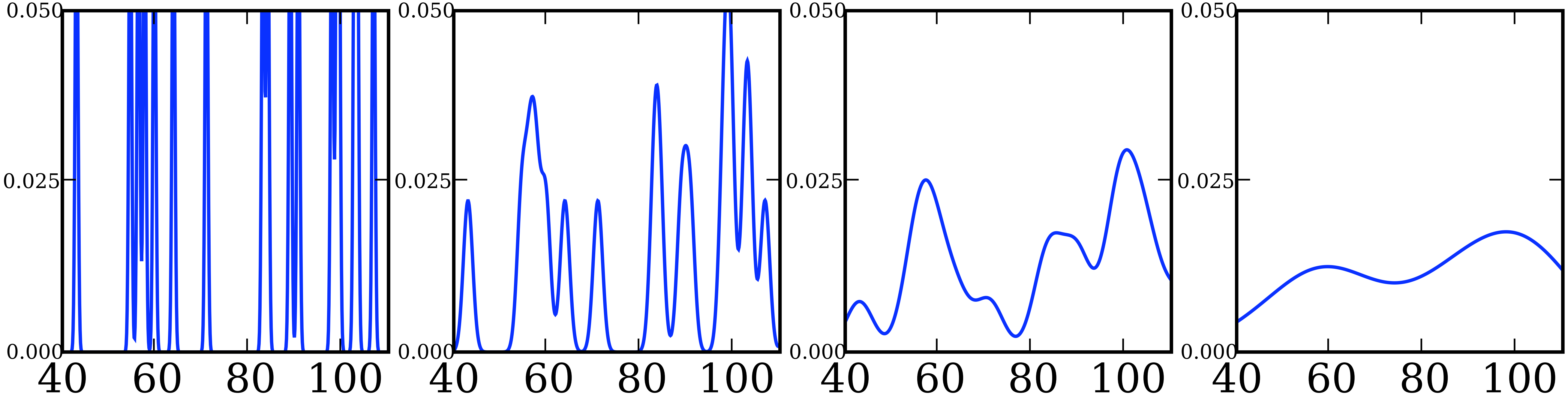


0.3 1 3 10





Size matters
Shape matters mostly in theory



- Kernel width $$k_{x_i}(x) = r^{-d} h\left(\frac{x - x_i}{r}\right)$$
  - Too narrow overfits
  - Too wide smoothes with constant distribution
- How to choose it?

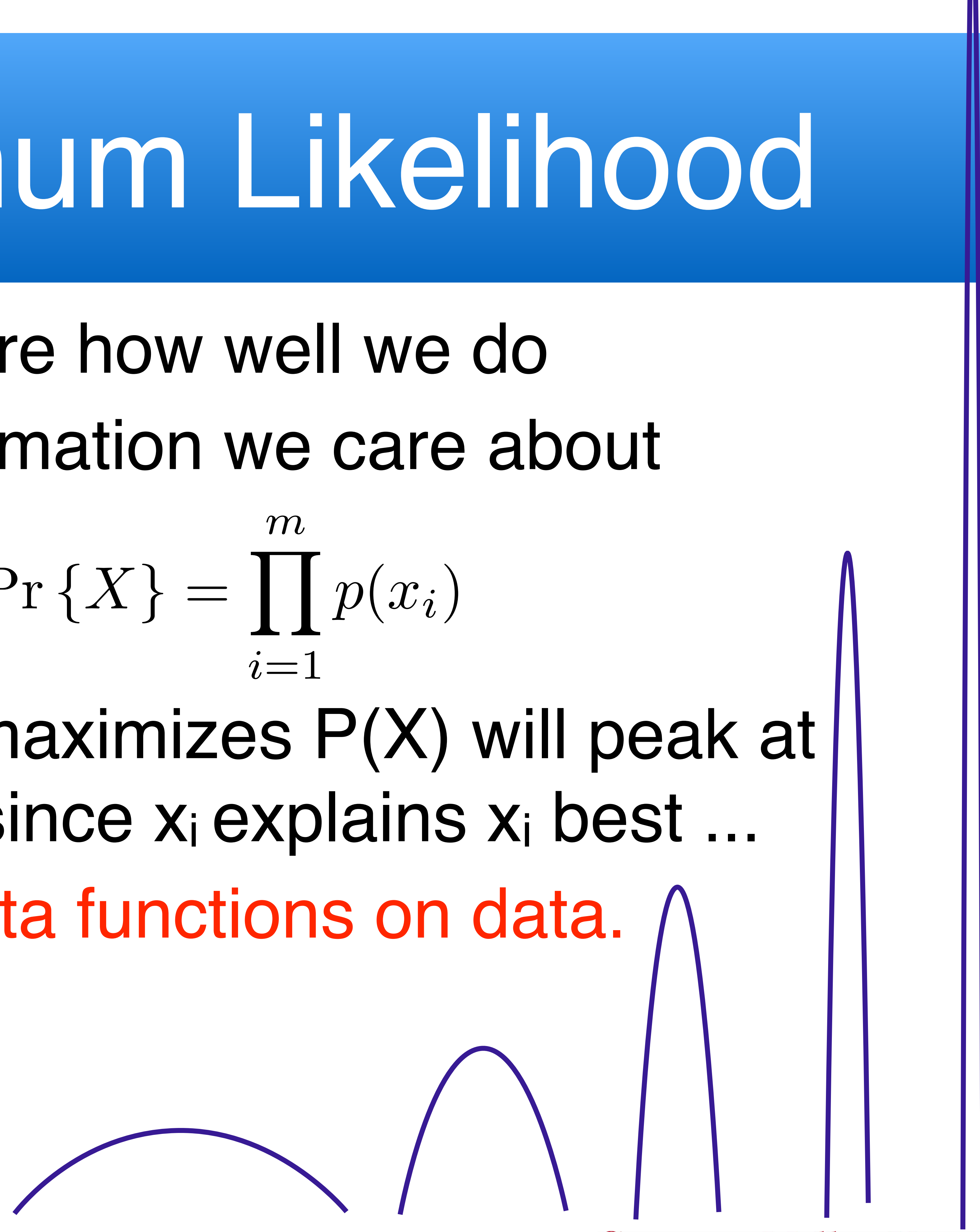
Model Selection

Maximum Likelihood

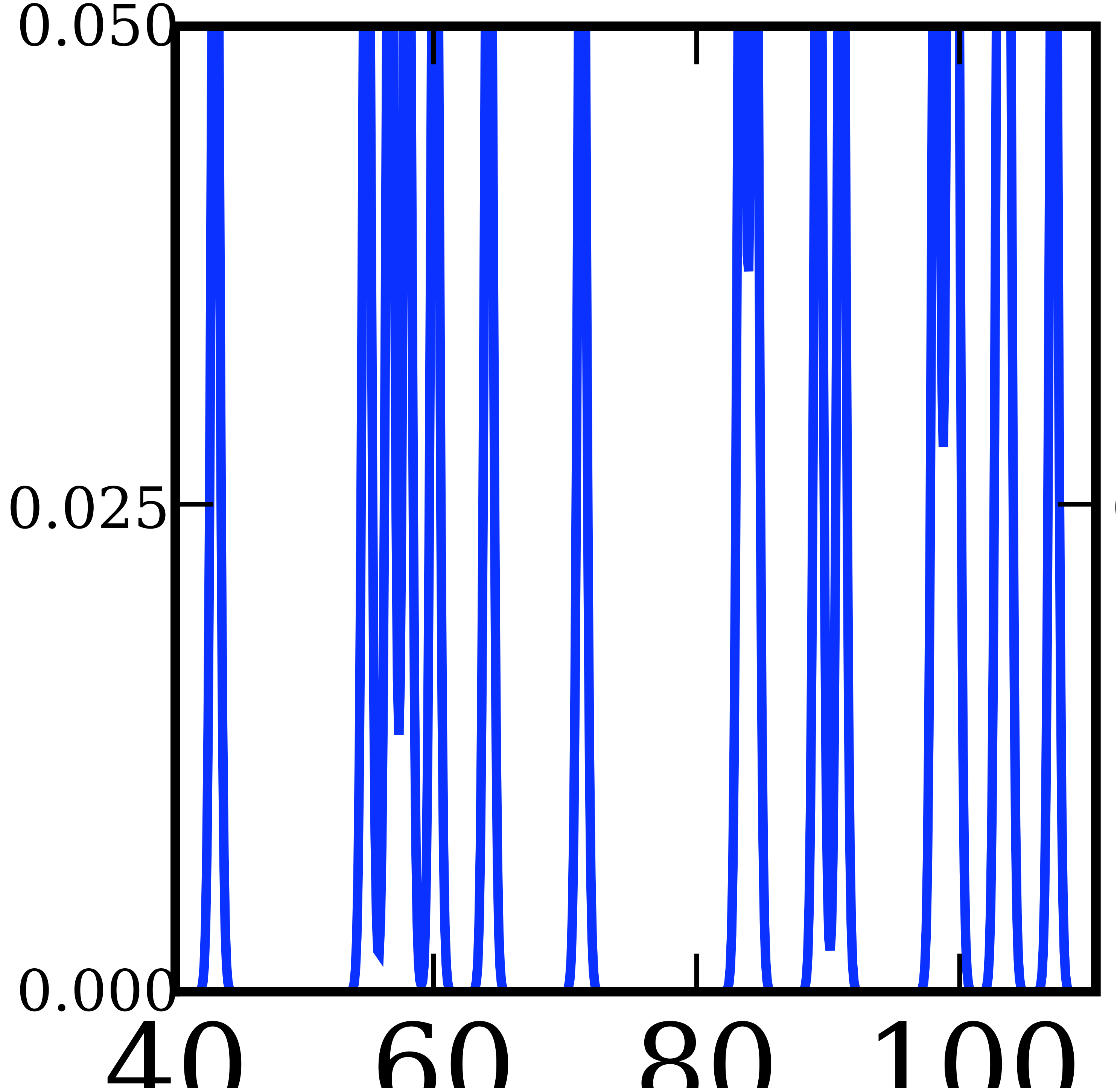
- Need to measure how well we do
- For density estimation we care about

$$\Pr\{X\} = \prod_{i=1}^{m} p(x_i)$$

- Finding a that maximizes P(X) will peak at all data points since x_i explains x_i best ...
- Maxima are delta functions on data.
- Overfitting!

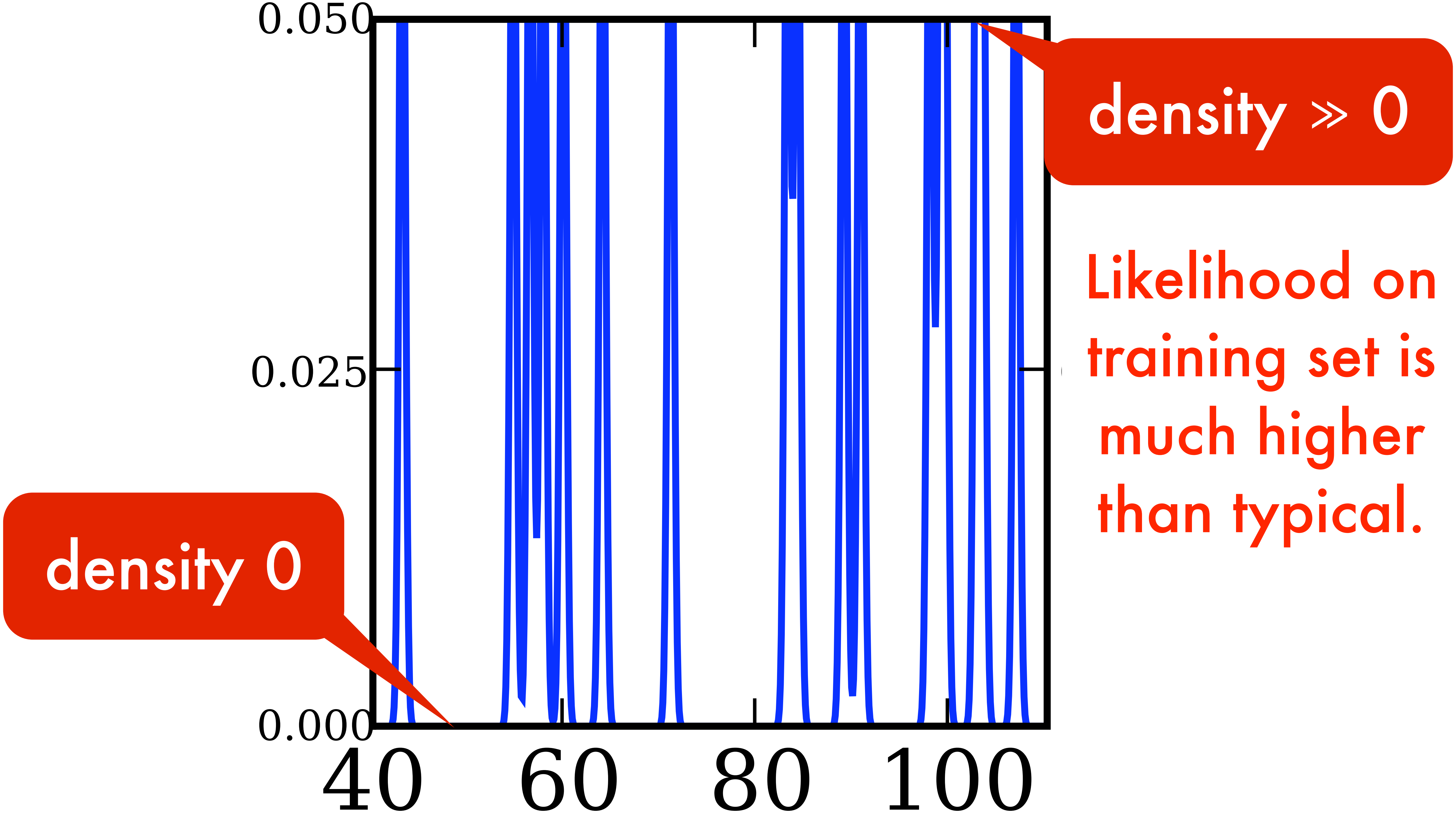


Overfitting



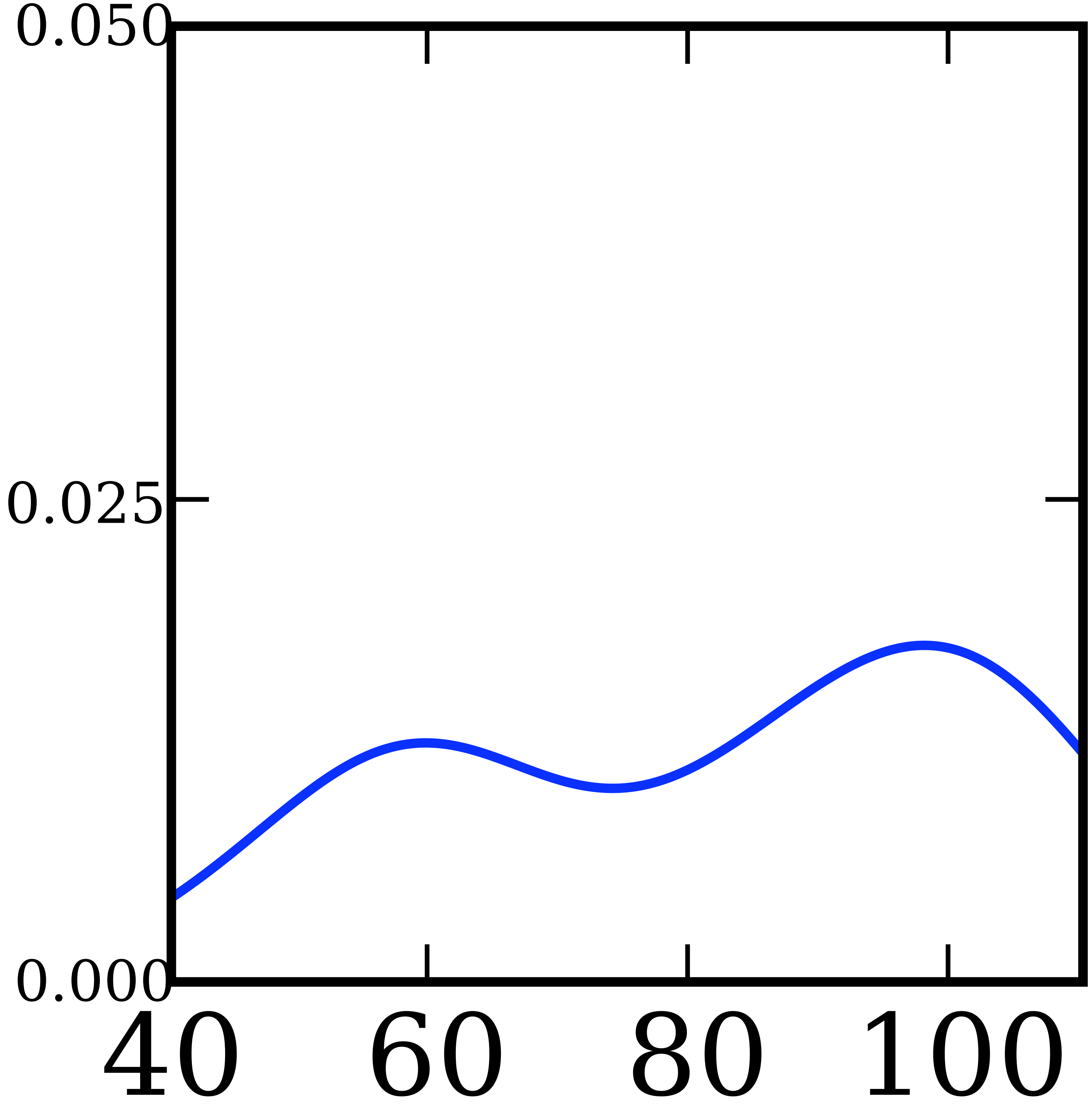
Likelihood on training set is much higher than typical.

Overfitting



Likelihood on training set is much higher than typical.

Underfitting



Likelihood on training set is very similar to typical one.

Too simple.

Model Selection

- Validation
  - Use some of the data to estimate density.
  - Use other part to evaluate how well it works
  - Pick the parameter that works best

$$\mathcal{L}(X'|X) := \frac{1}{n'} \sum_{i=1}^{n'} \log \hat{p}(x'_i)$$

- Learning Theory
  - Use data to build model
  - Measure complexity and use this to bound

$$\frac{1}{n} \sum_{i=1}^{n} \log \hat{p}(x_i) - \mathbf{E}_x \left[\log \hat{p}(x) \right]$$

Model Selection

easy

- Validation
  - Use some of the data to estimate density.
  - Use other part to evaluate how well it works
  - Pick the parameter that works best

$$\mathcal{L}(X'|X) := \frac{1}{n'} \sum_{i=1}^{n'} \log \hat{p}(x'_i)$$

- Learning Theory
  - Use data to build model
  - Measure complexity and use this to bound

$$\frac{1}{n} \sum_{i=1}^{n} \log \hat{p}(x_i) - \mathbf{E}_x \left[\log \hat{p}(x)\right]$$

Model Selection

- Validation
  - Use some of the data to estimate density.
  - Use other part to evaluate how well it works
  - Pick the parameter that works best

$$\mathcal{L}(X'|X) := \frac{1}{n'} \sum_{i=1}^{n'} \log \hat{p}(x_i')$$

easy

wasteful

- Learning Theory
  - Use data to build model
  - Measure complexity and use this to bound

$$\frac{1}{n} \sum_{i=1}^{n} \log \hat{p}(x_i) - \mathbf{E}_x \left[\log \hat{p}(x)\right]$$

Model Selection

- Validation
  - Use some of the data to estimate density.
  - Use other part to evaluate how well it works
  - Pick the parameter that works best

$$\mathcal{L}(X'|X) := \frac{1}{n'} \sum_{i=1}^{n'} \log \hat{p}(x'_i)$$

- Learning Theory
  - Use data to build model
  - Measure complexity and use this to bound

$$\frac{1}{n} \sum_{i=1}^{n} \log \hat{p}(x_i) - \mathbf{E}_x \left[\log \hat{p}(x)\right]$$

easy

wasteful

difficult

Model Selection

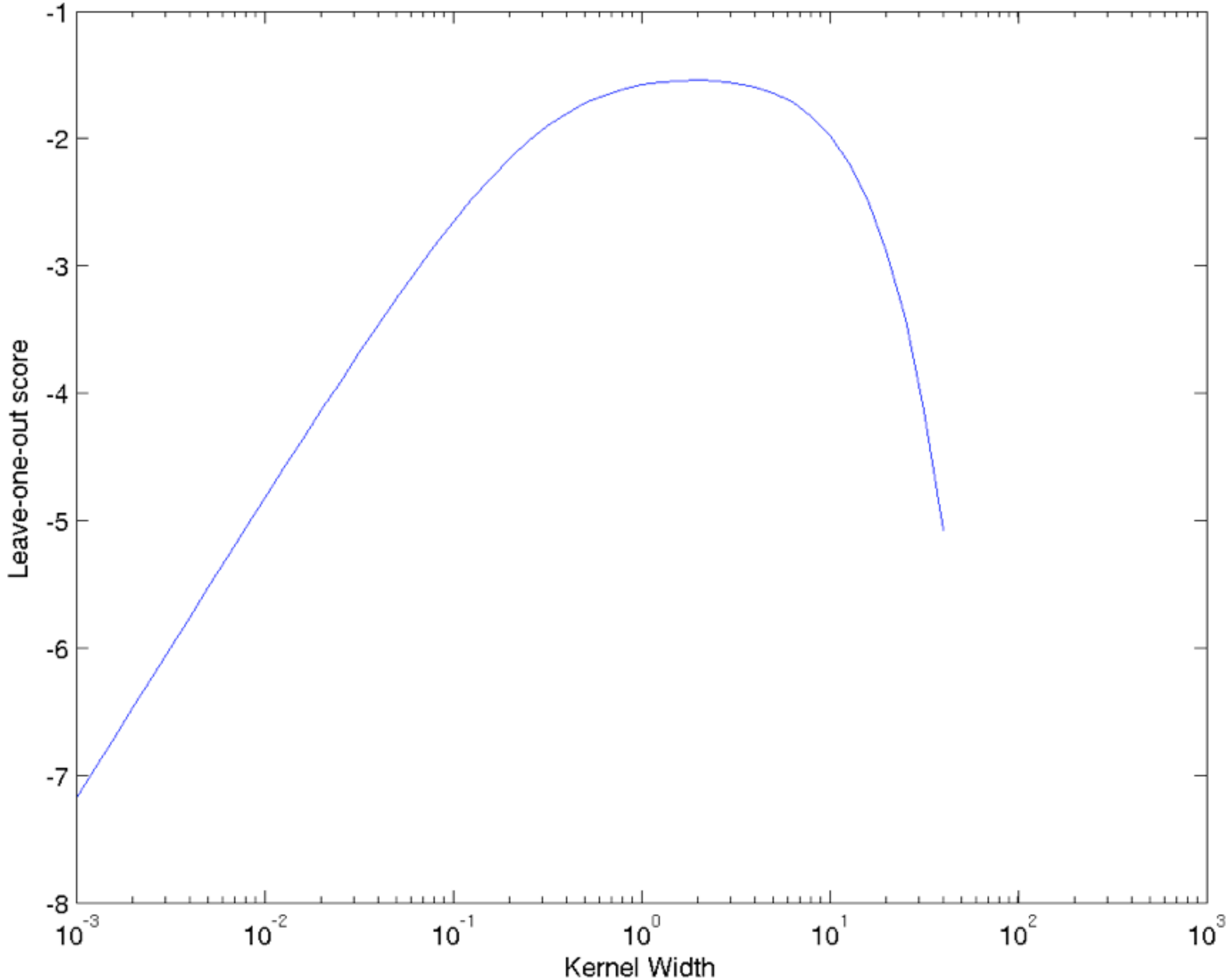
- Leave-one-out Crossvalidation
  - Use almost all data to estimate density.
  - Use single instance to estimate how well it works

$$\log p(x_i | X \backslash x_i) = \log \frac{1}{n-1} \sum_{j \neq i} k(x_i, x_j)$$

  - This has huge variance
  - Average over estimates for all training data
  - Pick the parameter that works best
- Simple implementation

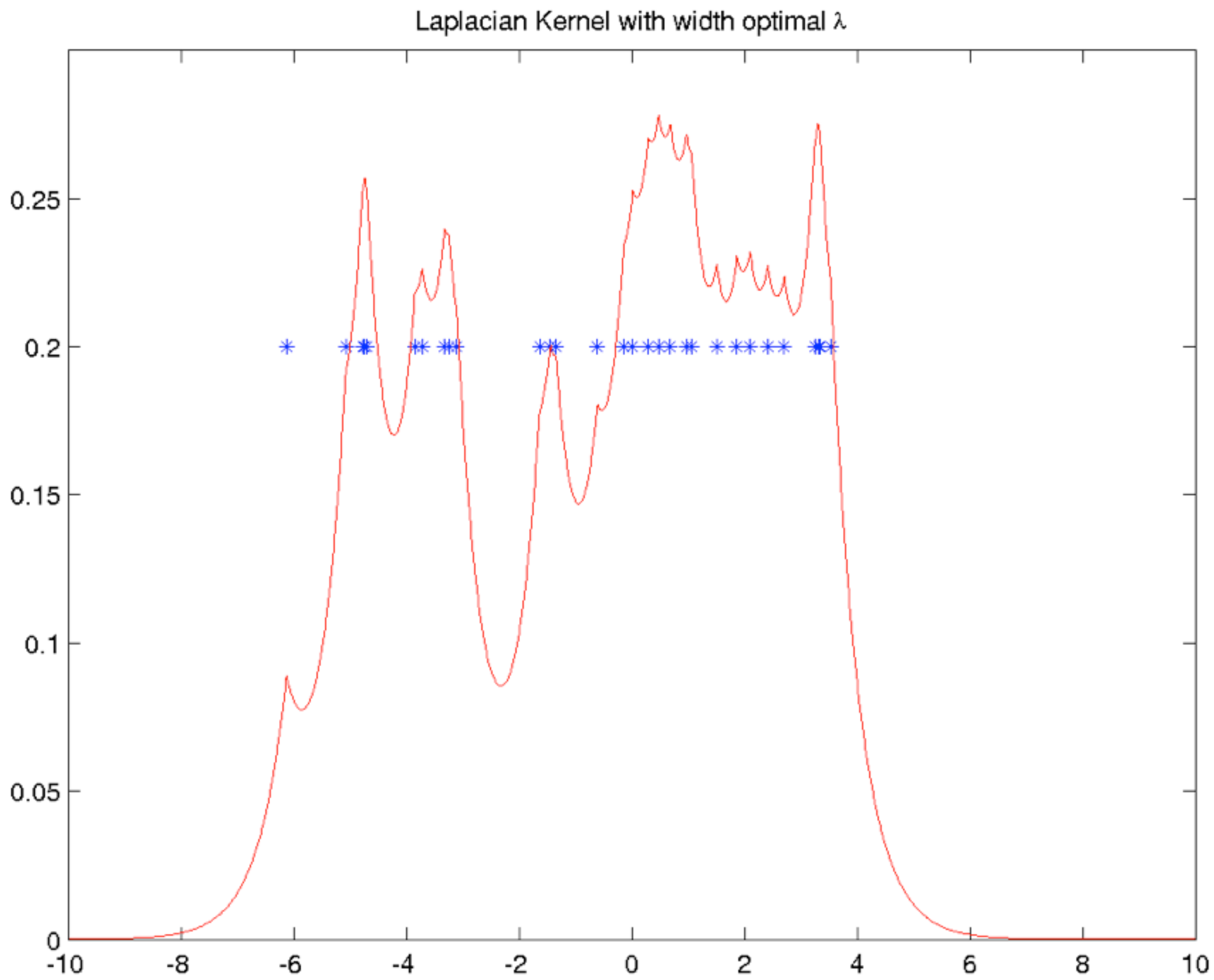
$$\frac{1}{n} \sum_{i=1}^{n} \log \left[\frac{n}{n-1} p(x_i) - \frac{1}{n-1} k(x_i, x_i) \right] \text{ where } p(x) = \frac{1}{n} \sum_{i=1}^{n} k(x_i, x)$$

Leave-one out estimate



niversity

Optimal estimate



iversity

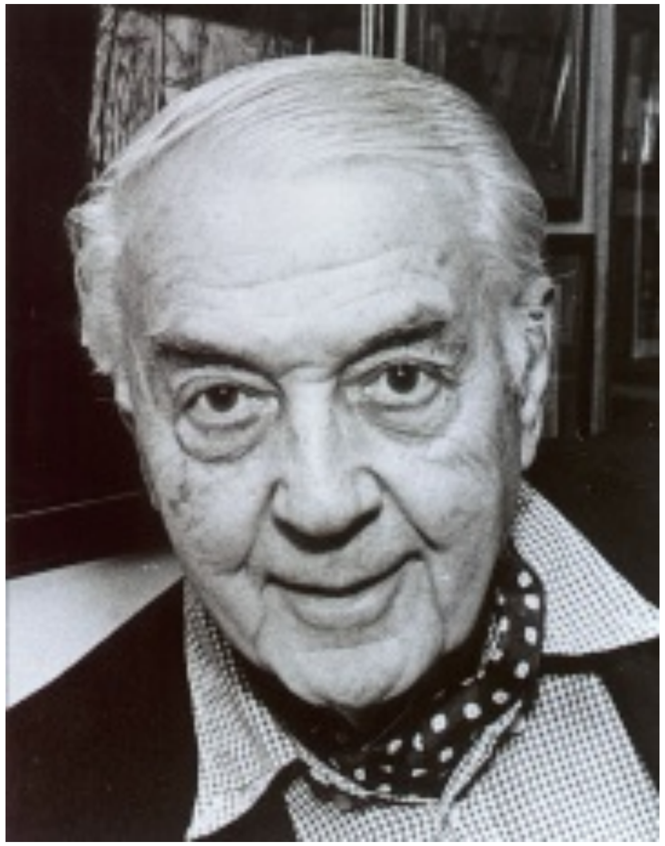
Model Selection

- k-fold Crossvalidation
  - Partition data into k blocks (typically 10)
  - Use all but one block to compute estimate
  - Use remaining block as validation set
  - Average over all validation estimates

$$\frac{1}{k}\sum_{i=1}^{k} l(p(X_i|X\backslash X_i))$$

- Almost unbiased, e.g. via Luntz and Brailovski, 1969 (the error is estimated for a (k-1)/k sized set)
- Pick best parameter (why must we not check too many?)

Watson Nadaraya Estimator



Geoff Watson

From density estimation to classification

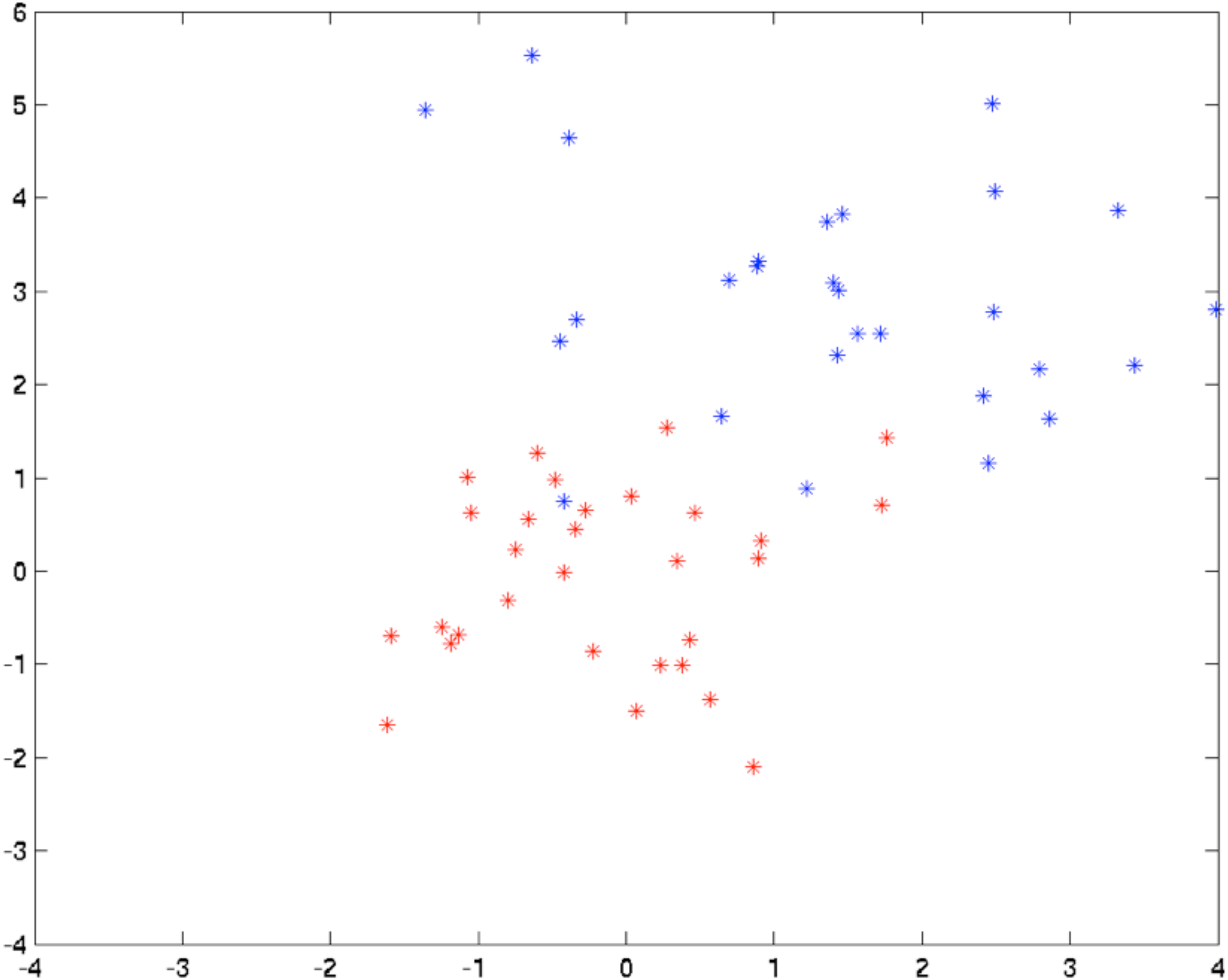
- Binary classification
  - Estimate $p(x|y=1)$ and $p(x|y=-1)$
  - Use Bayes rule

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{\frac{1}{m_y}\sum_{y_i=y} k(x_i, x) \cdot \frac{m_y}{m}}{\frac{1}{m}\sum_i k(x_i, x)}$$

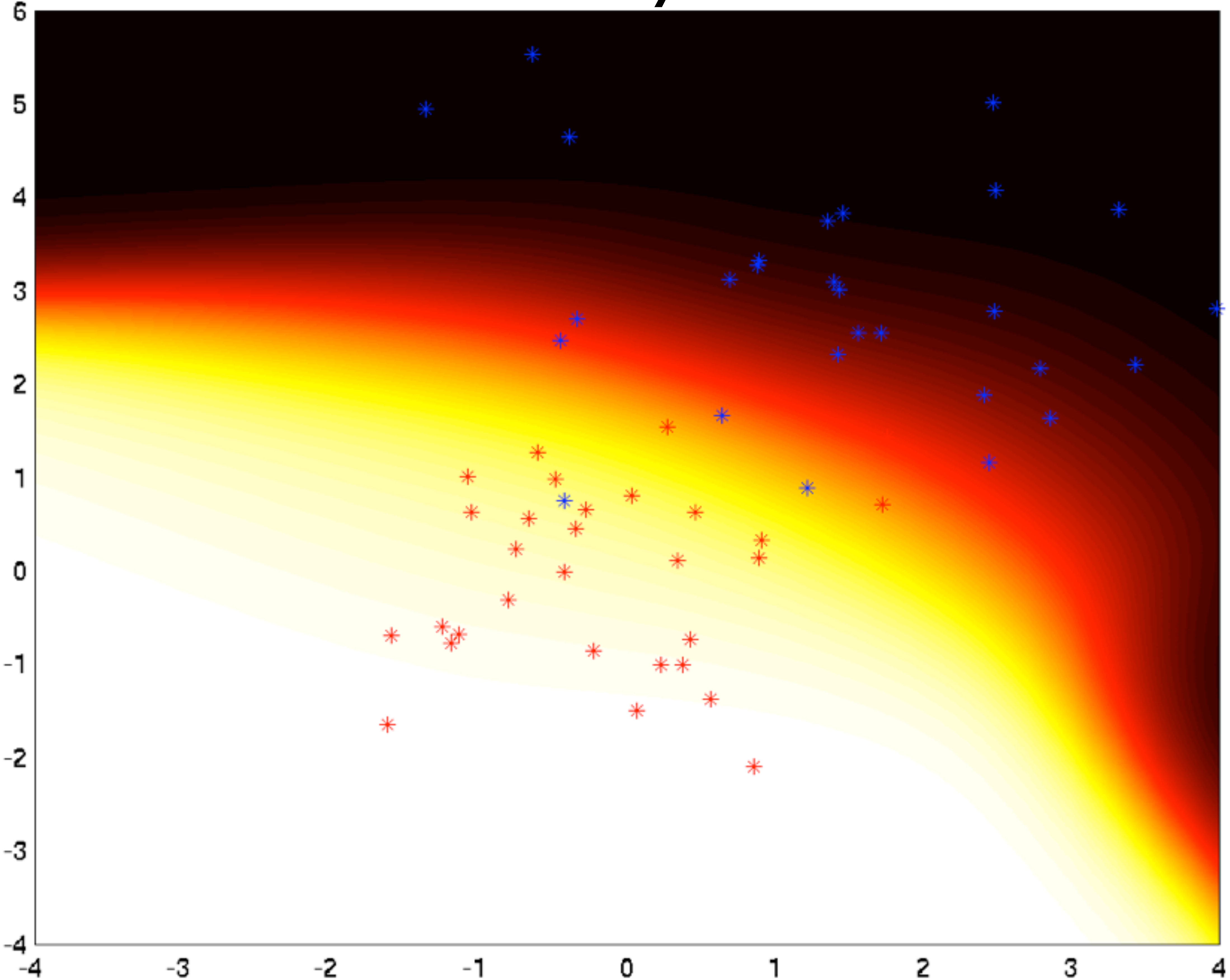
  - Decision boundary

$$p(y=1|x) - p(y=-1|x) = \frac{\sum_j y_j k(x_j, x)}{\sum_i k(x_i, x)} = \sum_j y_j \boxed{\frac{k(x_j, x)}{\sum_i k(x_i, x)}}$$

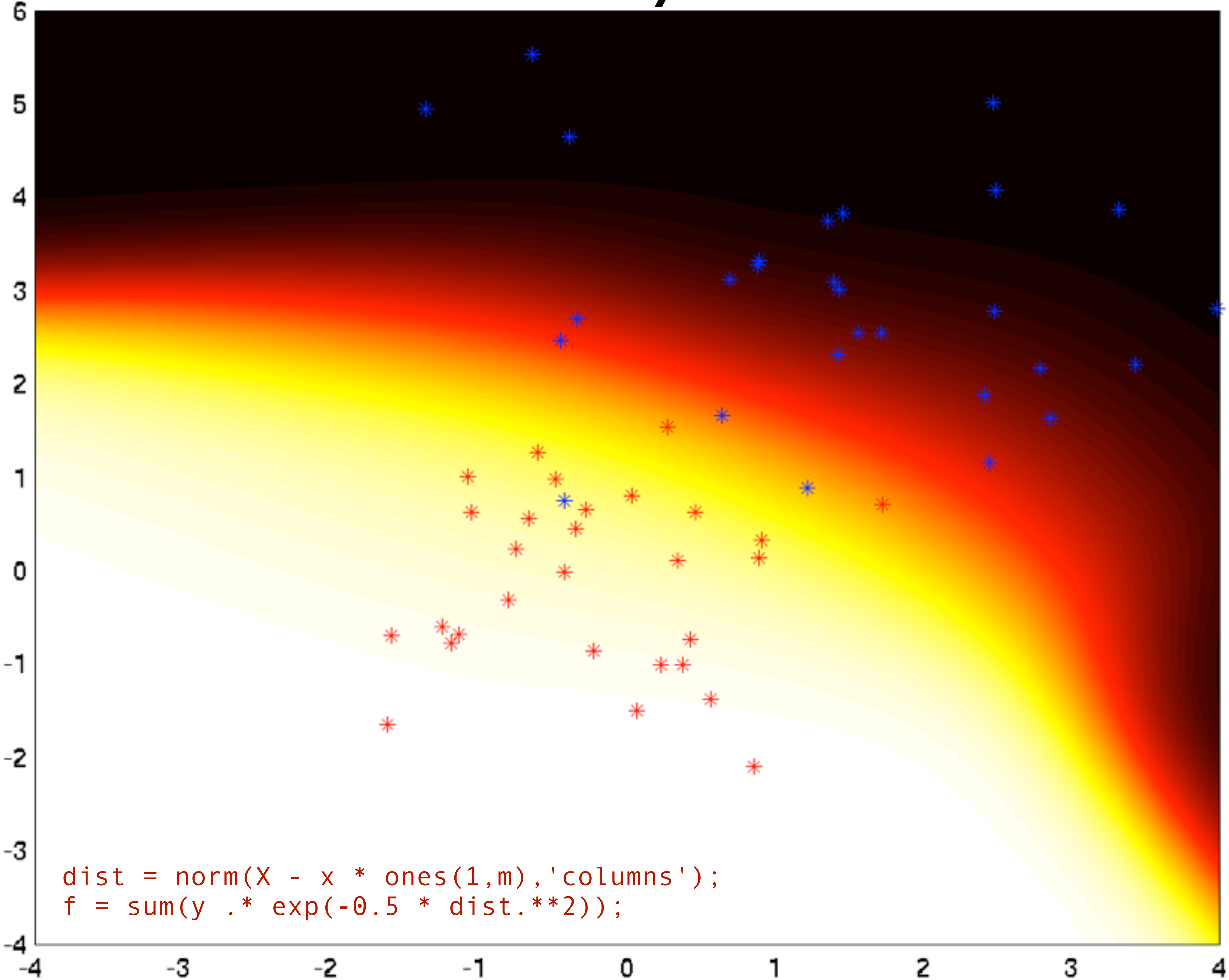
local weights


6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4

Watson-Nadaraya Classifier



Watson-Nadaraya Classifier



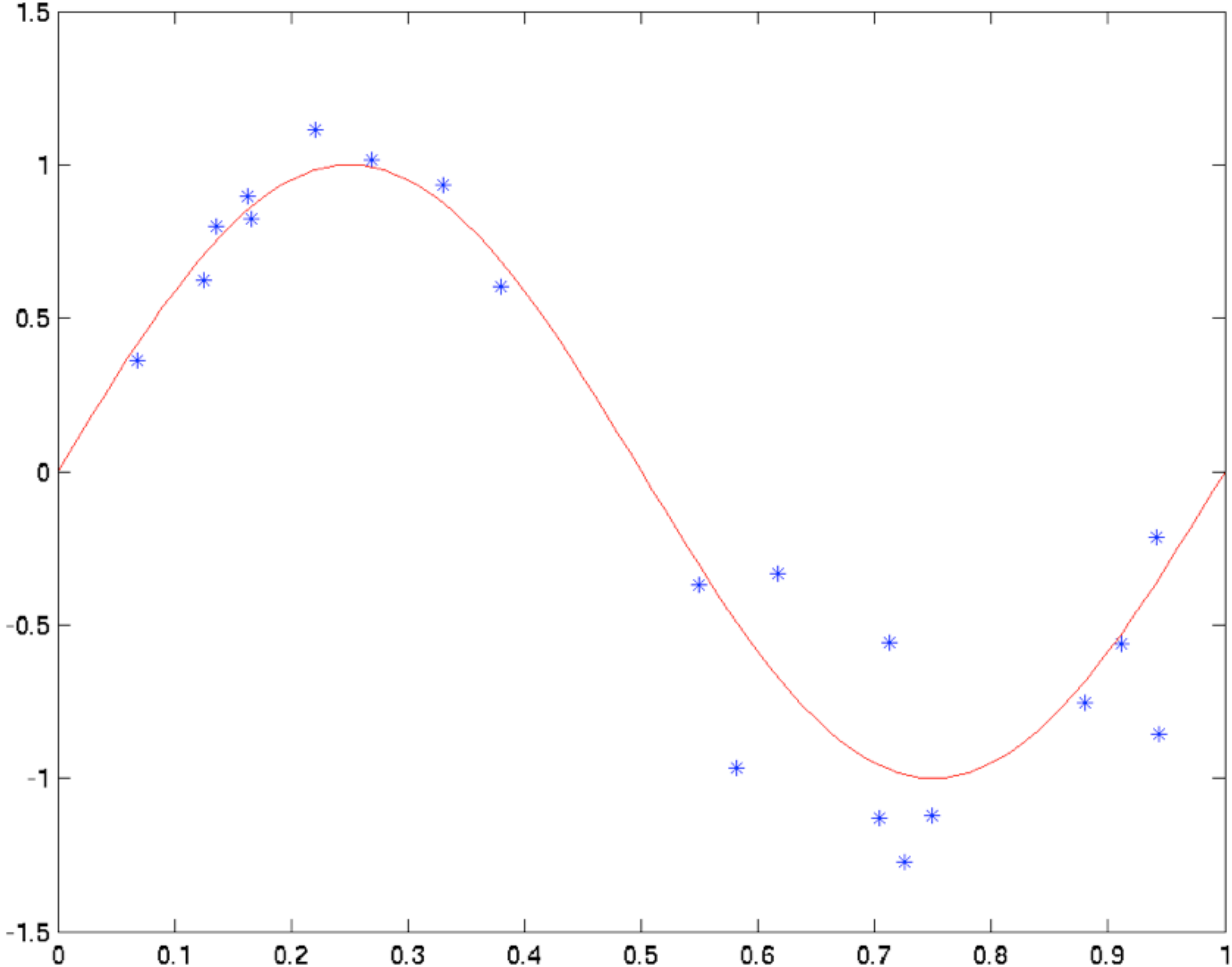
Watson Nadaraya Regression

- Binary classification

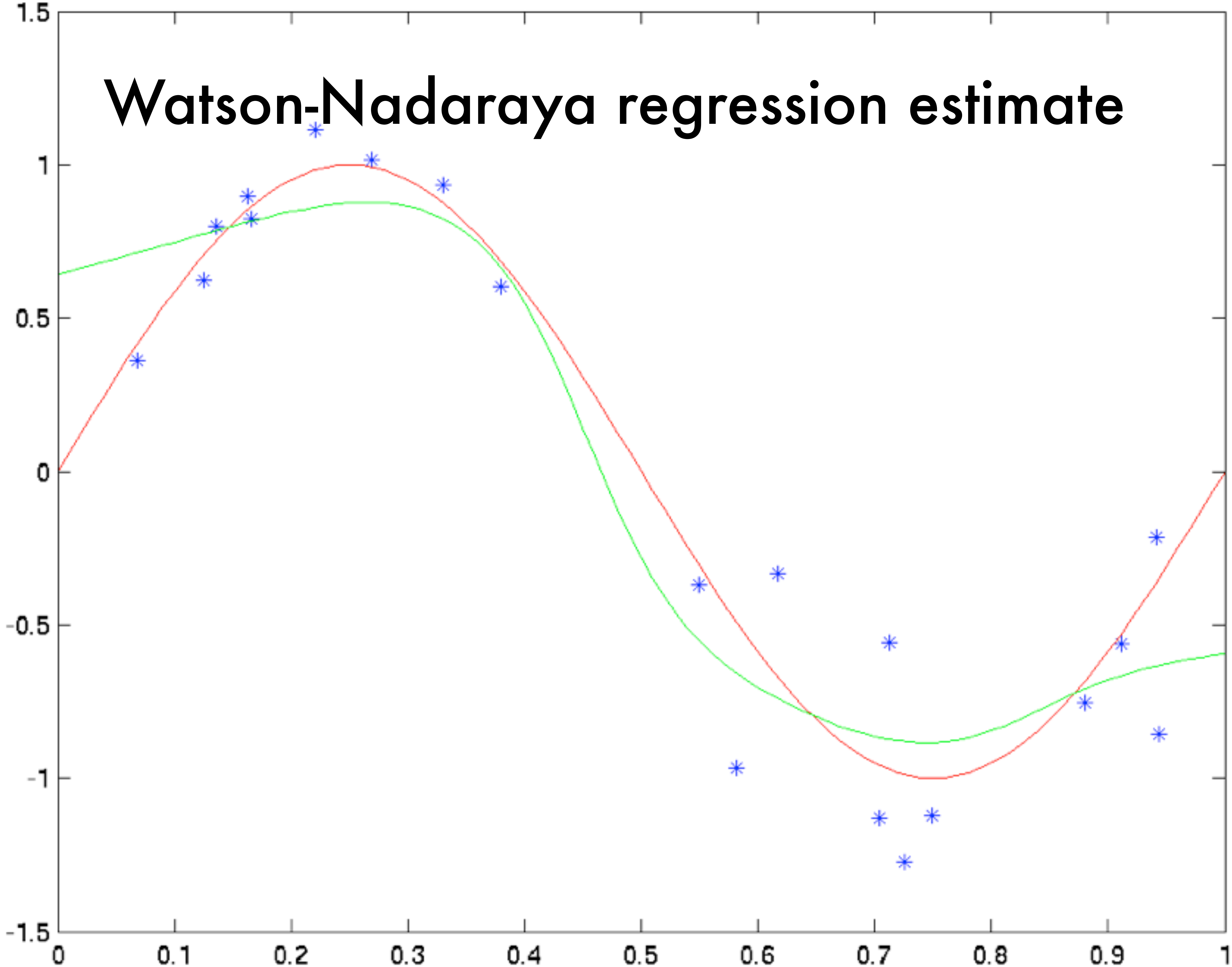
$$p(y=1|x) - p(y=-1|x) = \frac{\sum_j y_j k(x_j, x)}{\sum_i k(x_i, x)} = \sum_j y_j \frac{k(x_j, x)}{\sum_i k(x_i, x)}$$

- Regression - use same weighted expansion

$$\hat{y}(x) = \sum_j \underbrace{y_j}_{\text{labels}} \underbrace{\frac{k(x_j, x)}{\sum_i k(x_i, x)}}_{\text{local weights}}$$


1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Watson-Nadaraya regression estimate

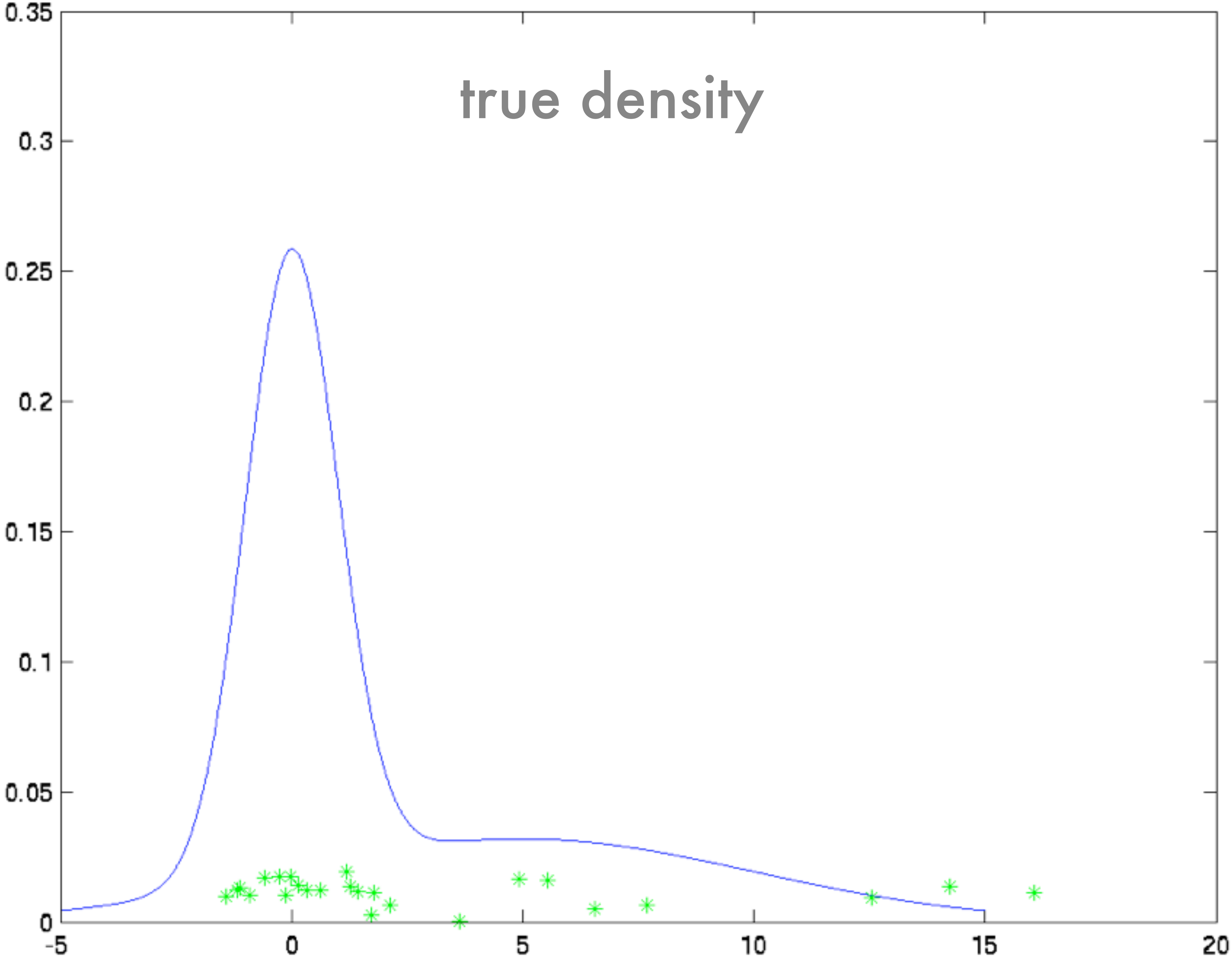


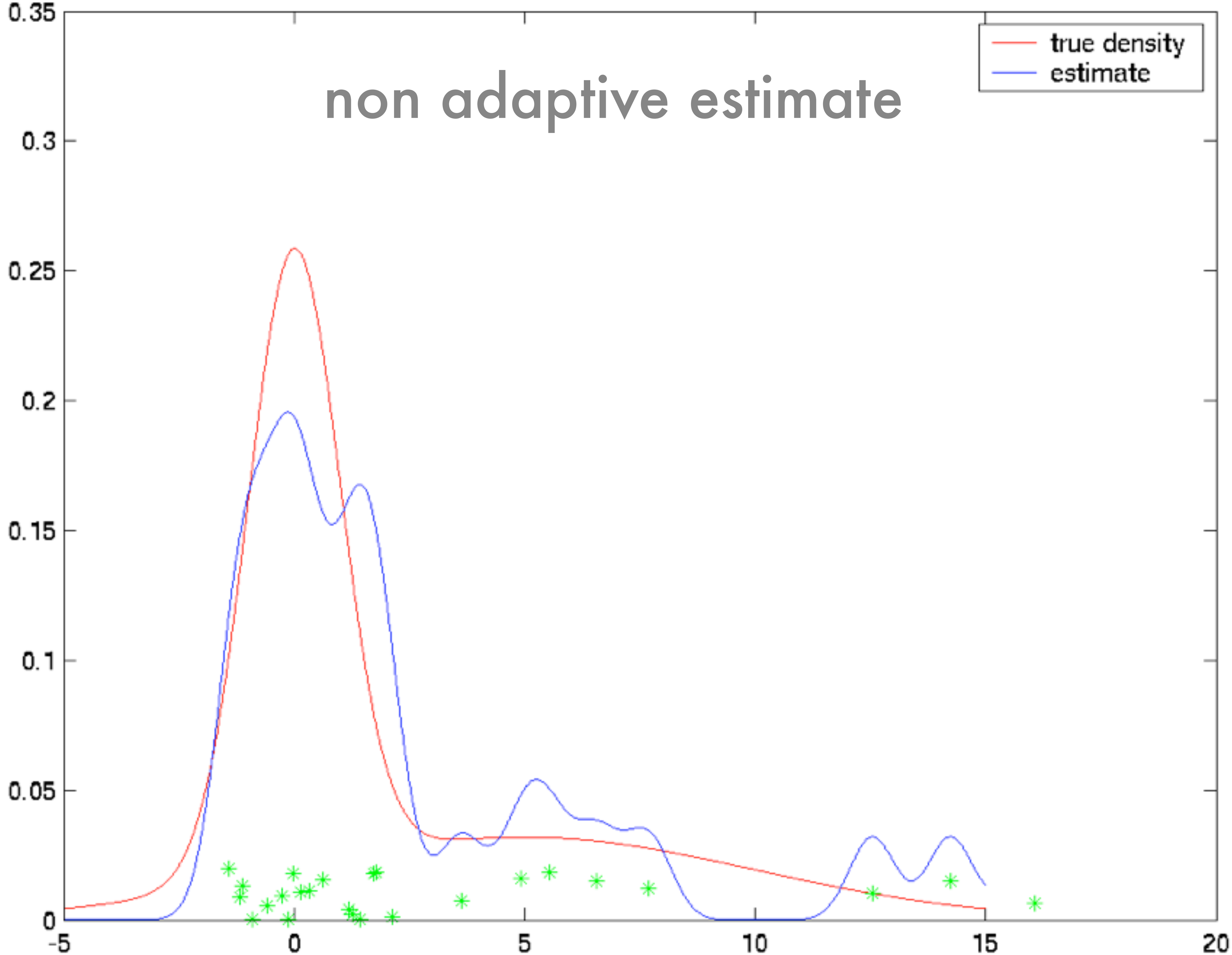
Silverman's rule

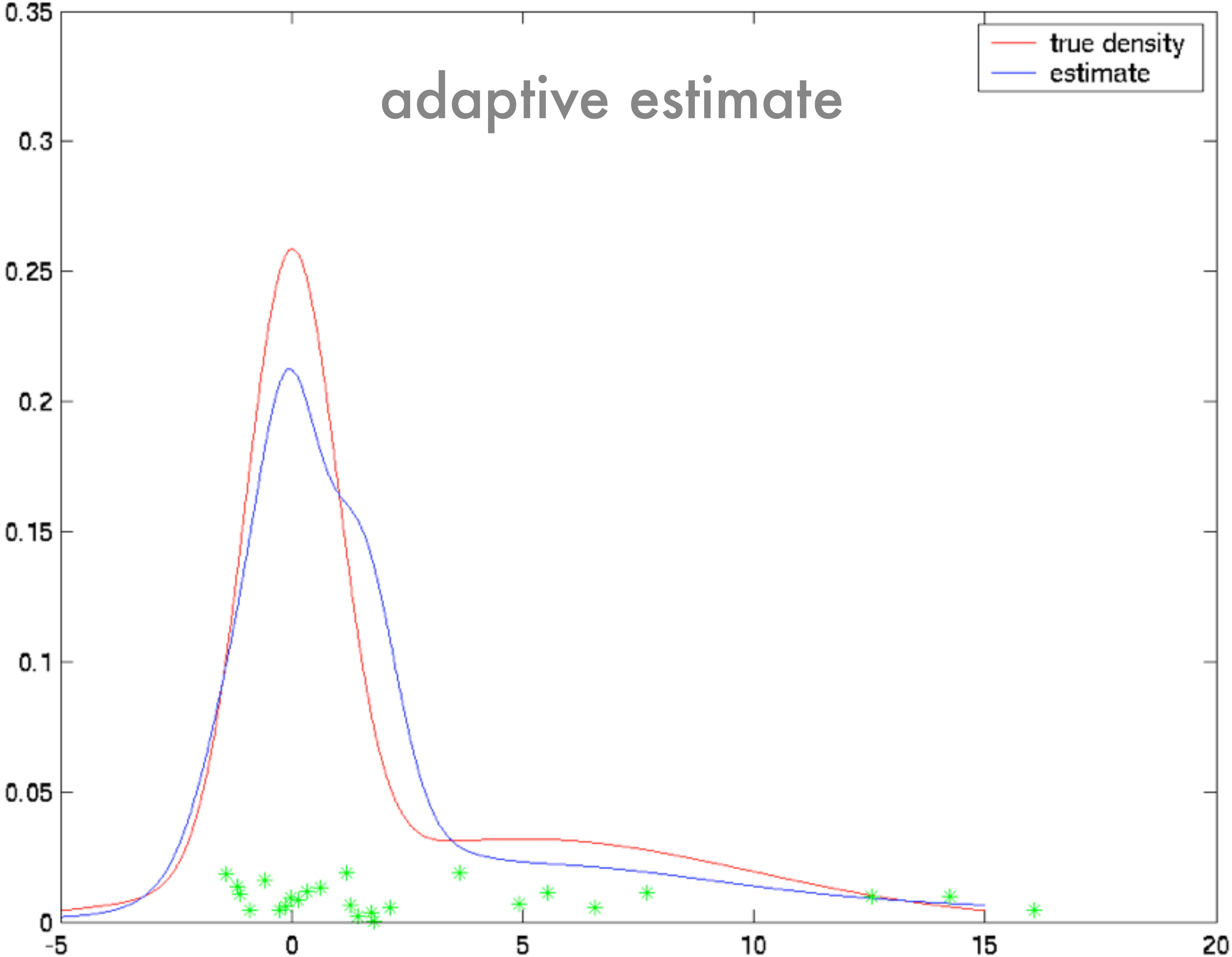
- Chicken and egg problem
  - Want wide kernel for low density region
  - Want narrow kernel where we have much data
  - Need density estimate to estimate density
- Simple hack
  Use average distance from k nearest neighbors

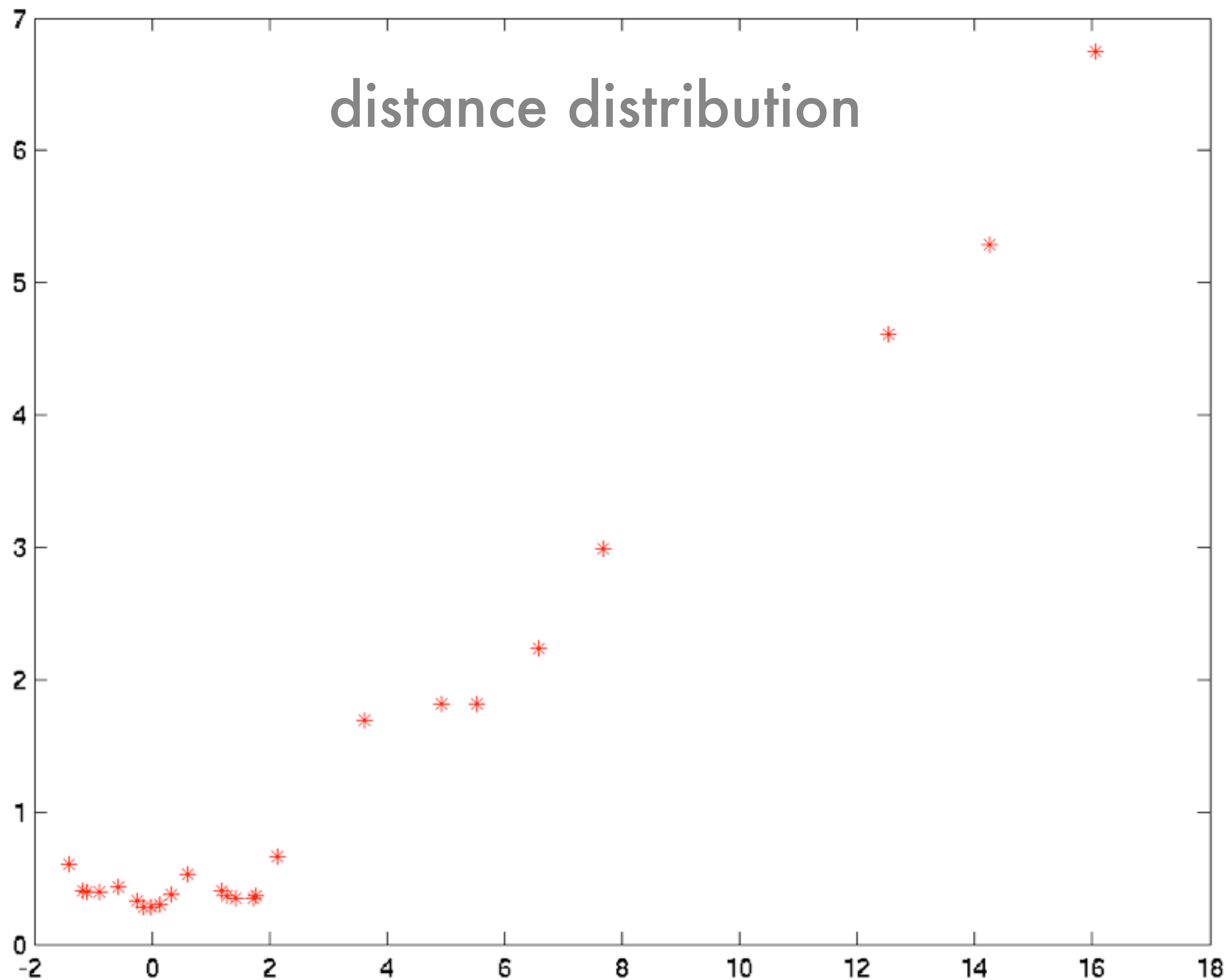
$$r_i = \frac{r}{k} \sum_{x \in \mathrm{NN}(x_i, k)} \|x_i - x\|$$

- Nonuniform bandwidth for smoother.


true density
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-5
0
5
10
15
20


non adaptive estimate
true density
estimate
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-5
0
5
10
15
20

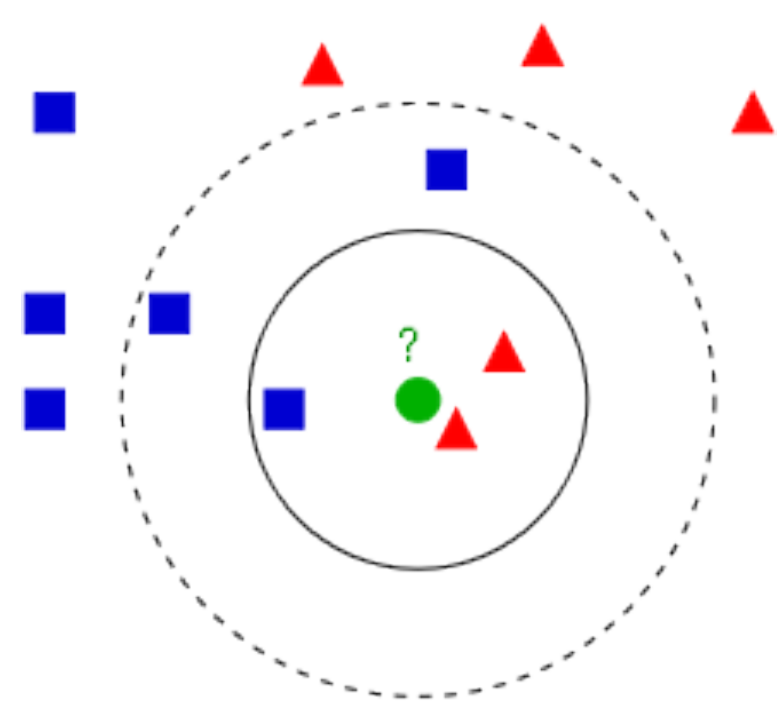

adaptive estimate
true density
estimate
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-5
0
5
10
15
20


distance distribution
0
1
2
3
4
5
6
7
-2
0
2
4
6
8
10
12
14
16
18


Nearest Neighbor
Recap

Nearest Neighbors

- Table lookup
  For previously seen instance remember label
- Nearest neighbor
  - Pick label of most similar neighbor
  - Slight improvement - use k-nearest neighbors
  - For regression average
  - Really useful baseline!
  - Easy to implement for small amounts of data. Why?



Relation to Watson Nadaraya

- Watson Nadaraya estimator

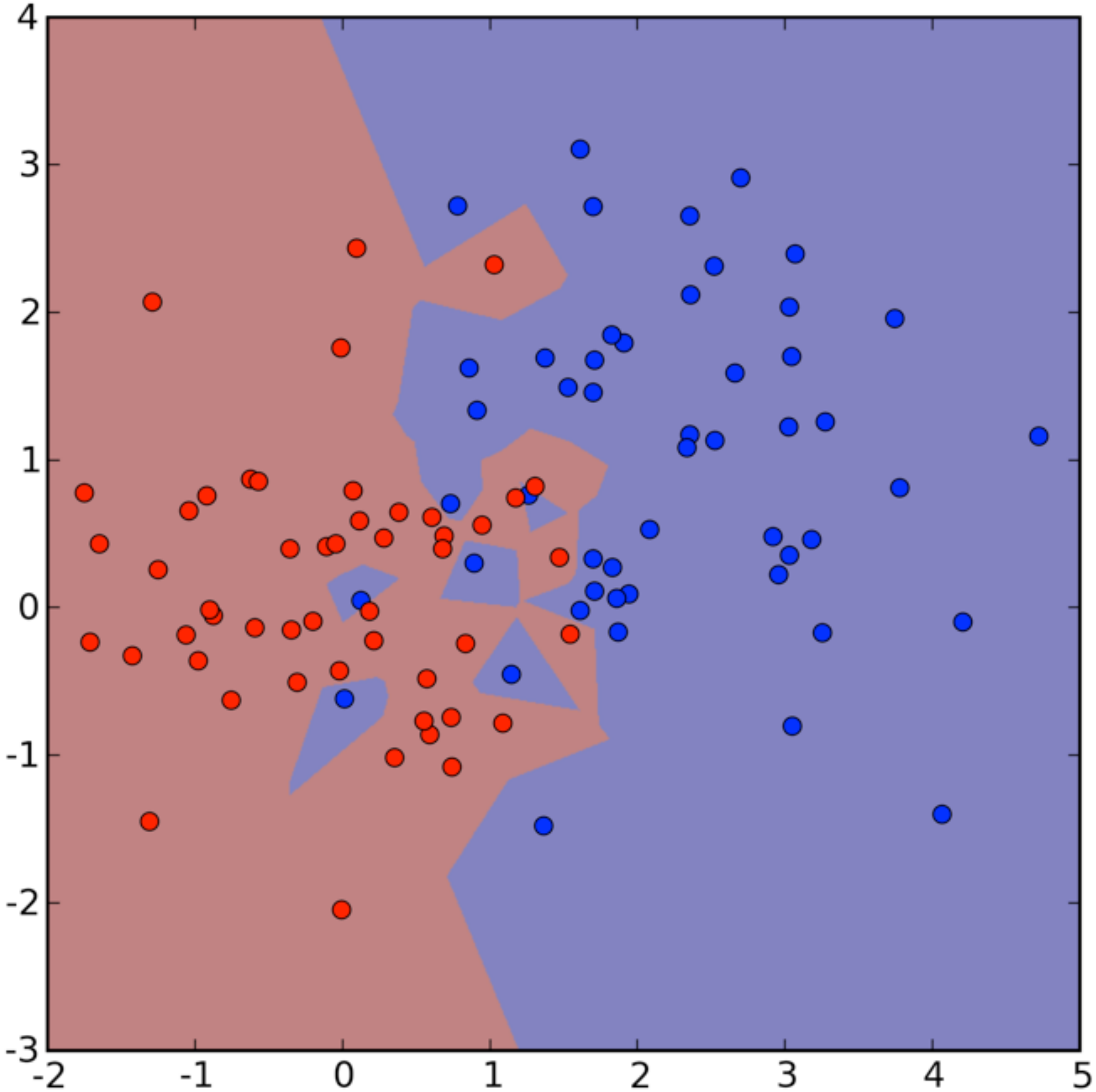
$$\hat{y}(x) = \sum_j y_j \frac{k(x_i, x)}{\sum_i k(x_i, x)} = \sum_j y_j w_j(x)$$

- Nearest neighbor estimator

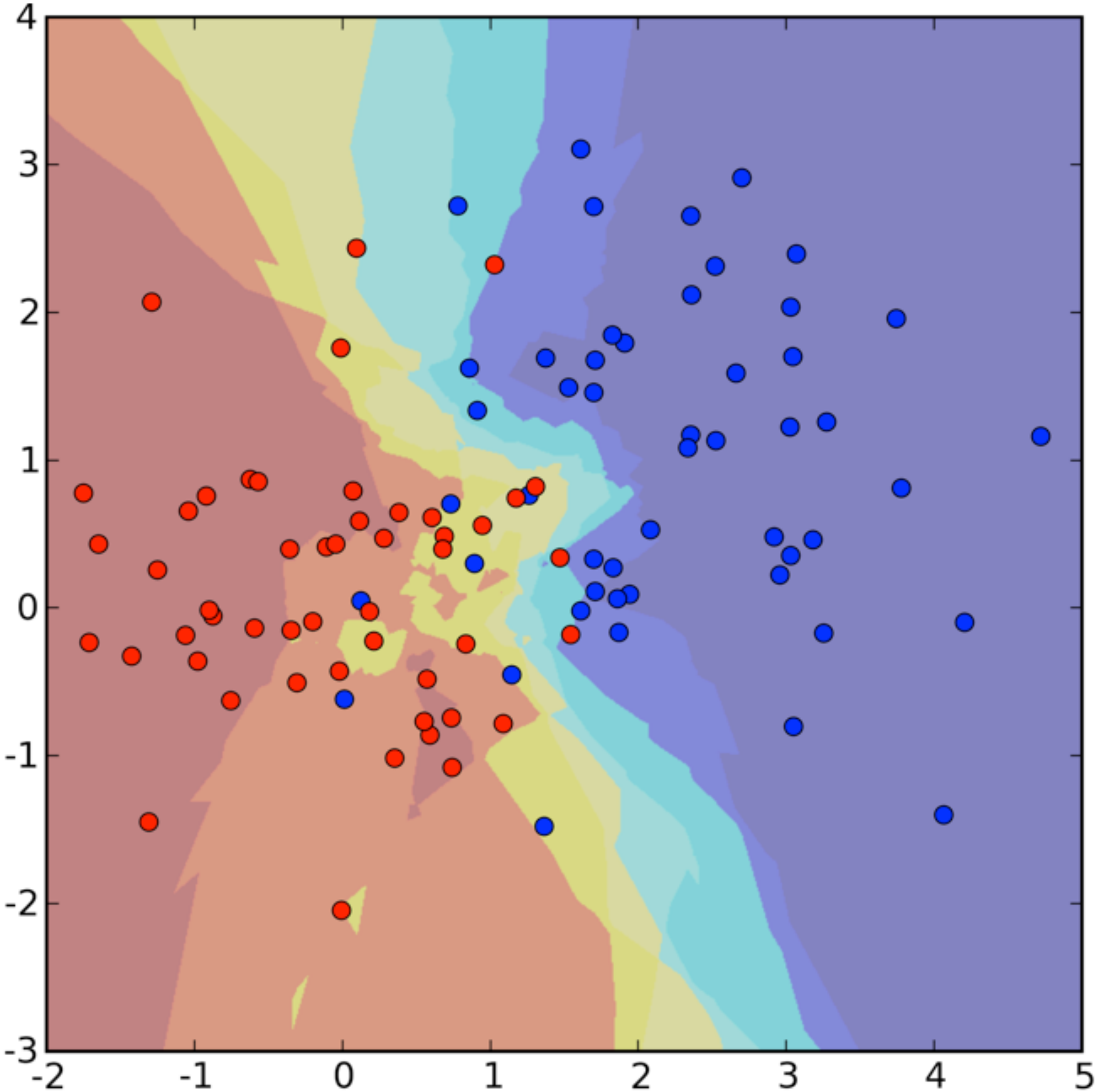
$$\hat{y}(x) = \sum_j y_j \frac{k(x_j, x)}{\sum_i k(x_i, x)} = \sum_j y_j w_j(x)$$

Neighborhood function is hard threshold.

1-Nearest Neighbor

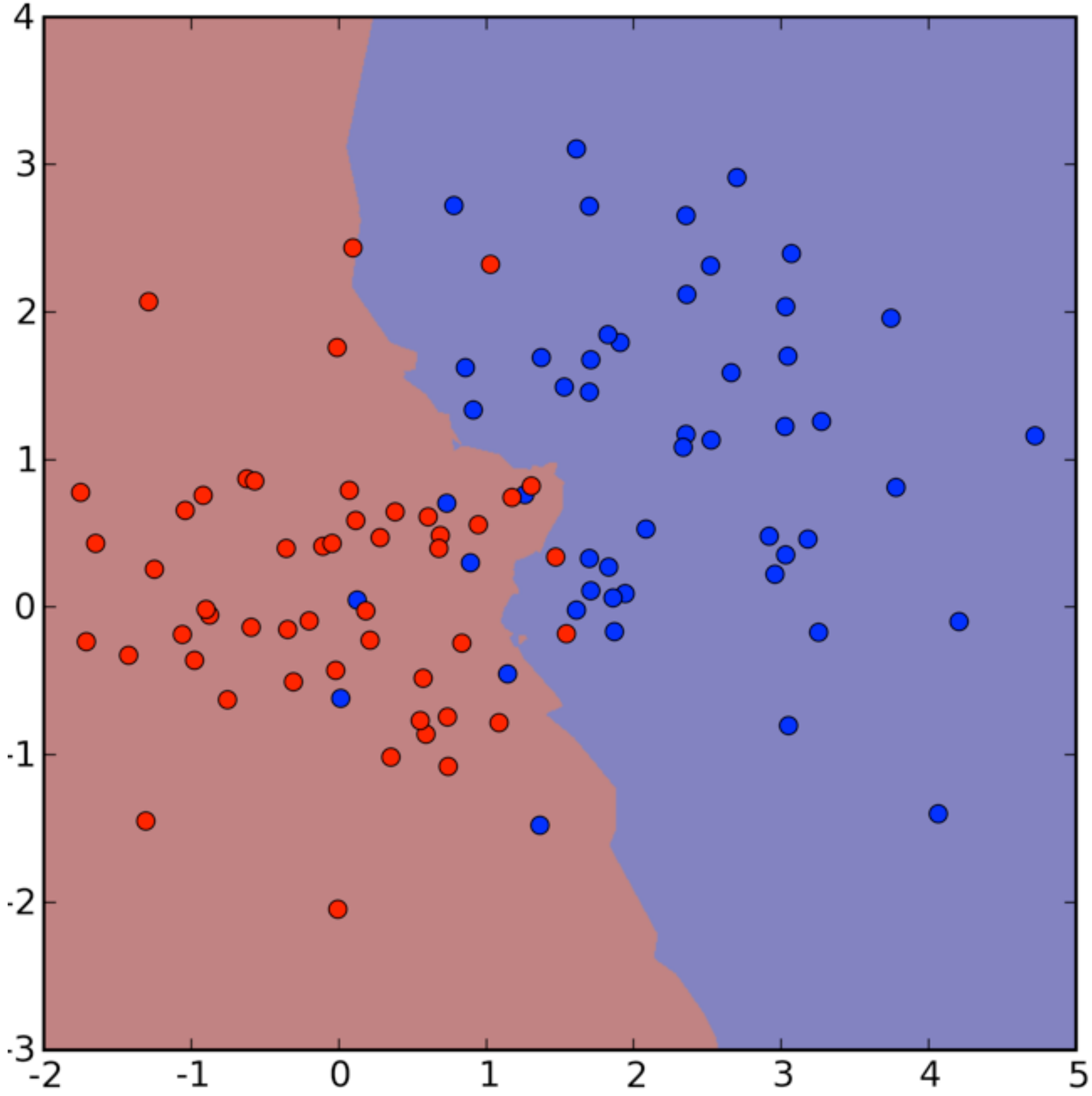


4-Nearest Neighbors

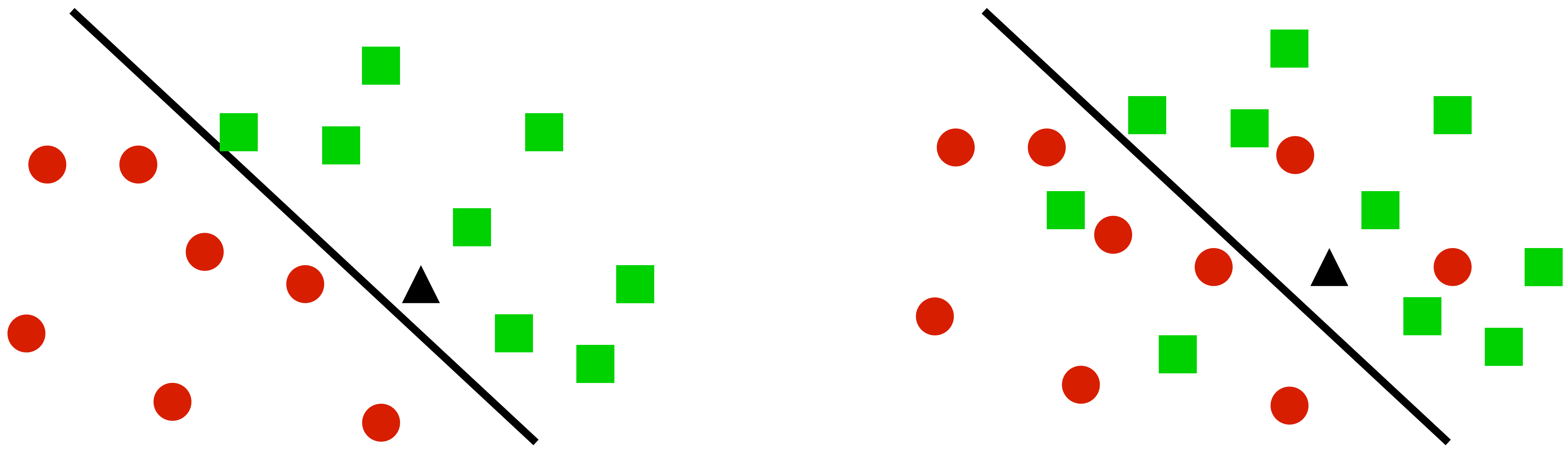


Carnegie Mellon University

4-Nearest Neighbors Sign

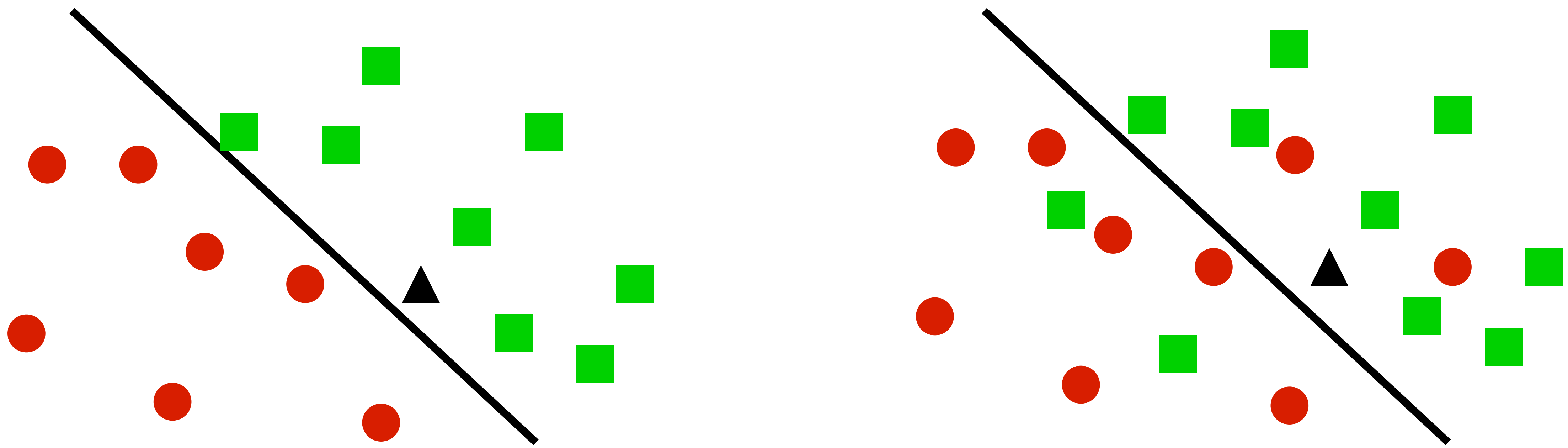


If we get more data



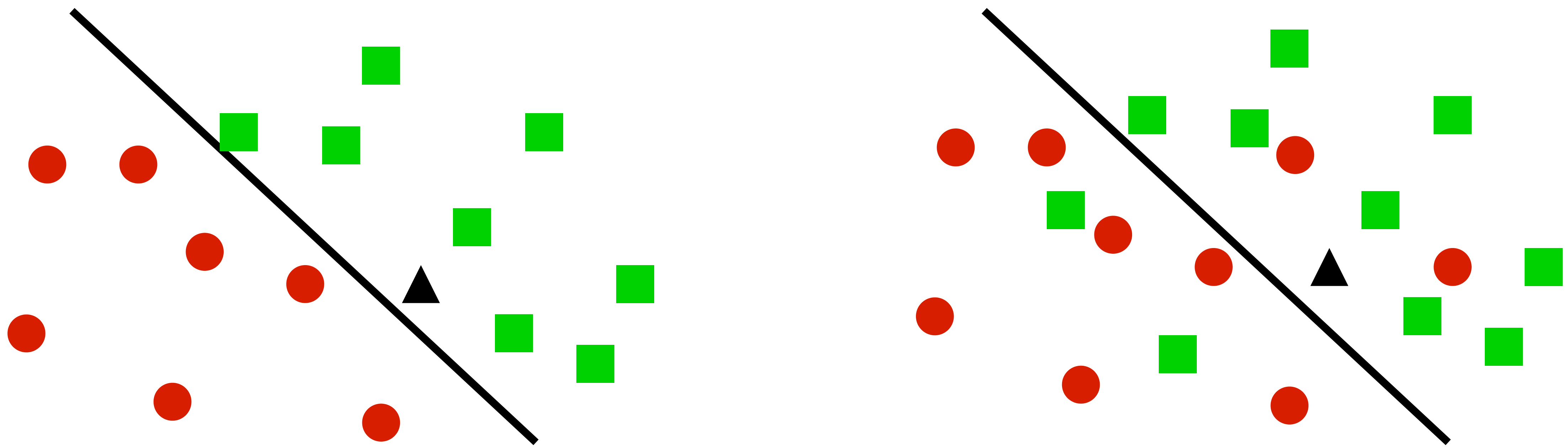
- 1 Nearest Neighbor
  - Converges to perfect solution if clear separation
  - Twice the minimal error rate 2p(1-p) for noisy problems
- k-Nearest Neighbor
  - Converges to perfect solution if clear separation (but needs more data)
  - Converges to minimal error min(p, 1-p) for noisy problems if k increases

1 Nearest Neighbor



- For given point x take є neighborhood N with probability mass > d/n
- Probability that at least one point of n is in this neighborhood is $1-e^{-d}$ so we can make this small
- Assume that probability mass doesn't change much in neighborhood
- Probability that labels of query and point do not match is 2p(1-p) (up to some approximation error in neighborhood)

k Nearest Neighbor



- For given point x take ϵ neighborhood N with probability mass > dk/n
- Small probability that we don't have at least k points in neighborhood.
- Assume that probability mass doesn't change much in neighborhood
- Bound probability that majority of points doesn't match majority for p (e.g. via Hoeffding's theorem for tail). Show that it vanishes
- Error is therefore min(p, 1-p), i.e. Bayes optimal error.

Fast lookup

- KD trees (Moore et al.)
  - Partition space (one dimension at a time)
  - Only search for subset that contains point
- Cover trees (Beygelzimer et al.)
  - Hierarchically partition space with distance guarantees
  - No need for nonoverlapping sets
  - Bounded number of paths to follow (logarithmic time lookup)

MACHINE LEARNING
DEPARTMENT

2.4 Exponential Families

2 Statistics

Alexander Smola
Introduction to Machine Learning 10-701
http://alex.smola.org/teaching/10-701-15

Carnegie Mellon University

Exponential Families

Exponential Families

Exponential Families

- Density function

$$p(x;\theta) = \exp\left(\langle \phi(x), \theta \rangle - g(\theta)\right)$$

$$\text{where } g(\theta) = \log \sum_{x'} \exp\left(\langle \phi(x'), \theta \rangle\right)$$

Exponential Families

- Density function

$$p(x;\theta) = \exp\left(\langle \phi(x), \theta \rangle - g(\theta)\right)$$

$$\text{where } g(\theta) = \log \sum_{x'} \exp\left(\langle \phi(x'), \theta \rangle\right)$$

- Log partition function generates cumulants

$$\partial_\theta g(\theta) = \mathbf{E}\left[\phi(x)\right]$$

$$\partial_\theta^2 g(\theta) = \mathrm{Var}\left[\phi(x)\right]$$

Exponential Families

- Density function

$$p(x;\theta) = \exp\left(\langle \phi(x), \theta \rangle - g(\theta)\right)$$

$$\text{where } g(\theta) = \log \sum_{x'} \exp\left(\langle \phi(x'), \theta \rangle\right)$$

- Log partition function generates cumulants

$$\partial_\theta g(\theta) = \mathbf{E}\left[\phi(x)\right]$$

$$\partial_\theta^2 g(\theta) = \mathrm{Var}\left[\phi(x)\right]$$

- g is convex (second derivative is p.s.d.)

Examples

- Binomial Distribution $\phi(x) = x$
- Discrete Distribution (e$_x$ is unit vector for x) $\phi(x) = e_x$
- Gaussian $\phi(x) = \left(x, \frac{1}{2}xx^\top\right)$
- Poisson (counting measure 1/x!) $\phi(x) = x$
- Dirichlet, Beta, Gamma, Wishart, ...

Binomial Distribution

- Features $\phi(x) = x$
- Domain is {-1, 1}
- Normalization

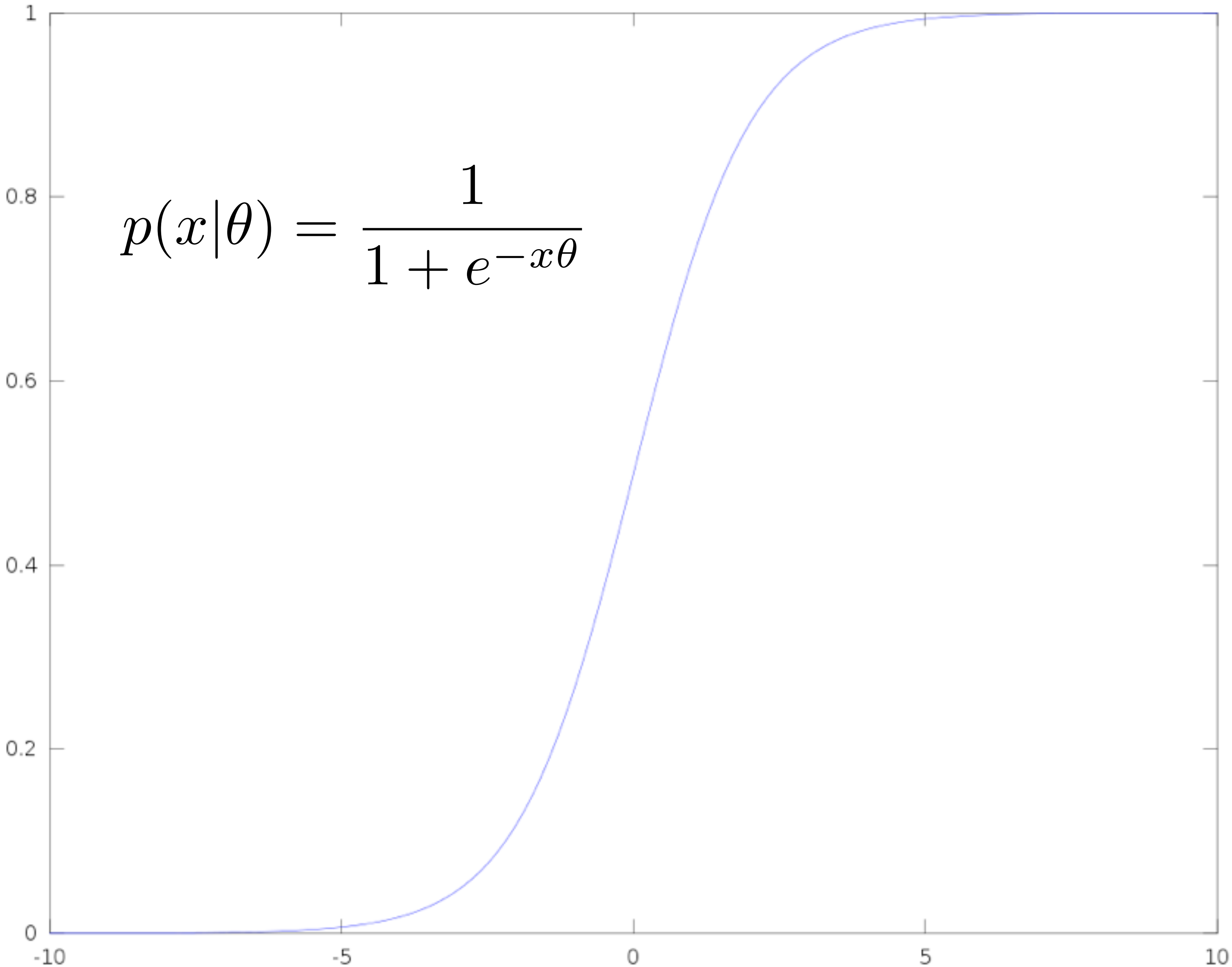
$$g(\theta) = \log\left[e^{-1\cdot\theta} + e^{1\cdot\theta}\right] = \log 2\cosh\theta$$

- Probability

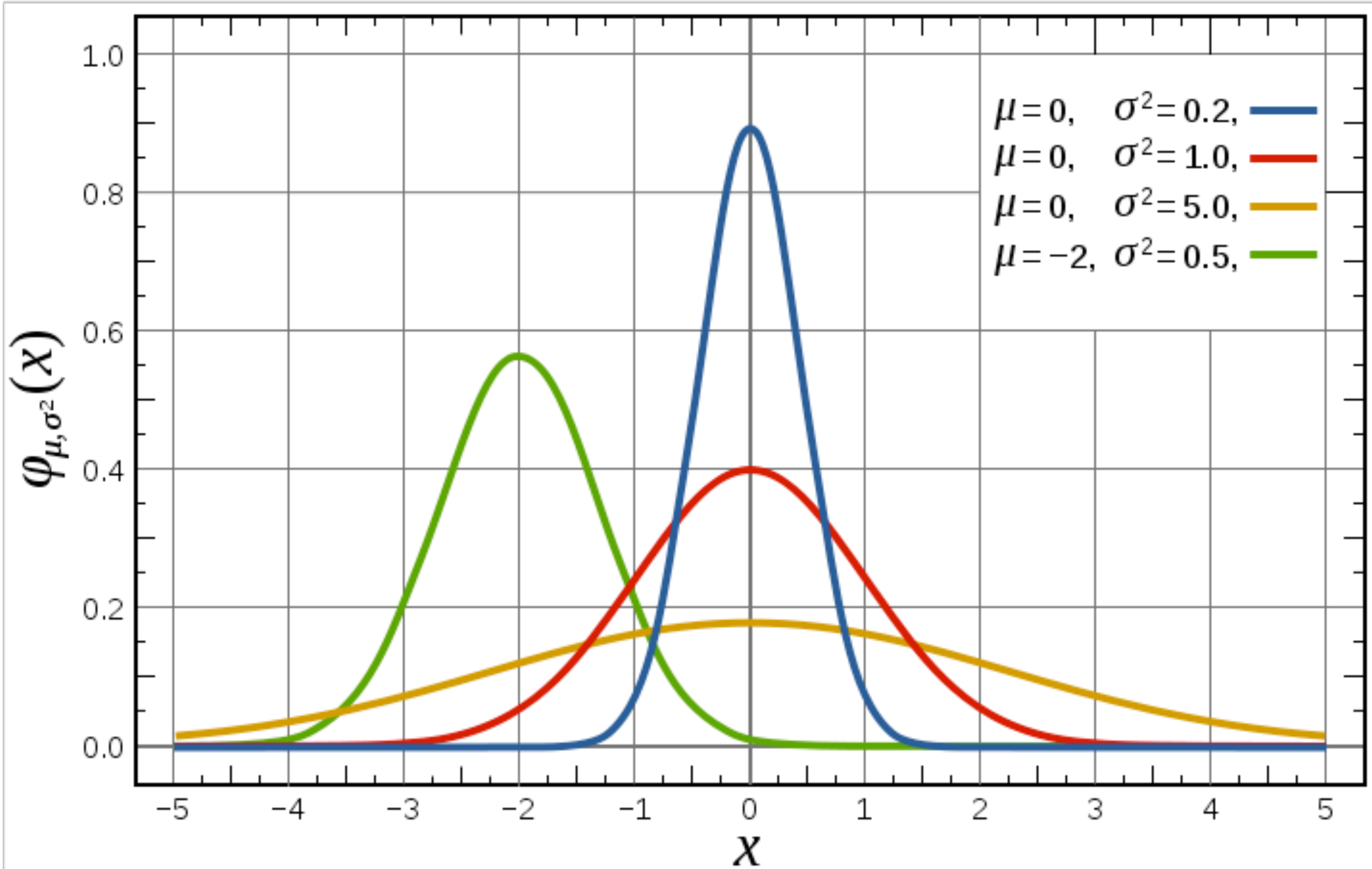
$$p(x|\theta) = \exp(x \cdot \theta - g(\theta)) = \frac{e^{x\theta}}{e^{-\theta} + e^{\theta}} = \frac{1}{1 + e^{-2x\theta}}$$

Logistic function

Logistic function

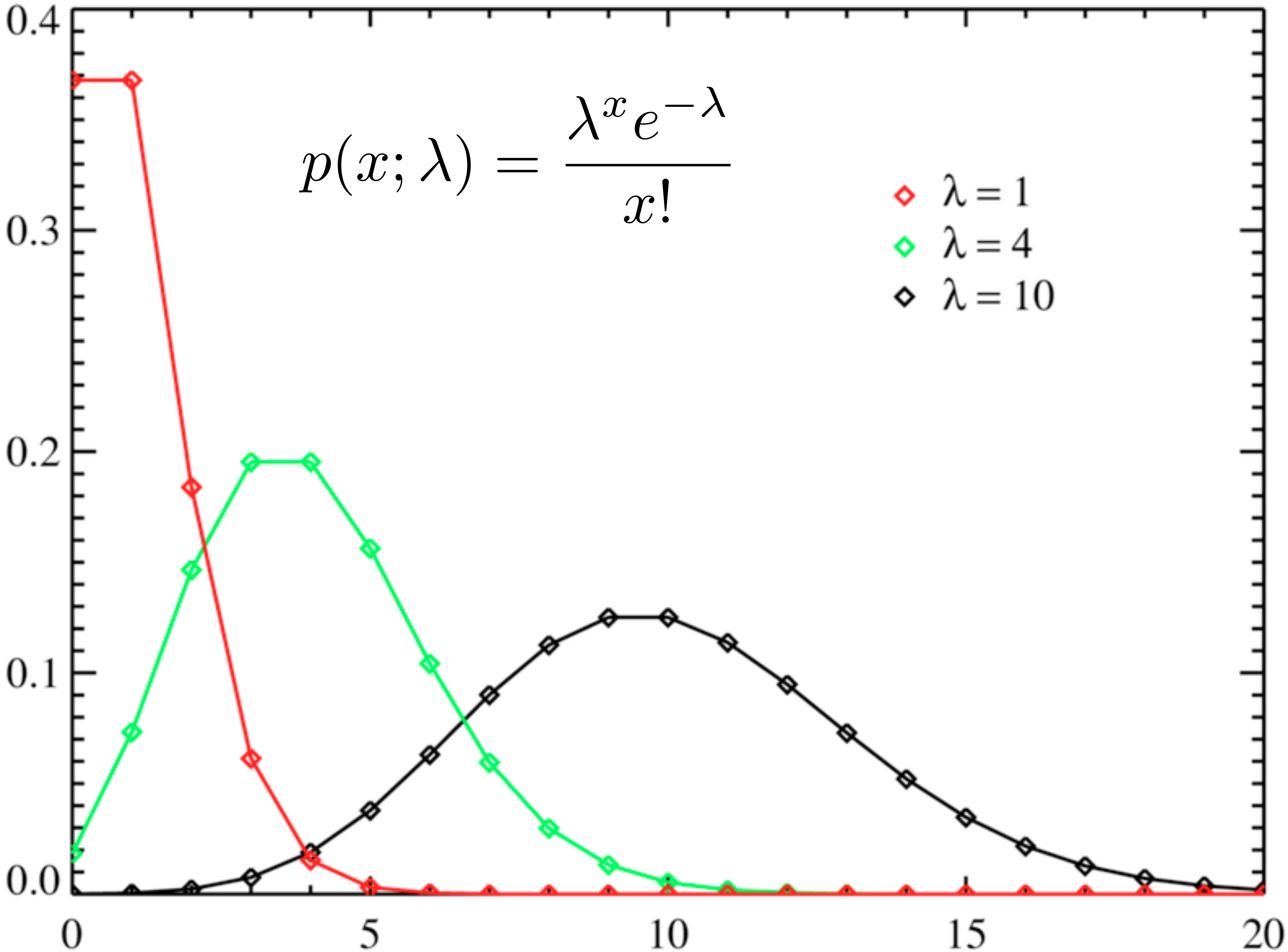


Normal Distribution

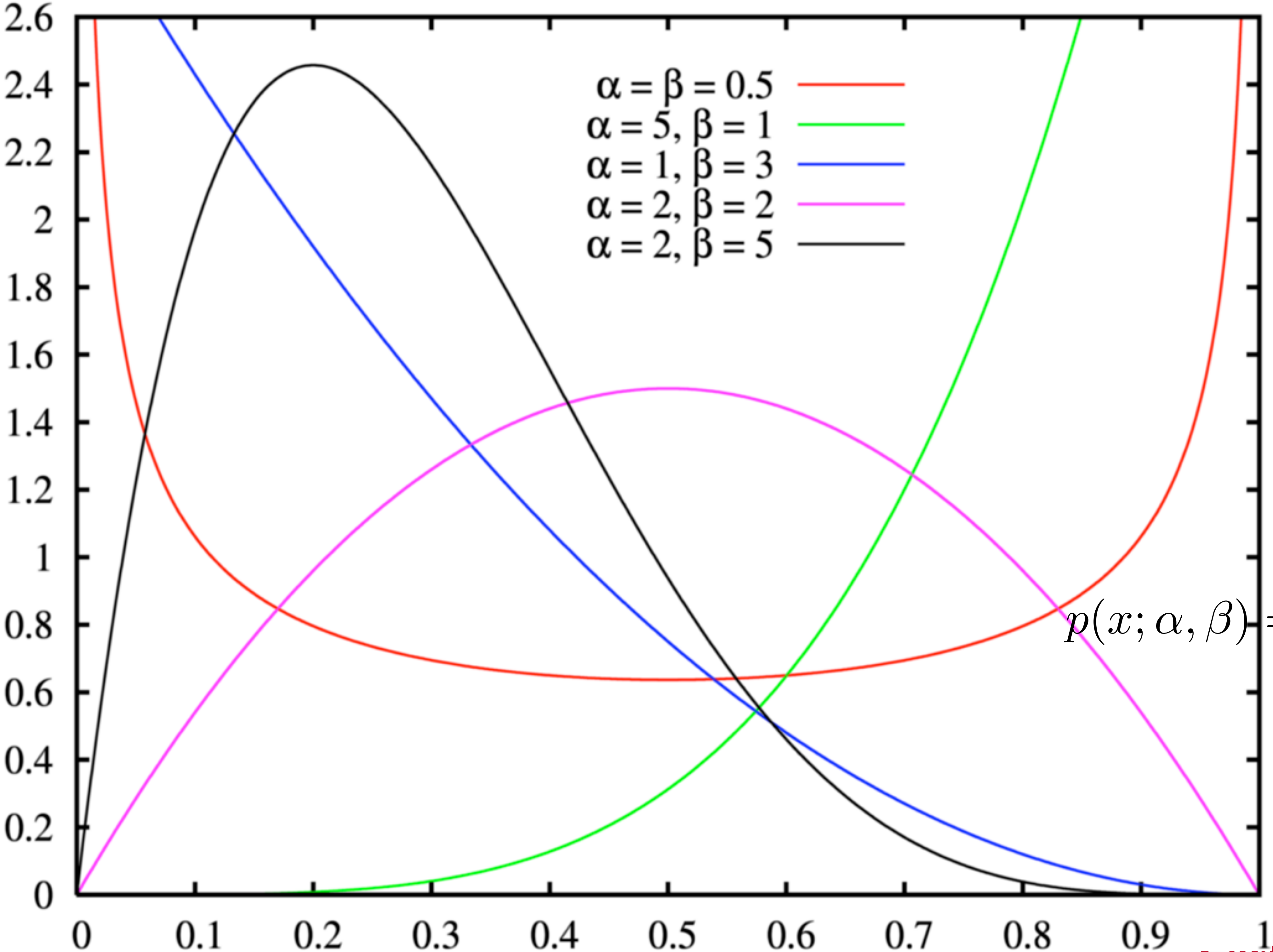


iversity

Poisson Distribution

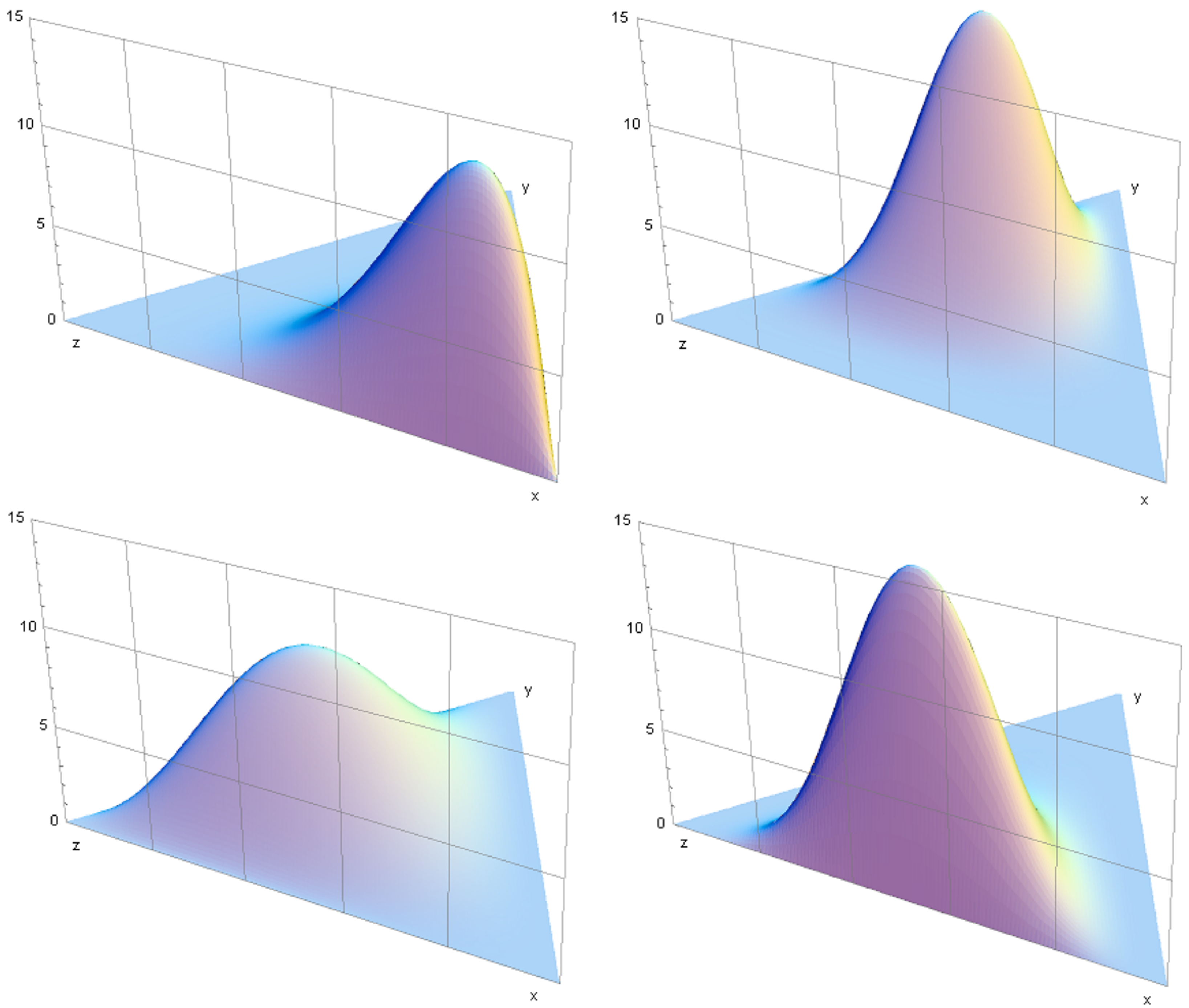


Beta Distribution



$$p(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}$$

Dirichlet Distribution



... this is a distribution over distributions...

Carnegie Mellon University

Inference in Exponential Families

Exponential Family Tree

Maximum Likelihood

Maximum Likelihood

- Negative log-likelihood

$$-\log p(X;\theta) = \sum_{i=1}^{n} g(\theta) - \langle \phi(x_i), \theta \rangle$$

Maximum Likelihood

- Negative log-likelihood

$$-\log p(X;\theta) = \sum_{i=1}^{n} g(\theta) - \langle \phi(x_i), \theta \rangle$$

- Taking derivatives

empirical average

mean

$$-\partial_\theta \log p(X;\theta) = m \left[\mathbf{E}[\phi(x)] - \frac{1}{m} \sum_{i=1}^{n} \phi(x_i) \right]$$

We pick the parameter such that the distribution matches the empirical average.

Conjugate Priors

- Unless we have lots of data estimates are weak
- Usually we have an idea of what to expect
$$p(\theta|X) \propto p(X|\theta) \cdot p(\theta)$$
we might even have 'seen' such data before
- Solution: add 'fake' observations
$$p(\theta) \propto p(X_{\text{fake}}|\theta) \text{ hence } p(\theta|X) \propto p(X|\theta)p(X_{\text{fake}}|\theta) = p(X \cup X_{\text{fake}}|\theta)$$
- Inference (generalized Laplace smoothing)

$$\frac{1}{n}\sum_{i=1}^{n}\phi(x_i) \longrightarrow \frac{1}{n+m}\sum_{i=1}^{n}\phi(x_i) + \frac{m}{n+m}\mu_0$$

fake count

fake mean

Example: Gaussian Estimation

- Sufficient statistics: x, x^2
- Mean and variance given by

$$\mu = \mathbf{E}_x[x] \text{ and } \sigma^2 = \mathbf{E}_x[x^2] - \mathbf{E}_x^2[x]$$

- Maximum Likelihood Estimate

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \text{ and } \sigma^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \hat{\mu}^2$$

- Maximum a Posteriori Estimate

smoother

$$\hat{\mu} = \frac{1}{n+n_0}\sum_{i=1}^{n} x_i \text{ and } \sigma^2 = \frac{1}{n+n_0}\sum_{i=1}^{n} x_i^2 + \frac{n_0}{n+n_0}\mathbf{1} - \hat{\mu}^2$$

smoother

Collapsing

- Conjugate priors

$$p(\theta) \propto p(X_{\text{fake}}|\theta)$$

  Hence we know how to compute normalization

- Prediction

$$p(x|X) = \int p(x|\theta)p(\theta|X)d\theta$$

$$\propto \int p(x|\theta)p(X|\theta)p(X_{\text{fake}}|\theta)d\theta$$

$$= \int p(\{x\} \cup X \cup X_{\text{fake}}|\theta)d\theta$$

(Beta, binomial)
(Dirichlet, multinomial)
(Gamma, Poisson)
(Wishart, Gauss)

look up closed form expansions

http://en.wikipedia.org/wiki/Exponential_family

Conjugate Prior in action

$$m_i = m \cdot [\mu_0]_i$$

$$p(x = i) = \frac{n_i}{n} \longrightarrow p(x = i) = \frac{n_i + m_i}{n + m}$$

Outcome	1	2	3	4	5	6
Counts	3	6	2	1	4	4
MLE	0.15	0.30	0.10	0.05	0.20	0.20
MAP ($m_0 = 6$)	0.15	0.27	0.12	0.08	0.19	0.19
MAP ($m_0 = 100$)	0.16	0.19	0.16	0.15	0.17	0.17

Conjugate Prior in action

- Discrete Distribution

$$m_i = m \cdot [\mu_0]_i$$

$$p(x = i) = \frac{n_i}{n} \longrightarrow p(x = i) = \frac{n_i + m_i}{n + m}$$

- Tossing a dice

Outcome	1	2	3	4	5	6
Counts	3	6	2	1	4	4
MLE	0.15	0.30	0.10	0.05	0.20	0.20
MAP ($m_0 = 6$)	0.15	0.27	0.12	0.08	0.19	0.19
MAP ($m_0 = 100$)	0.16	0.19	0.16	0.15	0.17	0.17

Conjugate Prior in action

- Discrete Distribution

$$m_i = m \cdot [\mu_0]_i$$

$$p(x=i) = \frac{n_i}{n} \longrightarrow p(x=i) = \frac{n_i + m_i}{n + m}$$

- Tossing a dice

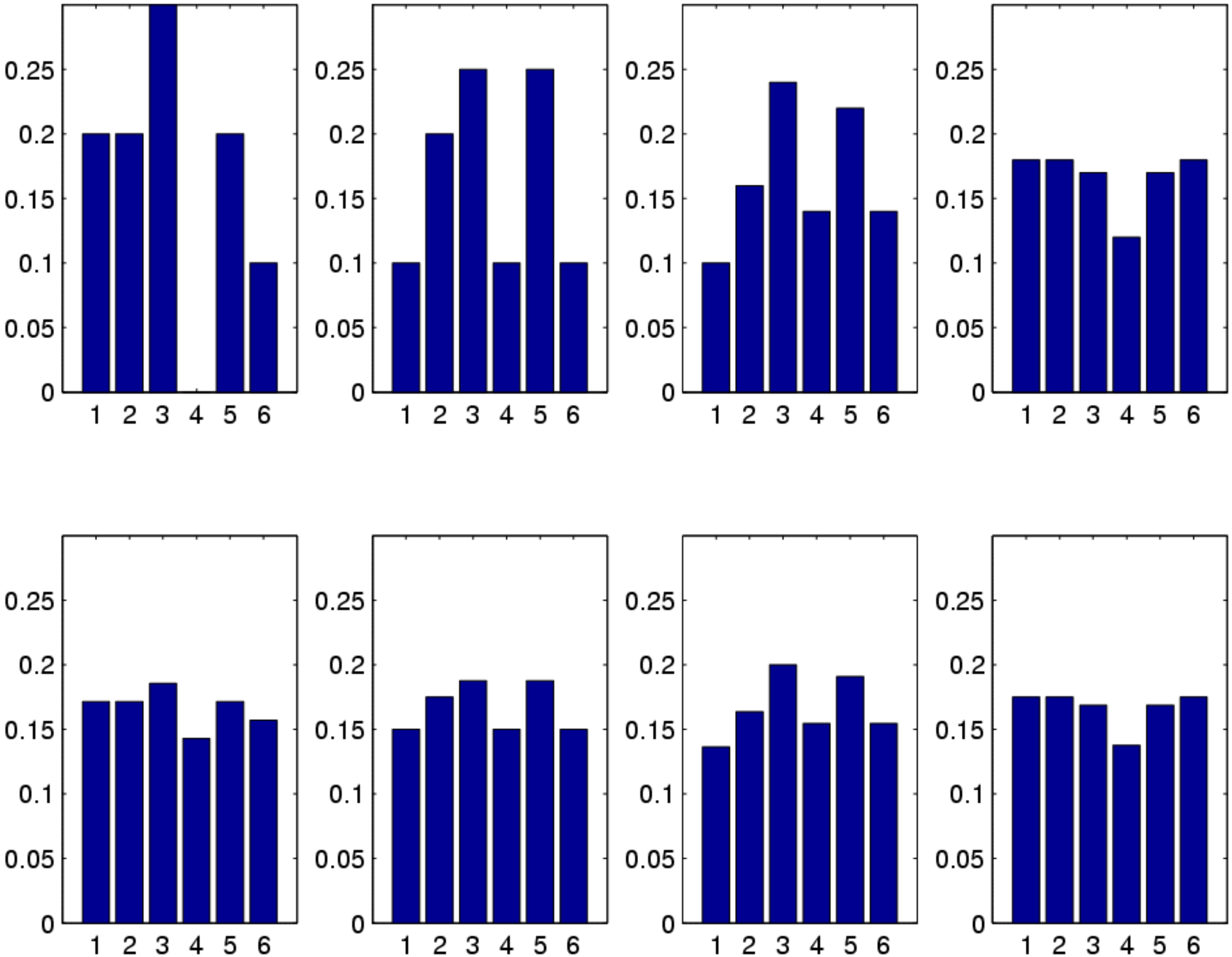
Outcome	1	2	3	4	5	6
Counts	3	6	2	1	4	4
MLE	0.15	0.30	0.10	0.05	0.20	0.20
MAP ($m_0 = 6$)	0.15	0.27	0.12	0.08	0.19	0.19
MAP ($m_0 = 100$)	0.16	0.19	0.16	0.15	0.17	0.17

- Rule of thumb
  need 10 data points (or prior) per parameter

Honest dice

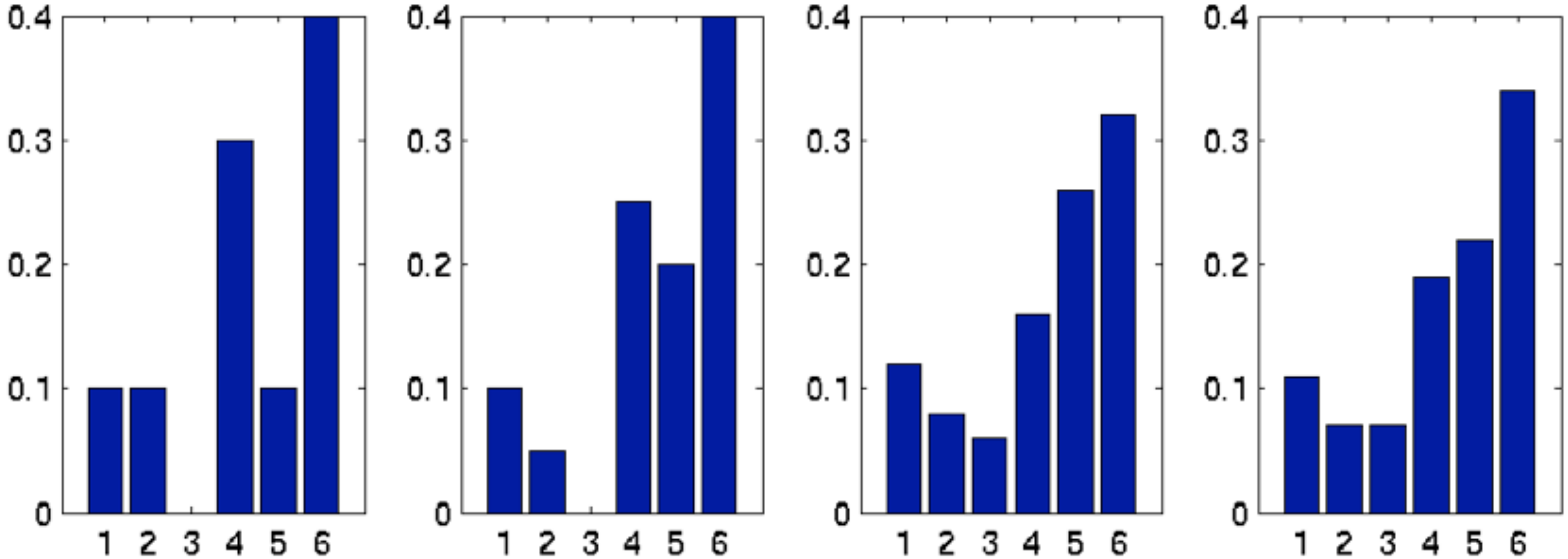
MLE

MAP

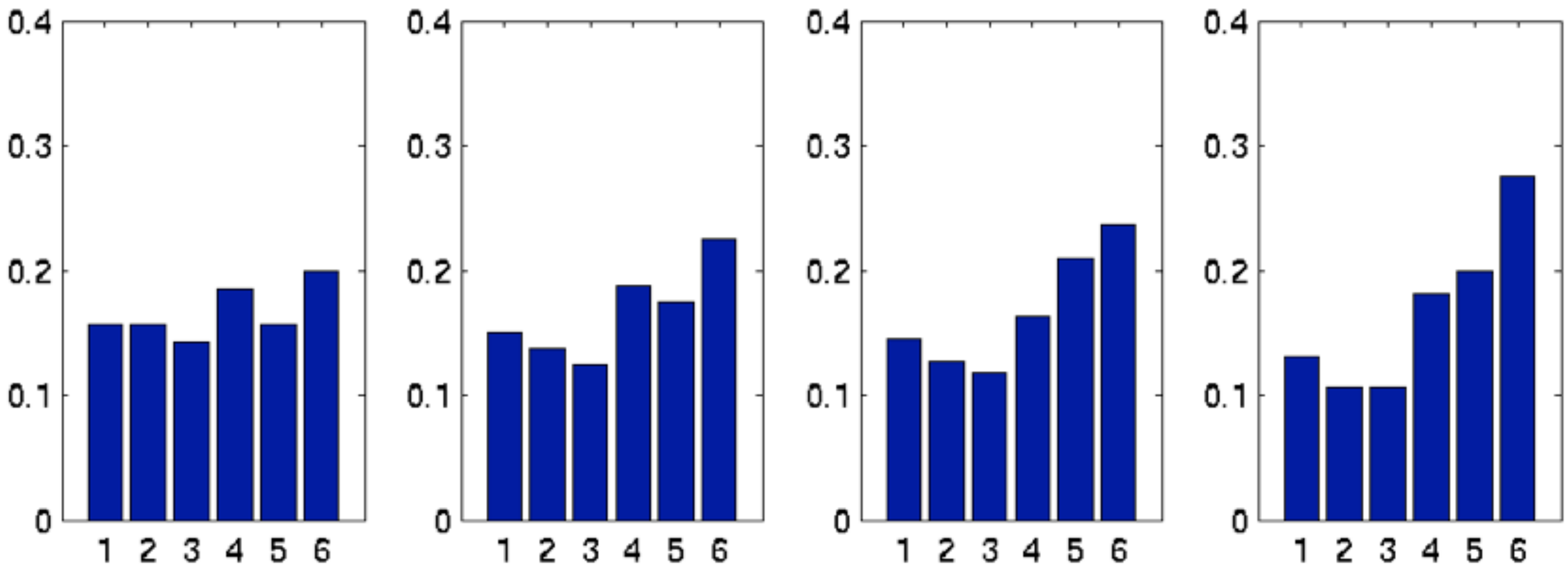


Tainted dice

MLE



MAP



Priors (part deux)

- Parameter smoothing

$$p(\theta) \propto \exp(-\lambda \|\theta\|_1) \text{ or } p(\theta) \propto \exp(-\lambda \|\theta\|_2^2)$$

- Posterior

$$\begin{aligned} p(\theta|x) &\propto \prod_{i=1}^{m} p(x_i|\theta)p(\theta) \\ &\propto \exp\left(\sum_{i=1}^{m} \langle \phi(x_i), \theta \rangle - mg(\theta) - \frac{1}{2\sigma^2} \|\theta\|_2^2\right) \end{aligned}$$

- Convex optimization problem (MAP estimation)

$$\underset{\theta}{\text{minimize}}\, g(\theta) - \left\langle \frac{1}{m} \sum_{i=1}^{m} \phi(x_i), \theta \right\rangle + \frac{1}{2m\sigma^2} \|\theta\|_2^2$$

Further Reading

- Cover tree homepage (paper & code)
  http://hunch.net/~jl/projects/cover_tree/cover_tree.html
- http://doi.acm.org/10.1145/361002.361007 (kd trees, original paper)
- http://www.autonlab.org/autonweb/14665/version/2/part/5/data/moore-tutorial.pdf
  (Andrew Moore's tutorial from his PhD thesis)
- Nadaraya's regression estimator (1964)
  http://dx.doi.org/10.1137/1109020
- Watson's regression estimator (1964)
  http://www.jstor.org/stable/25049340
- Watson-Nadaraya regression package in R
  http://cran.r-project.org/web/packages/np/index.html
- Stone's k-NN regression consistency proof
  http://projecteuclid.org/euclid.aos/1176343886
- Cover and Hart's k-NN classification consistency proof
  http://www-isl.stanford.edu/people/cover/papers/transIT/0021cove.pdf
- Tom Cover's rate analysis for k-NN
  Rates of Convergence for Nearest Neighbor Procedures.
- Sanjoy Dasgupta's analysis for k-NN estimation with selective sampling
  http://cseweb.ucsd.edu/~dasgupta/papers/nnactive.pdf
- Multiedit & Condense (Dasarathy, Sanchez, Townsend)
  http://cgm.cs.mcgill.ca/~godfried/teaching/pr-notes/dasarathy.pdf
- Geometric approximation via core sets
  http://valis.cs.uiuc.edu/~sariel/papers/04/survey/survey.pdf