CS664 Lecture #3: Density estimation in vision

Some slides taken from:

- **David Lowe et al.**

www.cs.ubc.ca/~lowe/425/slides/11-Classifiers.ppt

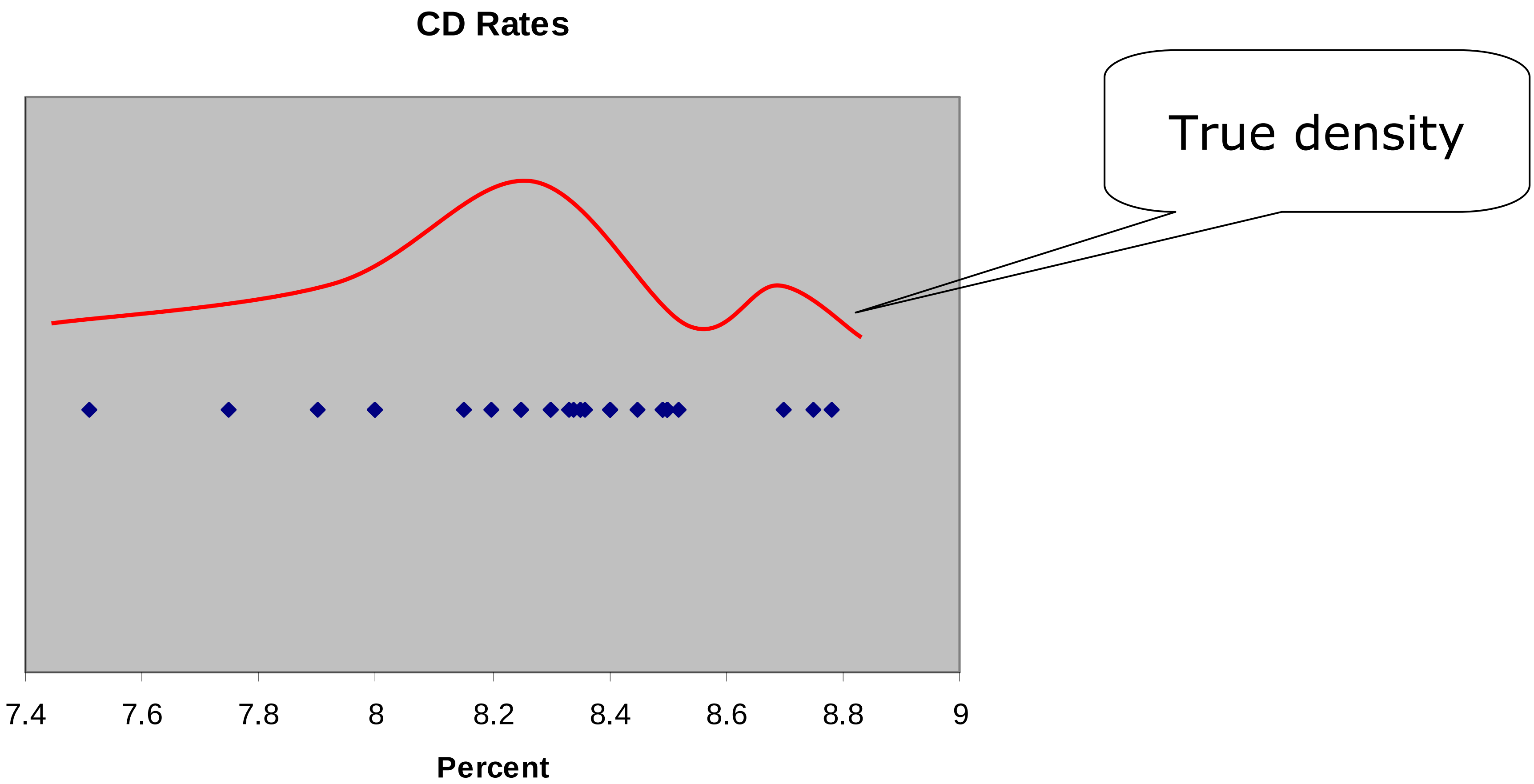
Announcements

- Lecture notes are on the web
- First quiz will be on Thursday
  - Coverage through today's lecture
- We will use CMS, the Course Management System
  - We'll be setting this up soon
- Guest lecture a week from Tuesday by Bryan Kressler
  - We'll have a few of these over the semester

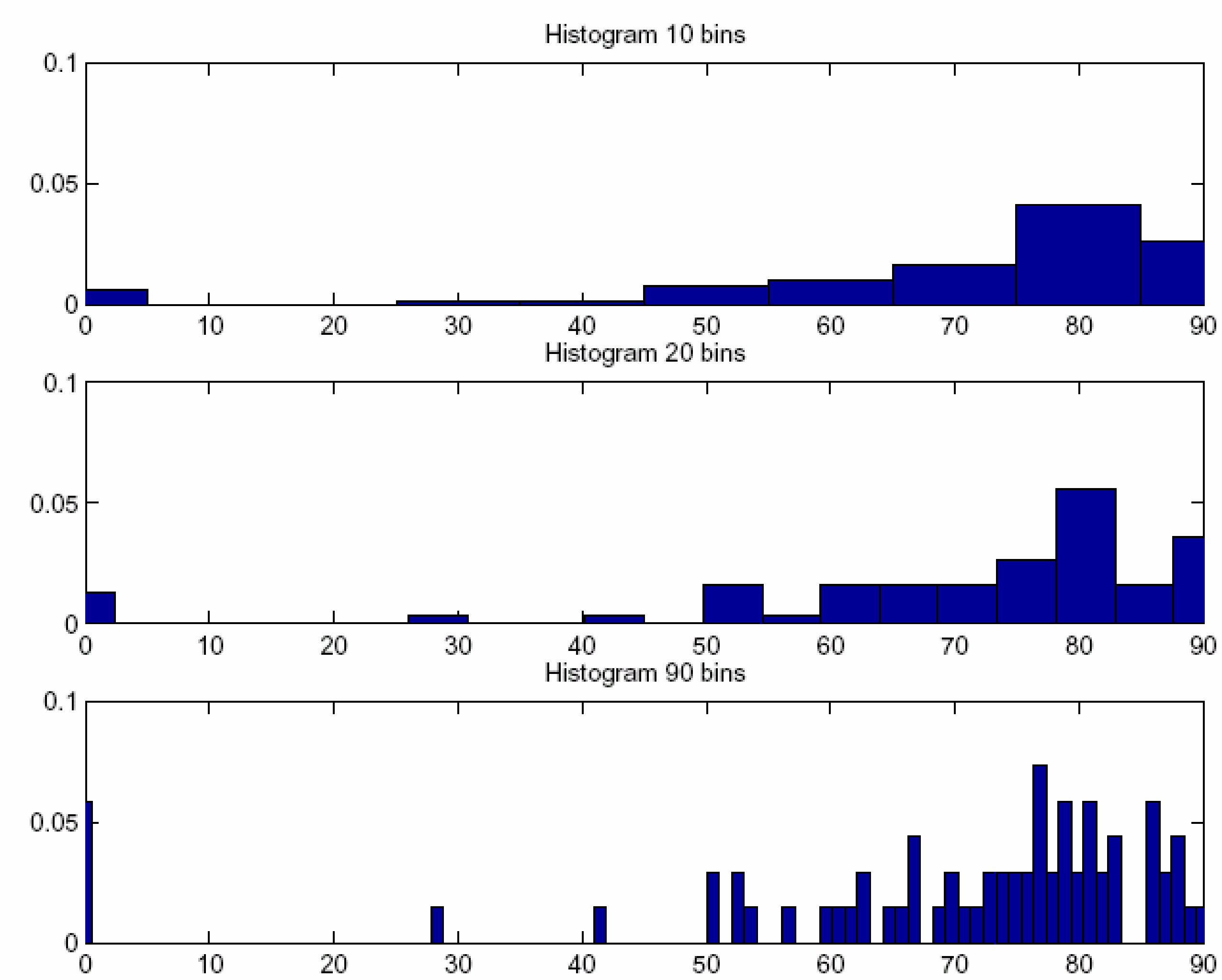
Last lecture we saw:

- Trigrams are an elegant way to generate text
  - Density estimation and sampling
- Same basic idea used by Efros & Leung to perform texture synthesis
  - Some surprisingly good results
- Parametric density estimation
  - i.e., fitting the data with a Gaussian
- Non-parametric density estimation
  - i.e., using a histogram

Density estimation



Histogram representation



Histogram-based estimates

- You can use a variety of fitting techniques to produce a curve from a histogram
  - Lines, polynomials, splines, etc.
  - Also called regression/function approximation
  - Normalize to make this a density
- If you know quite a bit about the underlying density you can compute a good bin size
  - But that's rarely realistic in vision
  - And defeats the whole purpose of the non-parametric approach

Nearest-neighbor estimate

- To estimate the density, count the number of nearby data points
  - Like histogramming with sliding bins
  - Avoid bin-placement artifacts

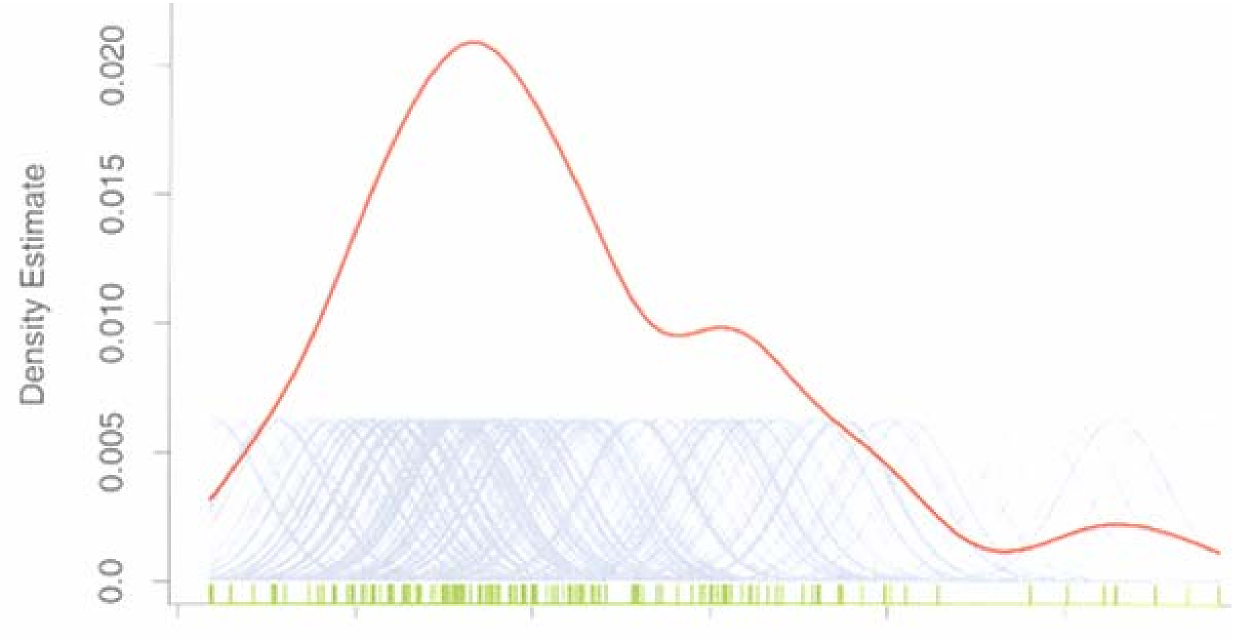
$$\widehat{p}(x) = \frac{\#\{\, x_i \mid \|x_i - x\| \leq \epsilon \,\}}{n}$$

  - We can fix ϵ and compute this quantity, or we can fix the quantity and compute ϵ

Parzen estimation

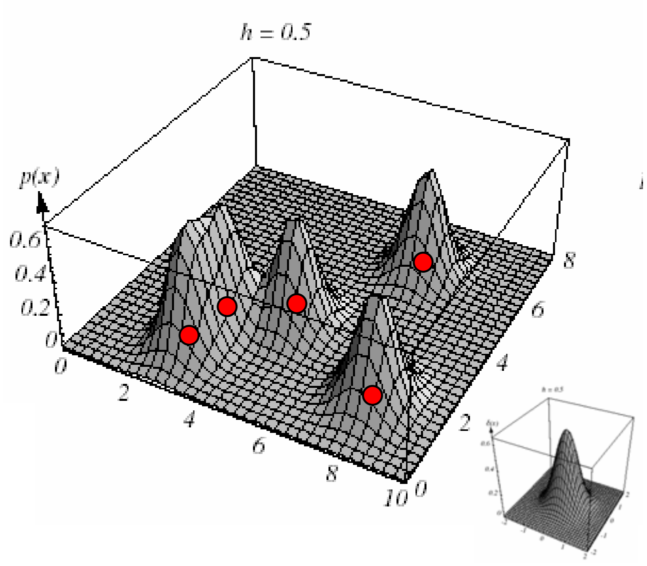
- Each observed datapoint increases our estimate of the probability nearby
  - Simplest case: raise the probability uniformly within a fixed radius
    - Place a fixed-height “box” at each datapoint, add them up to get the density estimate
  - This is nearest neighbor with fixed ϵ
- More generally, you can use some slowly decreasing function (such as a Gaussian)
  - Called the *kernel*

Parzen example



from Hastie *et al.*

Importance of scale



Relationship to Efros/Leung

- Can we store histograms of 11-by-11 patches?
  - How many such patches are there?
  - 256^{121} is a lot ($> 10^{240}$)
- They don't quite do density estimation
  - The method is procedural
    - And slightly ad hoc
  - But the effect is close to Parzen estimation
    - With some kind of unusual kernel
- This would be a natural follow-up paper

Computing local modes

- Often we don't need the entire density
  - Vision often has very high #/dimensions
- Suppose that we could find the nearest local maximum (mode)
  - Doing this repeatedly gives a simple clustering scheme
  - There is an elegant way to do this
  - One of the more successful methods in vision

Mean shift algorithm

- Non-parametric method to compute the nearest mode of a distribution
  - Density increases as we get near "center"

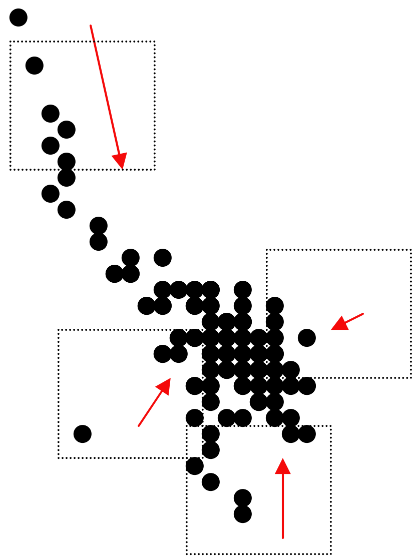
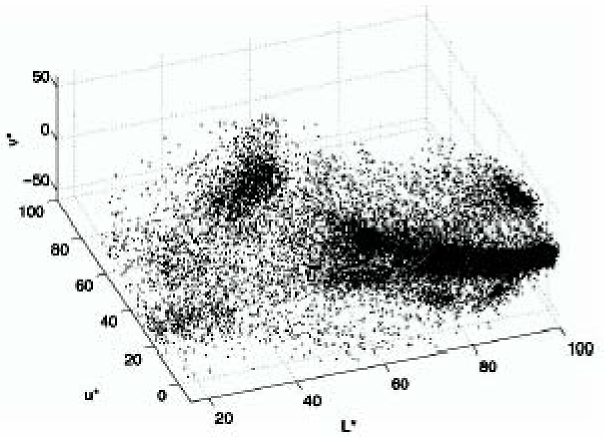
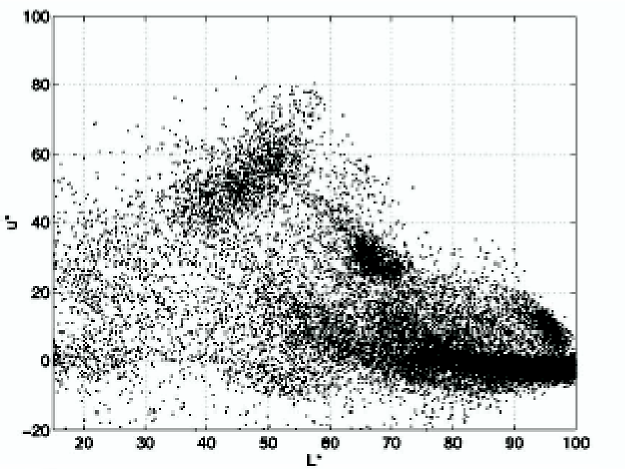


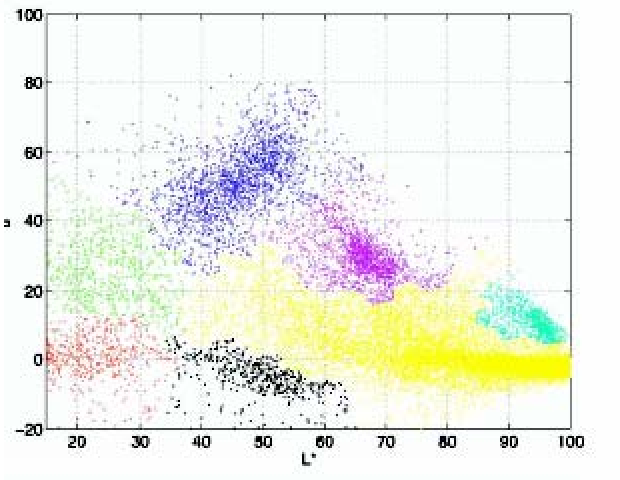
Image and histogram



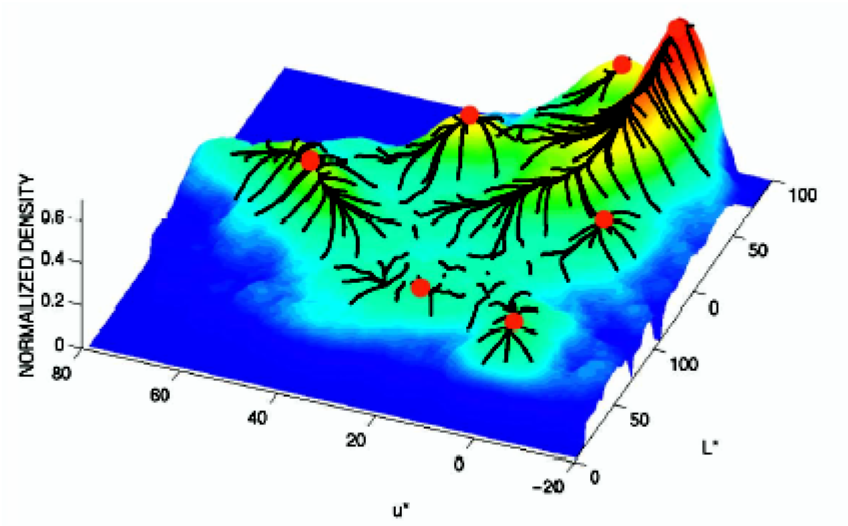
Local modes



100
80
60
40
20
0
−20
20
30
40
50
60
70
80
90
100
L*
u*



100
80
60
40
20
0
−20
20
30
40
50
60
70
80
90
100
L*

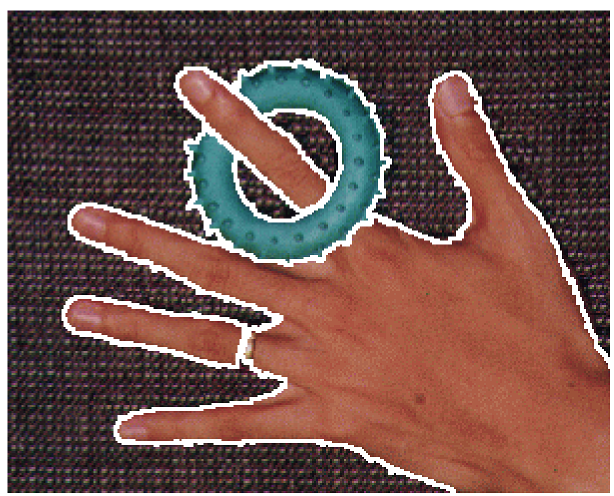


NORMALIZED DENSITY
0.6
0.4
0.2
0
80
60
40
20
0
−20
u*
100
50
0
100
50
0
L*

Mean shift segmentations







http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

Back to density estimation

- Many density estimates for same data
  - E.g., different Parzen windows, or mean
  - Is there a natural sense in which one estimate might be “optimal”?
- Maximum likelihood principle
  - If a particular hypothesized density were correct, it would have some probability of resulting in the data we observed
  - Pick the hypothesis with the largest likelihood

ML estimate of the mean

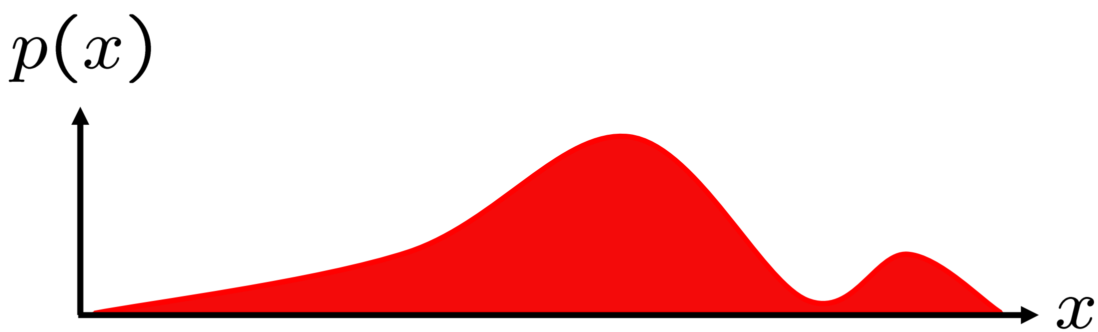
- Consider parametric density estimation with a Gaussian
  - What choice of mean μ and width σ maximizes the likelihood?
- Need to be a little more precise
  - What does it mean to say that a particular density actually generated the data we saw?

What is a density?

- Consider an arbitrary function p such that

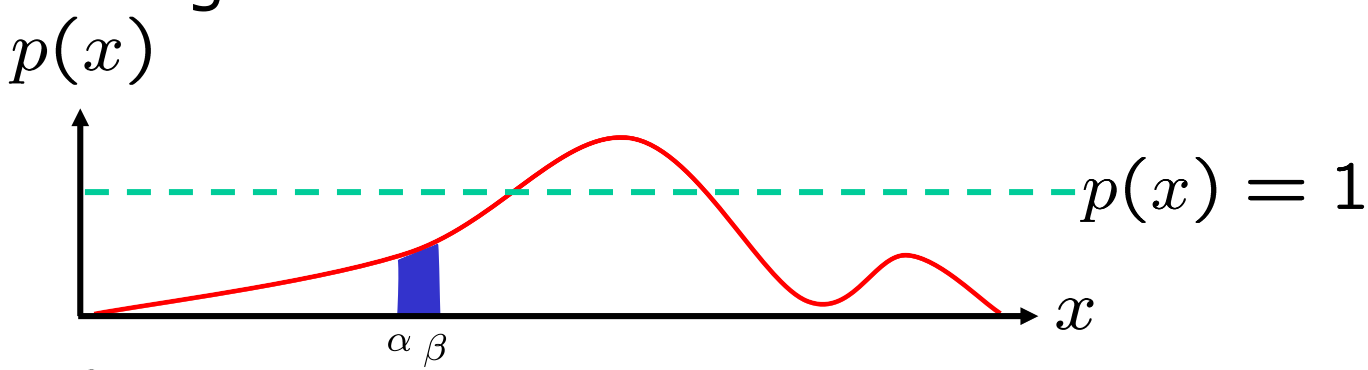
$$p(x) \geq 0$$

  – Can view it as a *probability density function*
    - The PDF for a real-valued random variable
    - If we asked ∞ banks, what frequency of CD rates would we get?



Interpreting densities

- The value of the PDF p at x is **not** the probability we would get rate x
  - Which is always zero (think about it!)
  - Instead, p gives the probability of getting a rate in a given interval



$$\int_{\alpha}^{\beta} p(x)dx = \text{Pr}[\alpha \leq \text{Rate} \leq \beta]$$

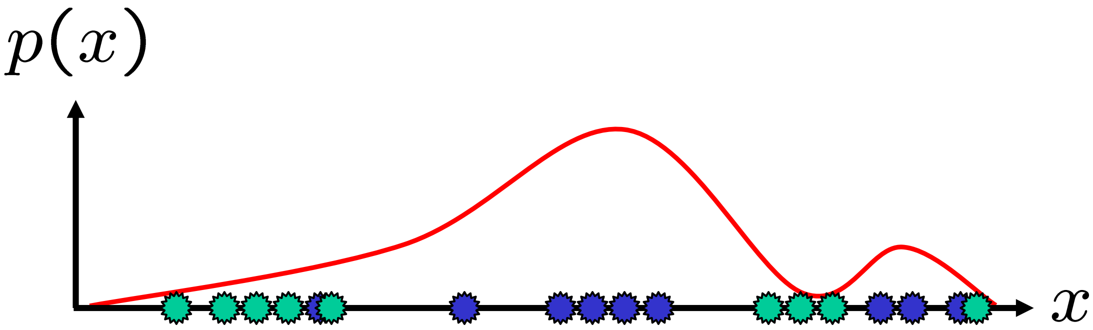
Discrete case is easier

- If the values of the random variable are discrete, things are simpler
  - Instead of a PDF you have a probability mass function (PMF)
  - I.e., a histogram whose entries sum to 1
    - No bucket has a value greater than 1
- This is the true relative frequencies (i.e., what we would get in the limit)
  - What is the bin size of the PMF histogram?

Sampling from a PDF

- Suppose we call up a number of banks and get their CD rates
  - This generates our *sample* (data set)
  - How does this relate to the true PDF?
- It simplifies life considerably to assume:
  - All the banks generate their rates from the same PDF (identical distributions)
  - There is no effect between the rate you get from one bank and another (independence)

Sample likelihood



Definition of likelihood

- Intuition: the true PDF should not make the sample (data) you saw a "fluke"
  - It's possible that the coin is fair even though you saw 10^6 heads in a row…
- The likelihood of a hypothesis is the probability that it would have resulted in the data you saw
  - Think of the data as fixed, and try to chose among the possible PDF's