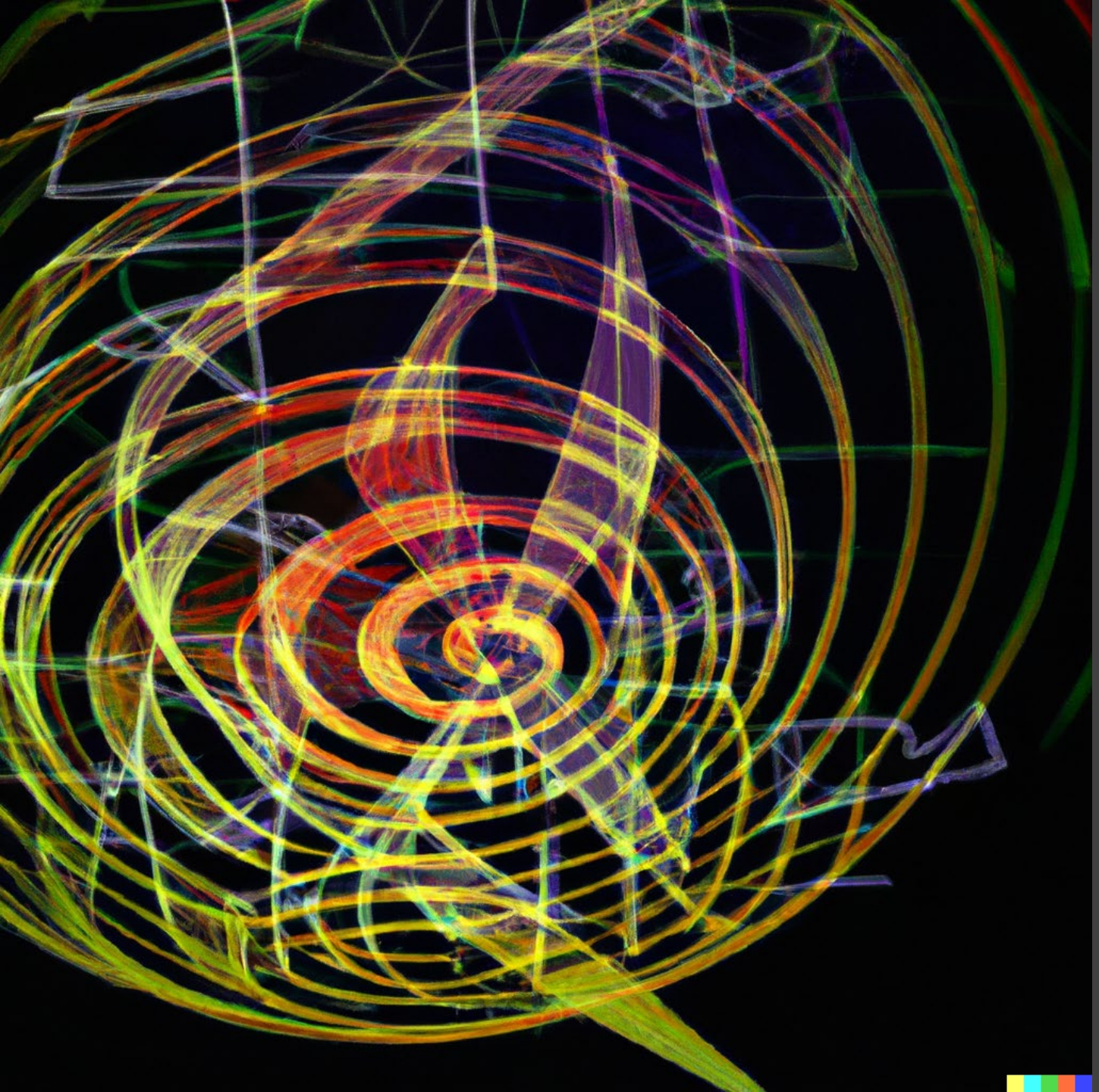


PDF Estimation

Applied Machine Learning
Derek Hoiem

Dall-E

Last class: EM

- True or False
  - The EM algorithm is a method for maximum likelihood estimation in the presence of missing or incomplete data.
  - The EM algorithm guarantees convergence to the global maximum of the likelihood function for any given dataset.
  - The E-step of the EM algorithm computes the maximum likelihood estimate of the parameters given the observed data.
  - In the EM algorithm, the likelihood of the observed data increases after each iteration of the algorithm.

True, False, False, True

This class – PDF Estimation

- We often want to estimate P(x) for some observation x
  - For classification, e.g. we want P(x | y)
  - For outlier/anomaly detection, e.g. we want to know if P(x) is small
  - For prediction, we want to know which answer or action is most likely
  - Model dependencies and mutual information
- For discrete variables, this just involves counting, but continuous variables are more tricky
- We consider models and methods for estimating P(x)

PDF = probability density function

Basic rules of PDFs

1. $\int p(x) = 1$ (but $p(x)$ can be greater than 1)
2. $p(x) \geq 0 \; \forall x$

That's it!

A common implicit assumption of pdfs is smoothness, i.e. $p(x + \epsilon) \approx p(x)$ for some small ϵ

1D PDFs

- E.g.
  - x is a temperature
  - Intensity
  - Sensor reading
  - Height of a person
  - Number of dolphins in a body of water

- x could be continuous or have too many possible values to count frequency for each
  - E.g. how many people are in NYC at a given day in the year

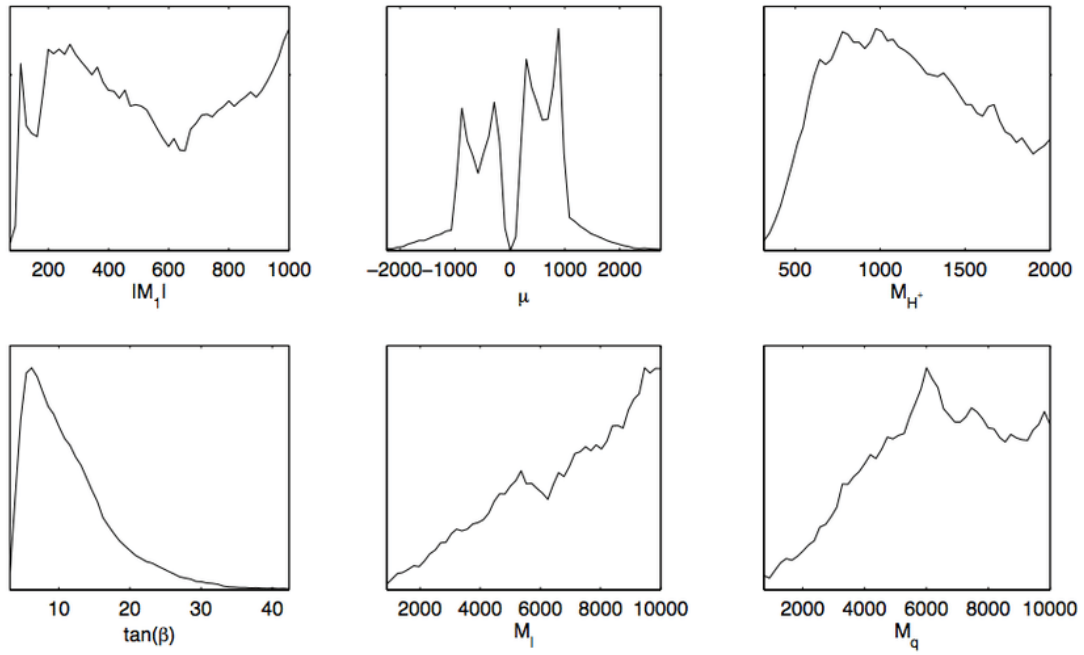


Fig src

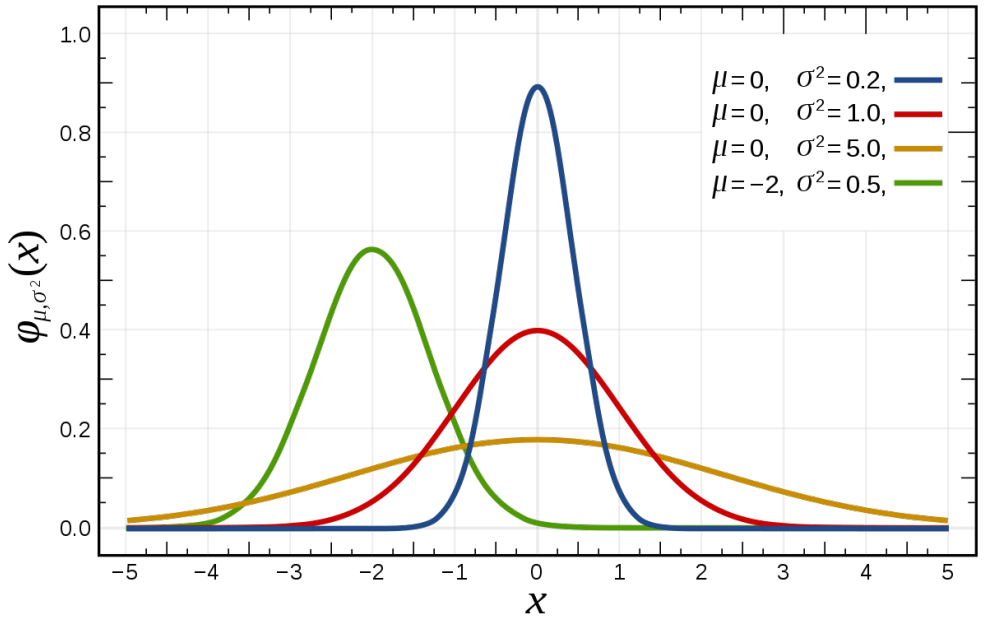
Methods to estimate PDFs

	Parametric Models	**Semi-Parametric**	**Non-Parametric**
Description	Assumes a fixed form for density	Can fit a broad range of functions with limited parameters	Can fit any distribution
Examples	Gaussian, exponential	Mixture of Gaussians	Discretization, kernel density estimation
Good when	Model is able to approximately fit the distribution	Low dimensional or smooth distribution	1-D data
Not good when	Model cannot approximate the distribution	Distribution is not smooth, challenging in high dimensions	Data is high dimensional

Gaussian distribution

- Easy to estimate parameters
- Symmetric, unimodal
- Products of Gaussian variables is Gaussian
- Sums of Gaussian random variables are Gaussian
- The sum of many random variables is approximately Gaussian (Central Limit Theorem)
  - Often used as a noise model
- Light tail (values become small quickly away from the mean)
- Examples of distributions fit well by Gaussian
  - Heights of adult males
  - Image intensity noise
  - Weights of newborn babies

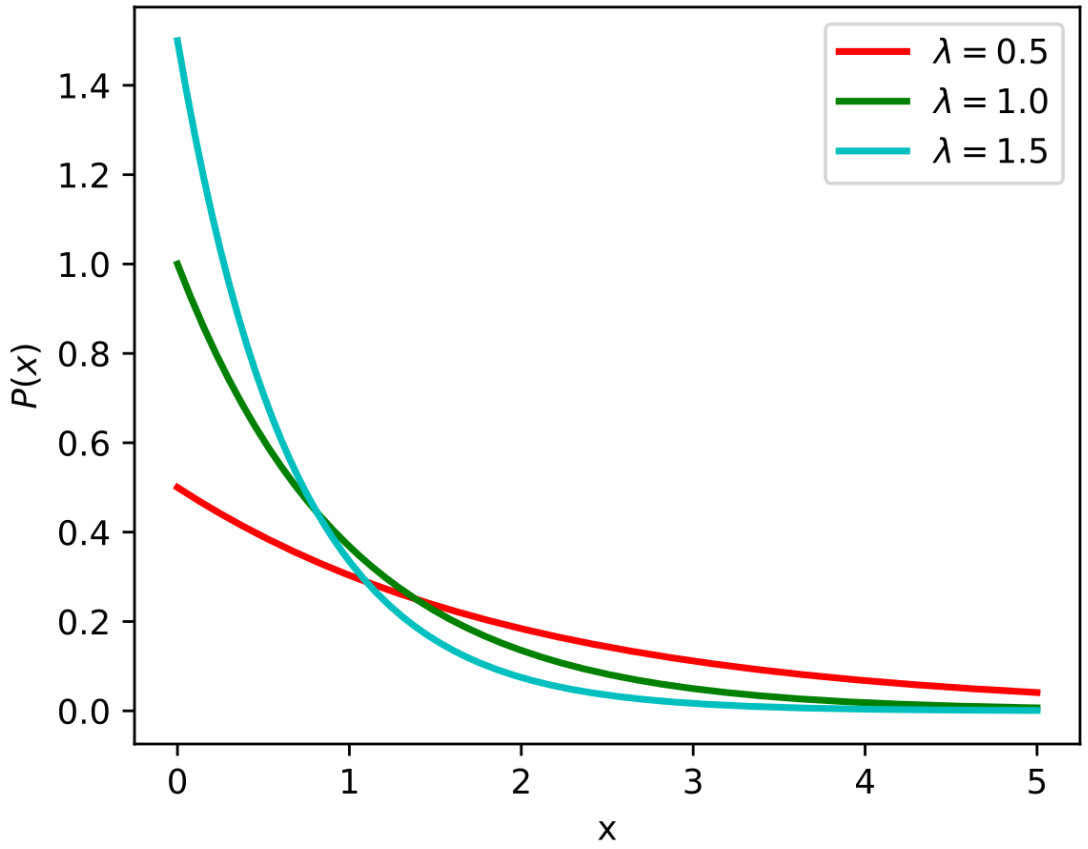
$$p(x_n \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_n - \mu)^2}{2\sigma^2}\right)$$



Exponential distribution

- Mean=std=$1/\lambda$
- Positive
- Heavy-tailed (significantly non-zero values far from mean)
- Examples of variables that have exponential distributions
  - When events occur with uniform likelihood, the time to next event, e.g. to emit radioactive particles, when customer next enters a shop
  - Frequency of words in a collection of documents
  - Amount of money spent per customer at supermarket

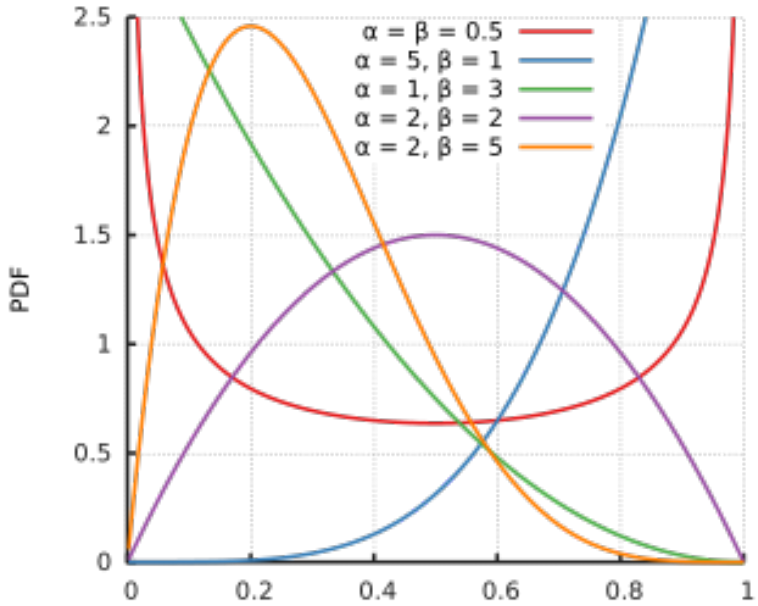
$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$



Beta distribution

- Range is 0 to 1
- Lots of different shapes, depending on parameters
- Used when a variable has a finite range of possible values

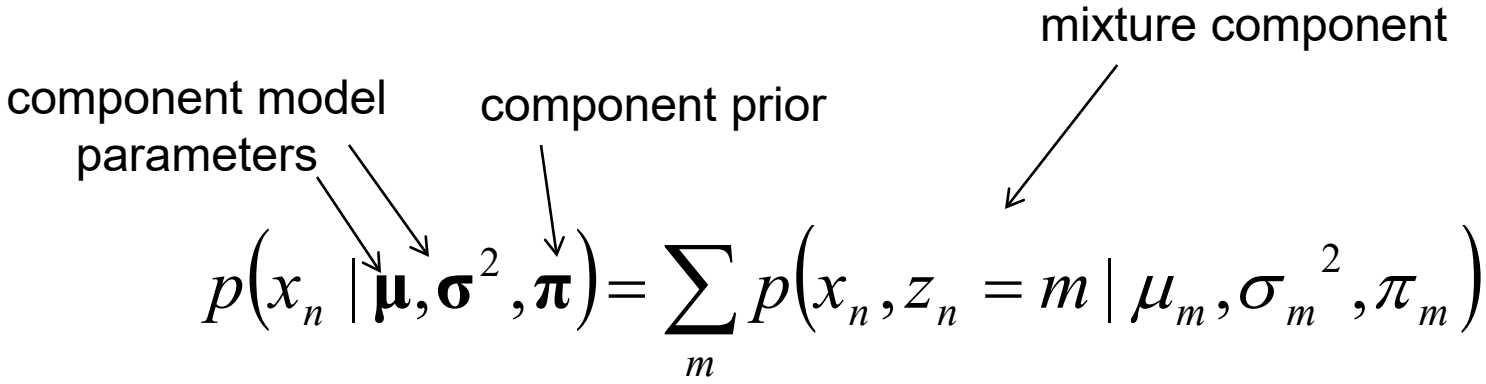
$$f(x;\alpha,\beta) = \text{constant} \cdot x^{\alpha-1}(1-x)^{\beta-1}$$



Estimating parameters of models

- Usually computable in closed form
- Easy to look up the MLE estimates (parameter values that maximize data likelihood)

Mixture of Gaussians



$$p\left(x_n \mid \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\pi}\right) = \sum_m p\left(x_n, z_n = m \mid \mu_m, {\sigma_m}^2, \pi_m\right)$$

$$p\left(x_n, z_n = m \mid \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\pi}\right) = p\left(x_n, z_n = m \mid \mu_m, {\sigma_m}^2, \pi_m\right)$$

$$= p\left(x_n \mid \mu_m, {\sigma_m}^2\right) p\left(z_n = m \mid \pi_m\right)$$

$$= \frac{1}{\sqrt{2\pi {\sigma_m}^2}} \exp\left(-\frac{\left(x_n - \mu_m\right)^2}{2{\sigma_m}^2}\right) \cdot \pi_m$$

Mixture of Gaussians

- With enough components, can represent any probability density function
  - Widely used as general purpose pdf estimator

- Bias toward smooth density functions

Estimate Mixture of Gaussians with EM Algorithm

1. Initialize parameters

2. Compute likelihood of hidden variables for current parameters

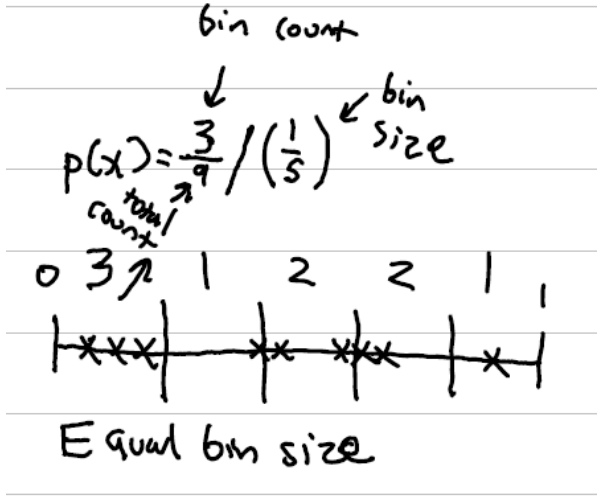
$$\alpha_{nm} = p(z_n = m \mid x_n, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2(t)}, \boldsymbol{\pi}^{(t)})$$

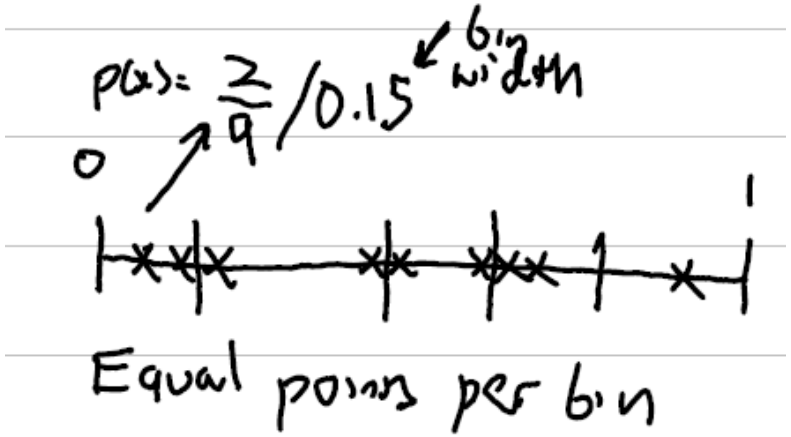
3. Estimate new parameters for each model, weighted by likelihood

$$\hat{\mu}_m^{(t+1)} = \frac{1}{\sum_n \alpha_{nm}} \sum_n \alpha_{nm} x_n \qquad \hat{\sigma}_m^{2(t+1)} = \frac{1}{\sum_n \alpha_{nm}} \sum_n \alpha_{nm} \left(x_n - \hat{\mu}_m\right)^2 \qquad \hat{\pi}_m^{(t+1)} = \frac{\sum_n \alpha_{nm}}{N}$$

“Histograms”, Discretized Functions

1. Convert a continuous value to discrete values
   - Divide into equal sized bins
   - Divide data into equal sized chunks and put bin boundaries around it
   - Perform K-means
2. Count occurrence within each bin
3. $p(x) = (bin\ count)/(total\ count)/(bin\ size)$

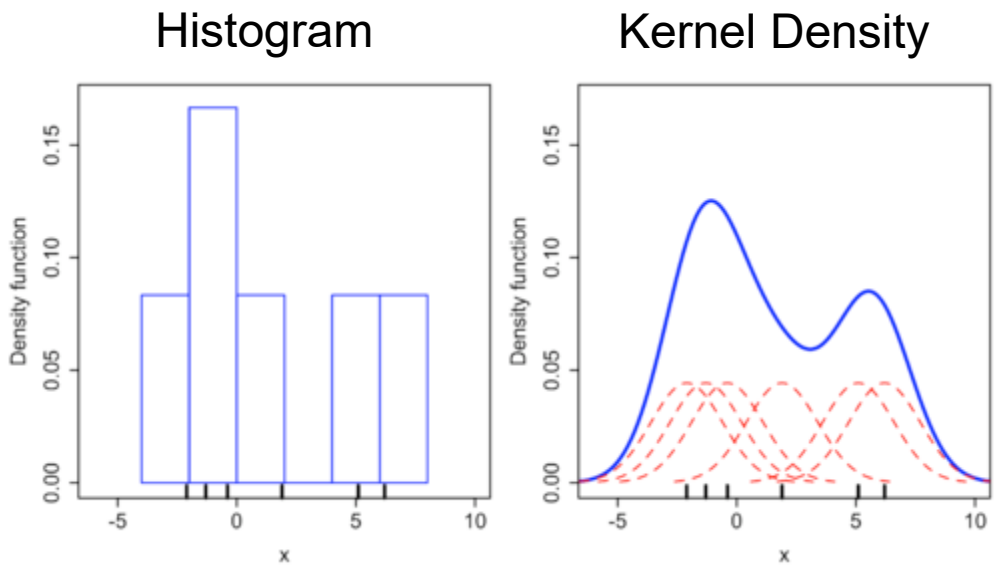




Kernel density estimation

- p(x) is weighted count of points near x, divided by number of points
- Potential for fine-granularity like histograms, but no defined boundaries
- Bandwidth controls smoothness
  - One rule of thumb for Gaussian kernel $h = \left(\frac{4\hat{\sigma}^5}{3n}\right)^{\frac{1}{5}} \approx 1.06\,\hat{\sigma}\,n^{-1/5}$
- Can be computed efficiently in combination with fine discretizations

$$\widehat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$



How to select parameters and evaluate

- Hyperparameters
  - Histogram: num bins
  - Kernel Density: bandwidth
  - MoG: num components
  - Also may set priors on bin count or variance

- To select hyperparameters:
  - Split data into a train and val set
  - For each candidate parameter:
    - Fit to train
    - Compute average log likelihood on val
    - Record best
  - Fit with best parameter on train+val

- To compare different models
  - Evaluate average log likelihood on a held out test set

Let’s experiment

Methods

- Gaussian
- Histogram, equal size bins
- Histogram, equal data bins
- KDE
- Mixture of Gaussians

Data

- Student exam scores
- Image intensities
- Government payroll data
- Penguins
- Simple distributions

https://colab.research.google.com/drive/1H4_jS1oxiOxZkfvh5w5KEWF2zFs4axty?usp=sharing

https://data.ct.gov/Government/State-Employee-Payroll-Data-Calendar-Year-2015-thr/virr-yb6n/data

Evaluation

	# Times Best	# Times Close to Best	Pros	Cons
Gaussian	1		Works for its own category, but even then KDE also works well	Doesn't work if data is not gaussian distributed
Hist, equal width	2-3	Two thirds of time	Always reasonable as long as you search over bin width	Sometimes too sharp, discrete and not smooth
Hist, equal data	1	Almost always	Works reliably with few samples, but gets peaky and not smooth with a lot of data	
Kernel density	4+	Almost always	Smooth, always works, stable	Doesn't necessarily find modes
Mix of Gauss	3	4	Good for salary/midterm with multiple modes	Sometimes overfits and creates weird squiggles

https://colab.research.google.com/drive/1H4_jS1oxiOxZkfvh5w5KEWF2zFs4axty?usp=sharing

Estimating probability for more than 1 dimension

- Estimate for each dimension and assume dimensions are independent

  $P(x_1, x_2, x_3) = P(x_1)P(x_2)P(x_3)$

- Use a multidimensional model
  - Gaussian
  - Mixture of Gaussian

- Project to lower dimension/manifold (e.g. PCA) and then estimate

- Non-linear compression, e.g. with auto-encoder (e.g. with deep network)

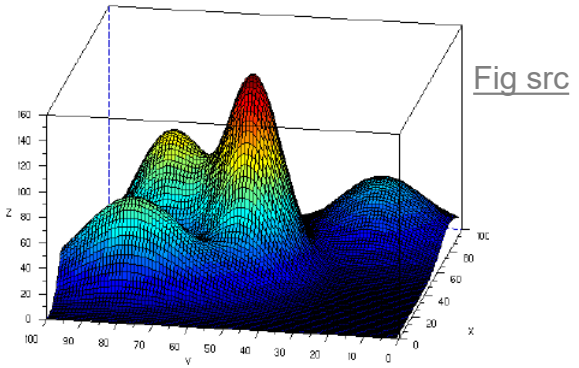
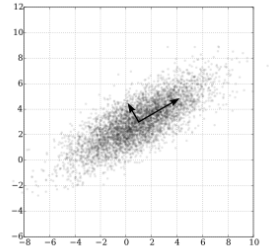
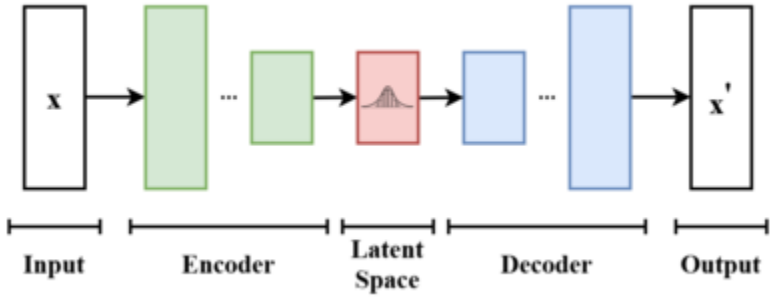


Fig src





Next class

- PCA and projection methods