Computational Data Analysis

Machine Learning

Yao Xie, Ph.D.

Associate Professor

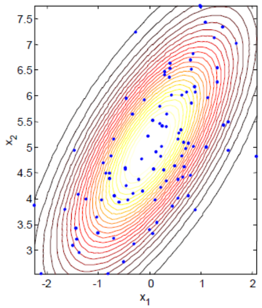
Harold R. and Mary Anne Nash Early Career Professor

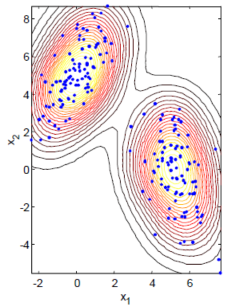
H. Milton Stewart School of Industrial and Systems Engineering

Density Estimation

Why do we need density estimation?

- Learn more about the "shape" of the data cloud



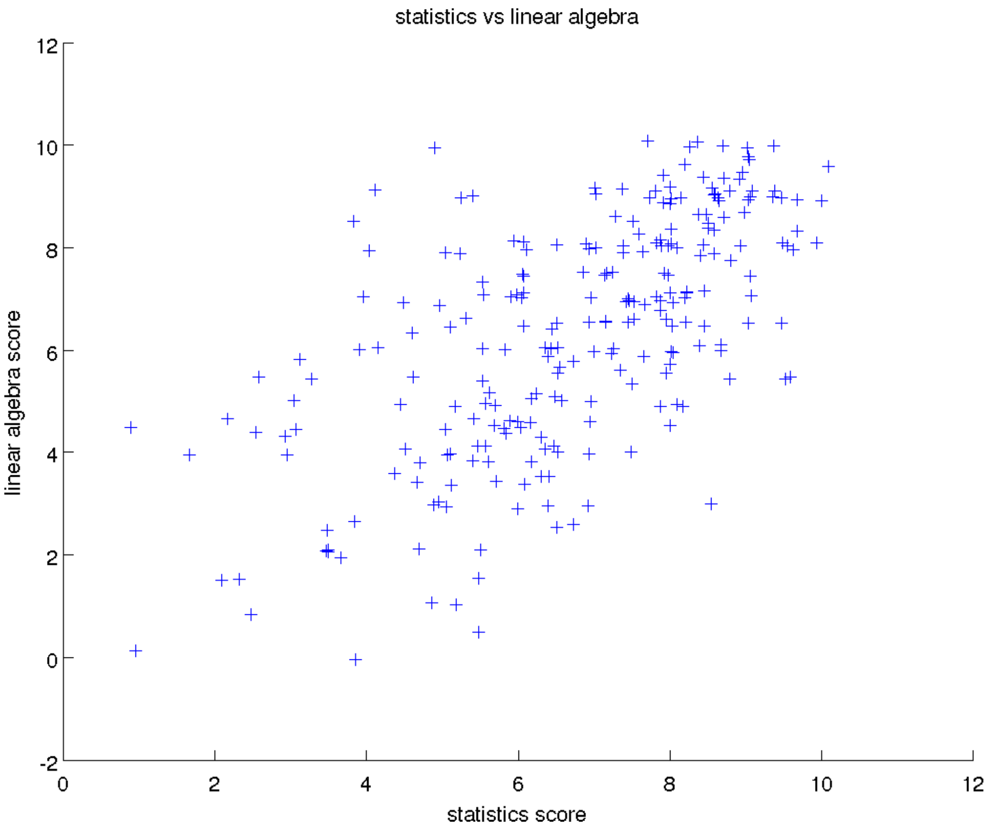


- Assess the likelihood of seeing a particular data point
  - Is this a typical data point? (high density value)
  - Is this an abnormal data point / outlier? (low density value)

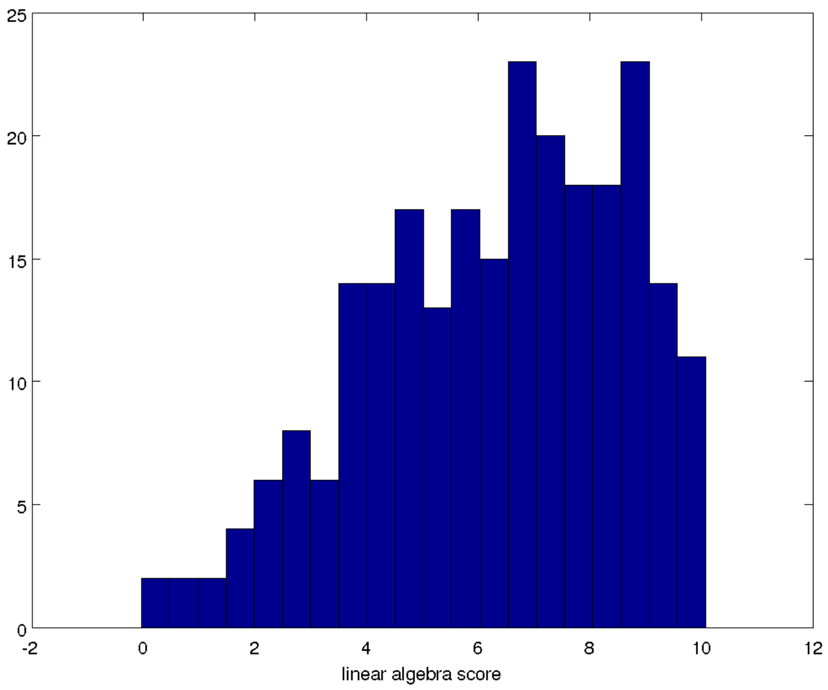
- Building block for more sophisticated learning algorithms
  - Classification, regression, graphical models …
  - A simple recommendation system

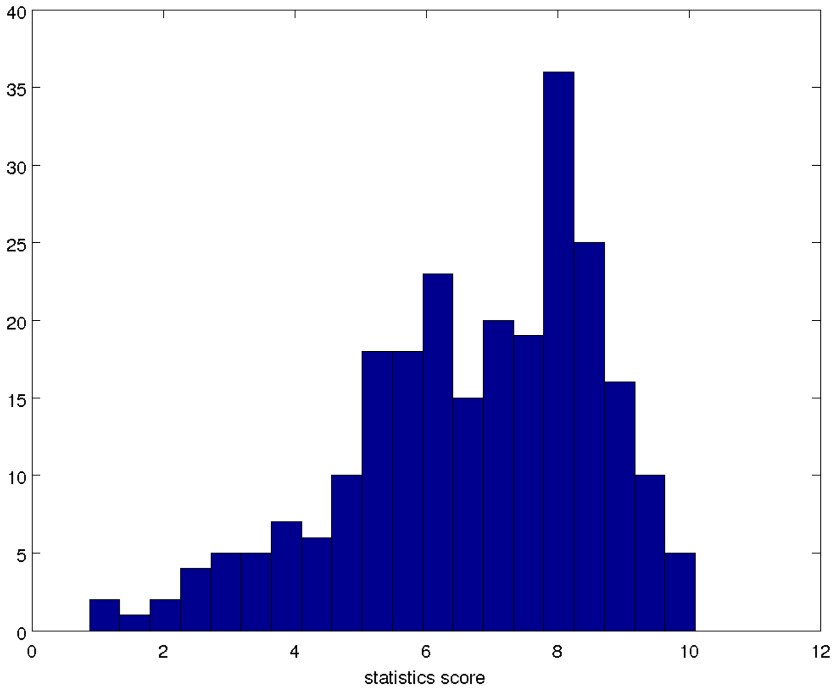
http://www.datawrangling.org/python-montage-code-for-displaying-arrays/

Example: test scores in a class

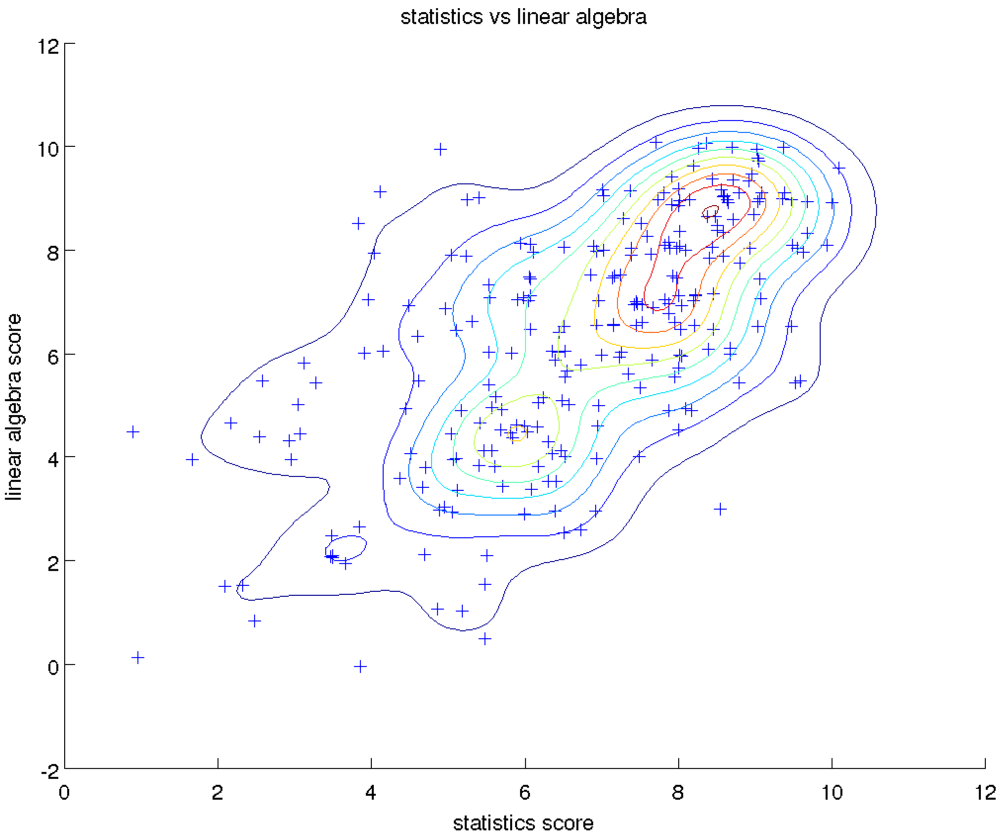


Example: test scores in a class





Example: background test scores (cont.)



Parametric models

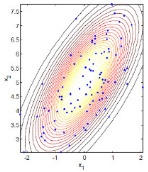
- Models which can be described by a fixed number of parameters

- Discrete case: eg. Bernoulli distribution

$$P(x|\theta) = \theta^x (1-\theta)^{1-x}$$

one parameter, $\theta \in [0,1]$, which generate a family of models, $\mathcal{F} = \{P(x|\theta) \mid \theta \in [0,1]\}$,

- Continuous case: eg. Gaussian distribution in R^n



$$p(x|\mu, \Sigma) = \frac{1}{|\Sigma|^{\frac{1}{2}}(2\pi)^{\frac{n}{2}}} exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

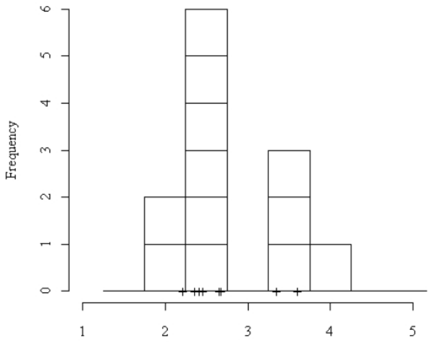
Two sets of parameters $\{\mu, \Sigma\}$, which again generate a family of models, $\mathcal{F} = \{p(x|\mu, \Sigma) \mid \mu \in R^n, \Sigma \in R^{n\times n}\ and\ PSD\}$,

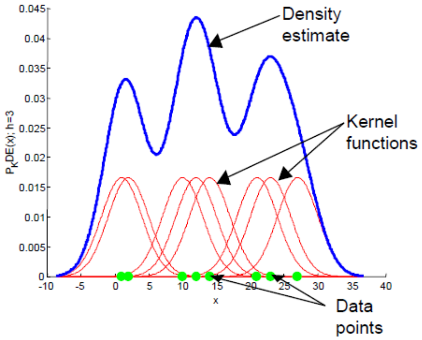
Nonparametric models

- What are nonparametric models?
  - “nonparametric” does not mean there are no parameters
  - can not be described by a fixed number of parameters
  - one can think of there are many many parameters

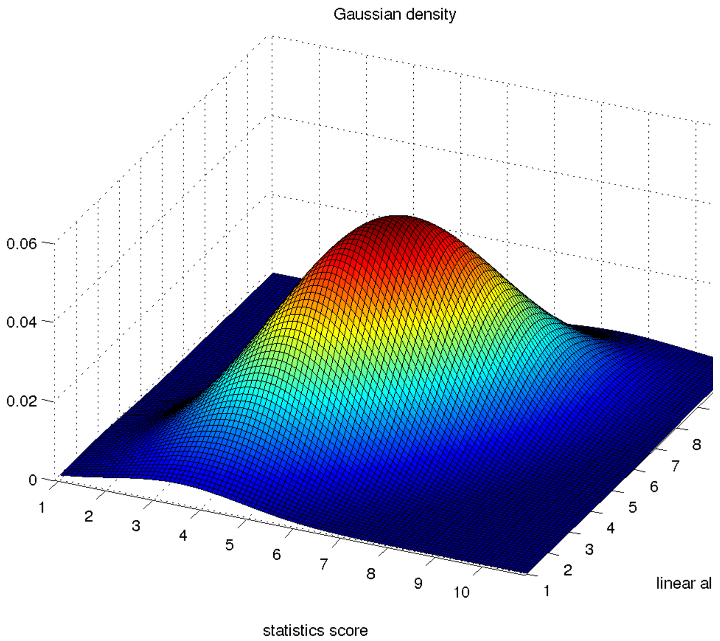
- Eg. Histogram

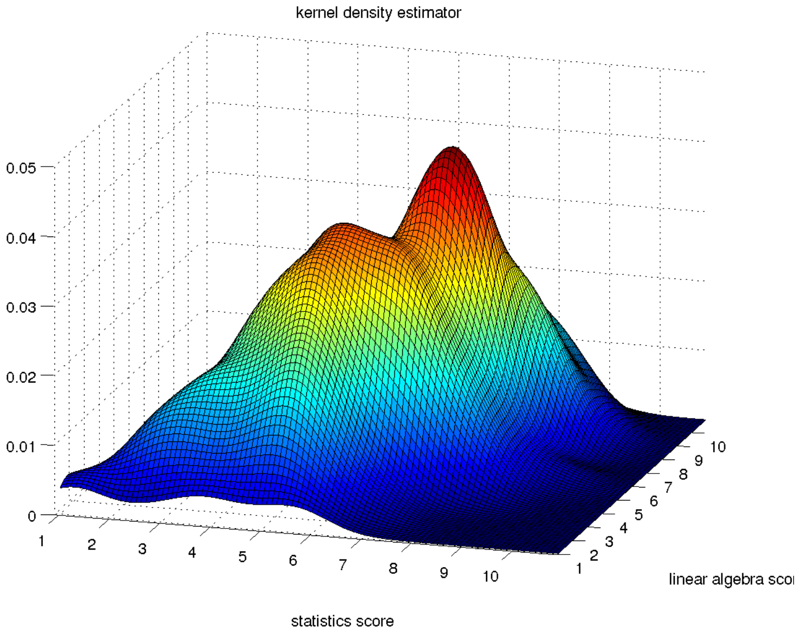
- Eg. Kernel density estimator



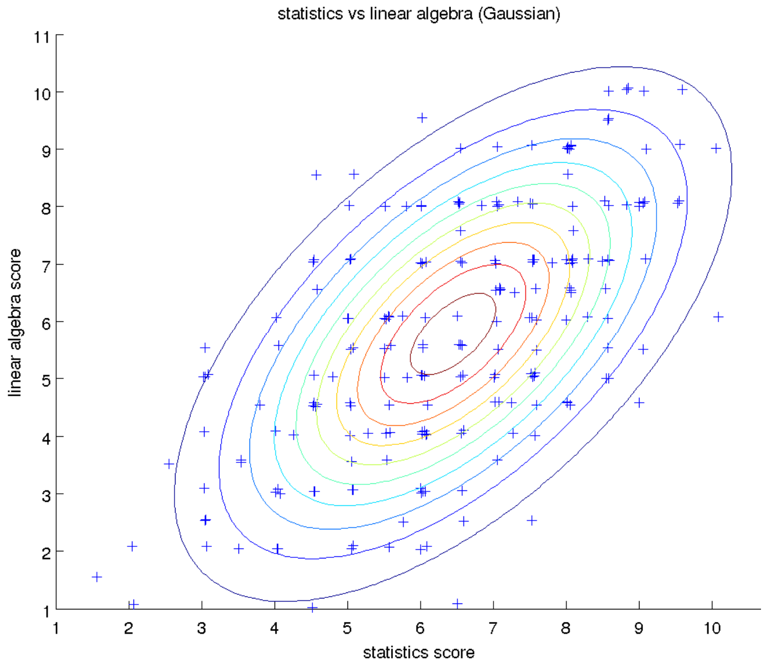


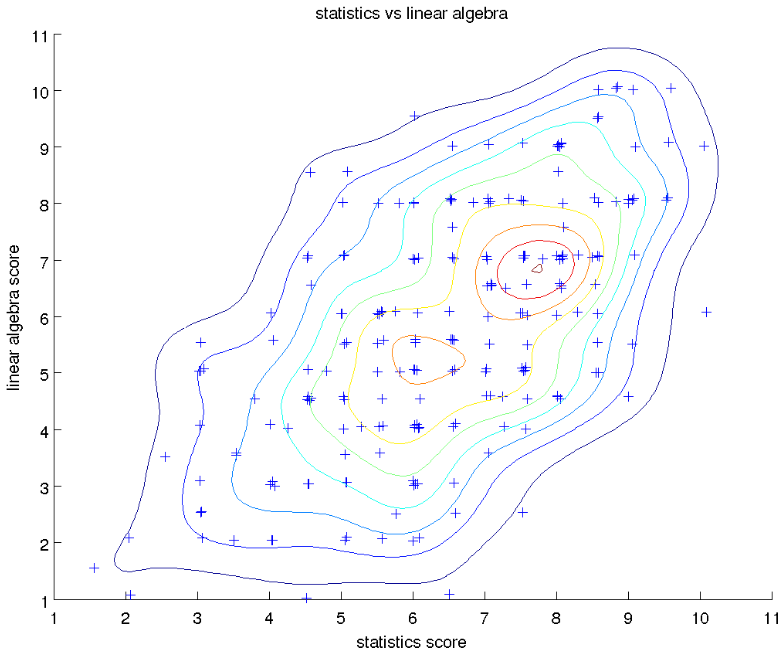
Parametric vs. nonparametric





Parametric vs. nonparametric





Estimation of parametric models

- A very popular estimator is the maximum likelihood estimator (MLE), which is simple and has good statistical properties

- Assume that m data points $\mathcal{D} = \{x^1, x^2, \ldots x^m\}$ drawn independently and identically (iid) from some distribution $P^*(x)$

- Want to fit the data with a model $P(x|\theta)$ with parameter θ

$$\theta = argmax_\theta \, \log P(\mathcal{D}|\theta) = argmax_\theta \, \log \prod_{i=1}^{m} P(x^i|\theta)$$

Example problem

- Estimate the probability θ of landing in heads using a biased coin

- Given a sequence of m independently and identically distributed (iid) flips
  - Eg.,$\mathcal{D} = \{x^1, x^2, \dots x^m\} = \{1,0,1, \dots, 0\}, x^i \in \{0,1\}$

- Model: $P(x|\theta) = \theta^x(1-\theta)^{1-x}$
  - $P(x|\theta\,) \;=\; \begin{cases} 1-\theta, for\ x = 0 \\ \theta, \qquad for\ x = 1 \end{cases}$

- Likelihood of a single observation x_i ?
  - $P\left(x^i|\theta\right) = \theta^{x^i}(1-\theta)^{1-x^i}$

MLE for Biased Coin

- Objective function, log likelihood

$$l(\theta;\mathcal{D}) = \log P(\mathcal{D}|\theta) = \log \theta^{n_h}(1-\theta)^{n_t}$$
$$= n_h \log\theta + (m - n_h)\log(1-\theta)$$

n_h: number of heads, n_t: number of tails

- Maximize $l(\theta;\mathcal{D})$ w.r.t. θ

- Take derivatives w.r.t. θ

$$\frac{\partial l}{\partial \theta} = \frac{n_h}{\theta} - \frac{(m-n_h)}{1-\theta} = 0$$

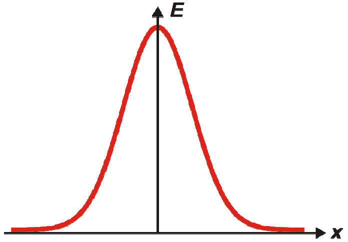
$\Rightarrow\ \hat{\theta}_{MLE} = \frac{n_h}{m}$ or $\hat{\theta}_{MLE} = \frac{1}{m}\sum_i x^i$

Estimating Gaussian distribution

- Gaussian distribution in R

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

- Need to estimate two sets of parameters μ, σ



- Given m iid samples

$$\mathcal{D} = \{x^1, x^2, \dots x^m\}, x^i \in R$$

- Likelihood of one data point:

$$p\left(x^i \middle| \mu,\sigma\right) \propto exp\left(-\frac{1}{2\sigma^2}\left(x^i-\mu\right)^2\right)$$

MLE for Gaussian distribution

- Objective function, log likelihood

$$l(\mu,\sigma;\mathcal{D}) = \log \prod_{i=1}^{m} \frac{1}{(2\pi)^{\frac{1}{2}}\sigma} exp\left(-\frac{1}{2\sigma^2}\left(x^i-\mu\right)^2\right)$$

$$= -\frac{m}{2}\log 2\pi - \frac{m}{2}\log\sigma^2 \ - \sum_{i=1}^{m}\frac{\left(x^i-\mu\right)^2}{2\sigma^2}$$

- Maximize $l(\mu,\sigma;\mathcal{D})$ with respect to μ,σ

- Take derivatives w.r.t. μ, σ^2

$$\frac{\partial l}{\partial \mu} = 0$$

$$\frac{\partial l}{\partial \sigma^2} = 0$$

Gaussian

$$l(\mu, \sigma; \mathcal{D}) = -\frac{m}{2}\log 2\pi - \frac{m}{2}\log \sigma^2 \; - \sum_{i=1}^{m} \frac{\left(x^i - \mu\right)^2}{2\sigma^2}$$

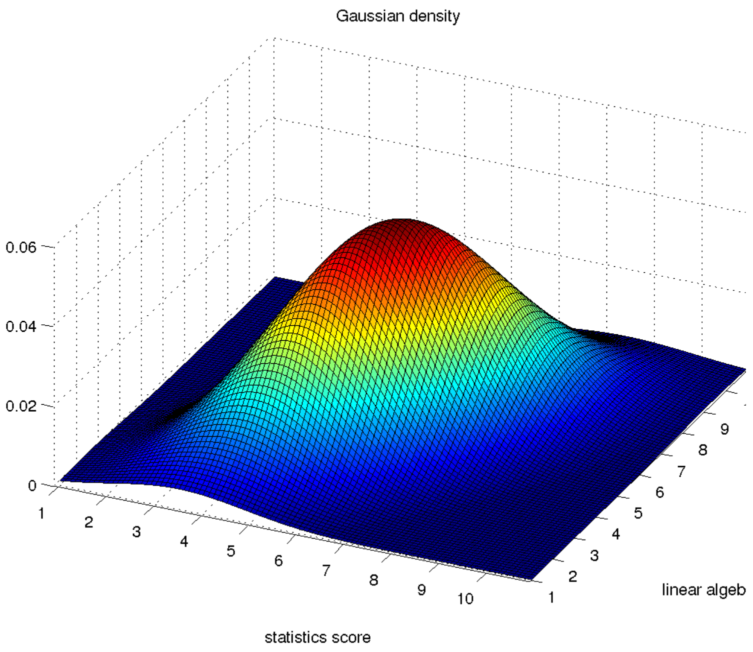
$$\frac{\partial l}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{m}(x^i - \mu) = 0$$

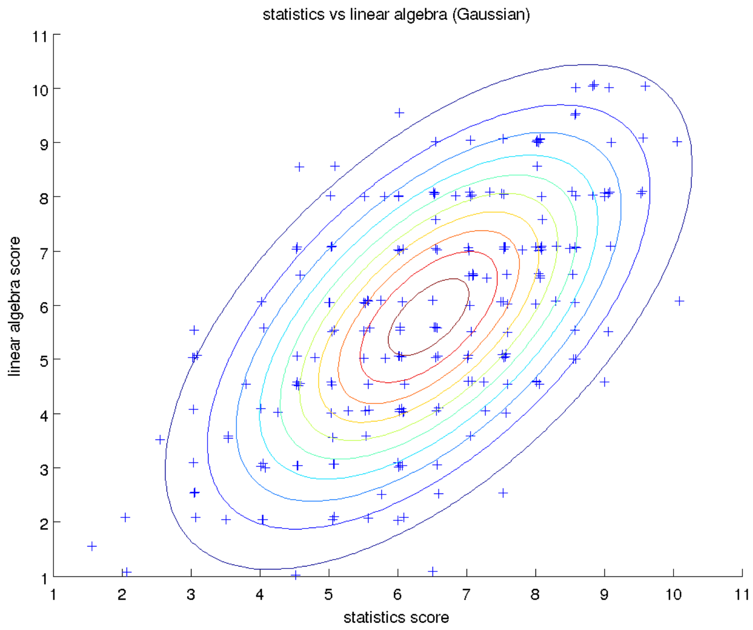
$$\Rightarrow \sum_{i}^{m} x^i = m\mu \Rightarrow \mu = \frac{1}{m}\sum_{i=1}^{m} x^i$$

$$\frac{\partial l}{\partial \sigma^2} = -\frac{m}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i}^{m}\left(x^i - \mu\right)^2 = 0$$

$$\Rightarrow \sigma^2 \Rightarrow \frac{1}{m}\sum_{i=1}^{m}\left(x^i - \mu\right)^2$$

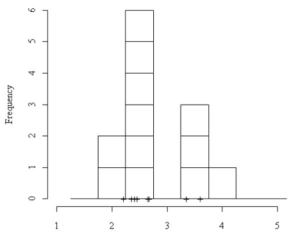
Density example





1-D Histogram

- One the simplest nonparametric density estimator



- Given m iid samples $\mathcal{D} = \{x^1, x^2, \dots x^m\}, x^i \in [0,1)$

- Split $[0,1)$ into n bins

$$B_1 = \left[0, \frac{1}{n}\right), B_2 = \left[\frac{1}{n}, \frac{2}{n}\right), \dots B_n = \left[\frac{n-1}{n}, 1\right)$$

- Count the number of points, c_1 within B_1, c_2 within B_2…

- For a new test point x

$$p(x) = \sum_{j=1}^{n} \frac{nc_j}{m} I(x \in B_j)$$

Why is histogram valid?

Requirement for density $p(x)$

$$p(x) \geq 0, \int_{\Omega} p(x)dx = 1$$

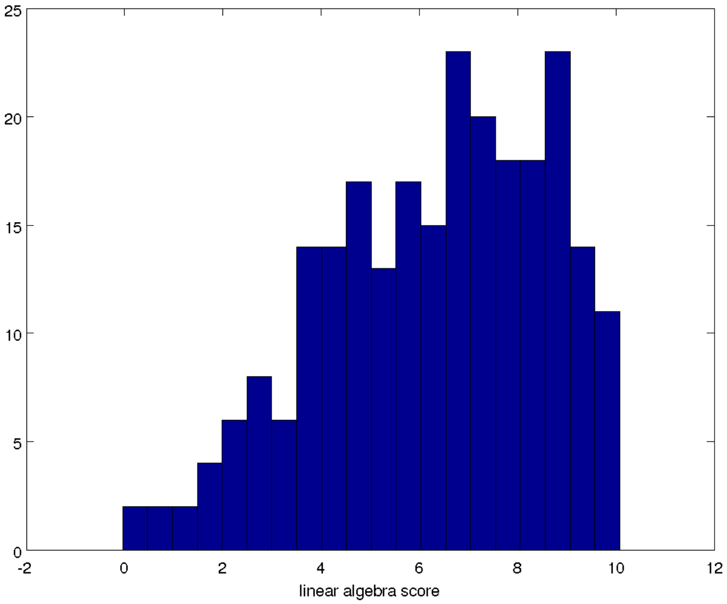
Verify this is satisfied for our histogram,

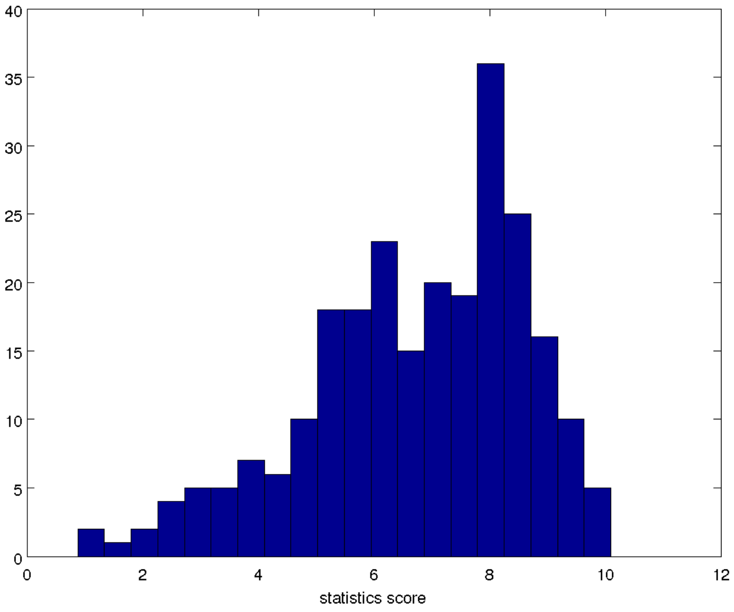
$$\int_{[0,1)} p(x)dx = \int_{[0,1)} \sum_{j=1}^{n} \frac{nc_j}{m} I\left(x \in B_j\right)dx$$

$$= \sum_{j=1}^{n} \int_{\left[\frac{j-1}{n},\frac{j}{n}\right)} \frac{nc_j}{m} dx$$

$$= \sum_{j=1}^{n} \frac{c_j}{m} = 1$$

Example: test scores





Higher dimensional histogram

- Given m iid samples $\mathcal{D} = \{x^1, x^2, \dots x^m\}, x^i \in [0,1)^d$

- Split $[0,1)^d$ evenly into n^d bins

$$B_1 = \left[0, \frac{1}{n}\right) \times \left[0, \frac{1}{n}\right) \dots \times \left[0, \frac{1}{n}\right),$$

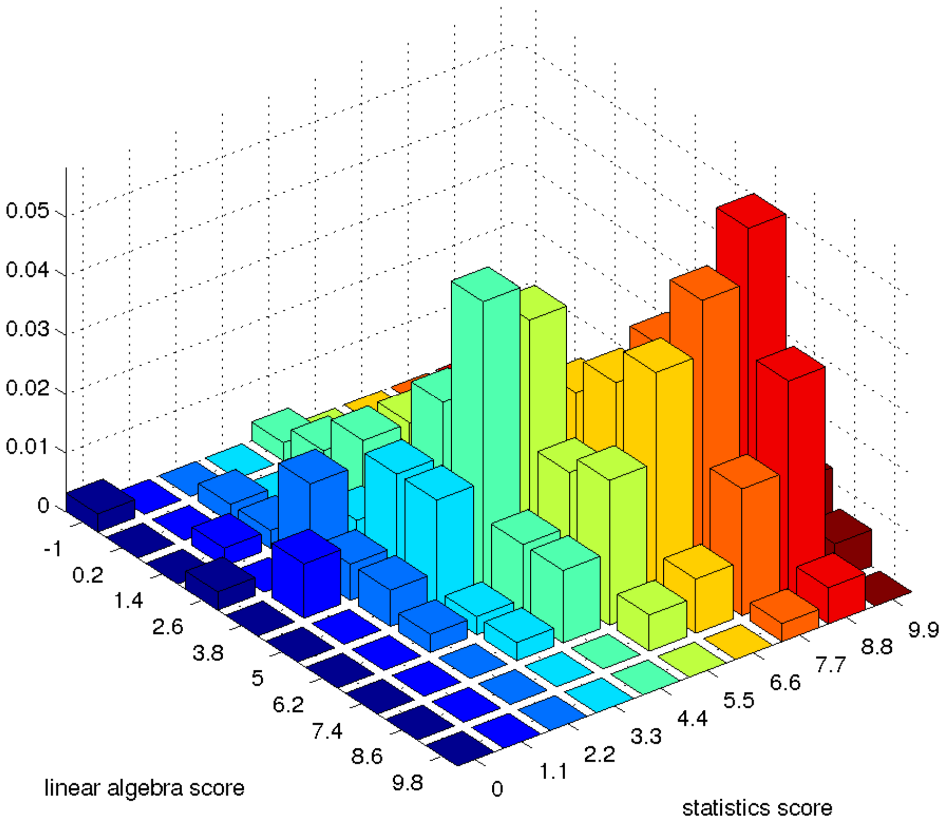
$$B_2 = \left[\frac{1}{n}, \frac{2}{n}\right) \times \left[0, \frac{1}{n}\right) \dots \times \left[0, \frac{1}{n}\right),$$

$$\dots$$

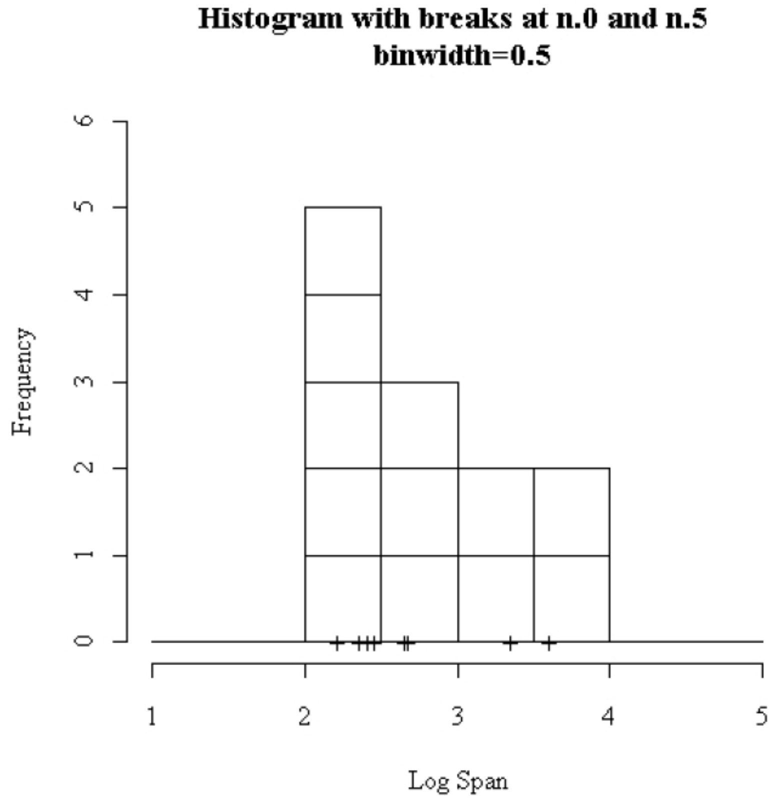
$$B_{n^d} = \left[\frac{n-1}{n}, 1\right) \times \left[\frac{n-1}{n}, 1\right) \dots \times \left[\frac{n-1}{n}, 1\right)$$

- Bin size is $h = \frac{1}{n}$

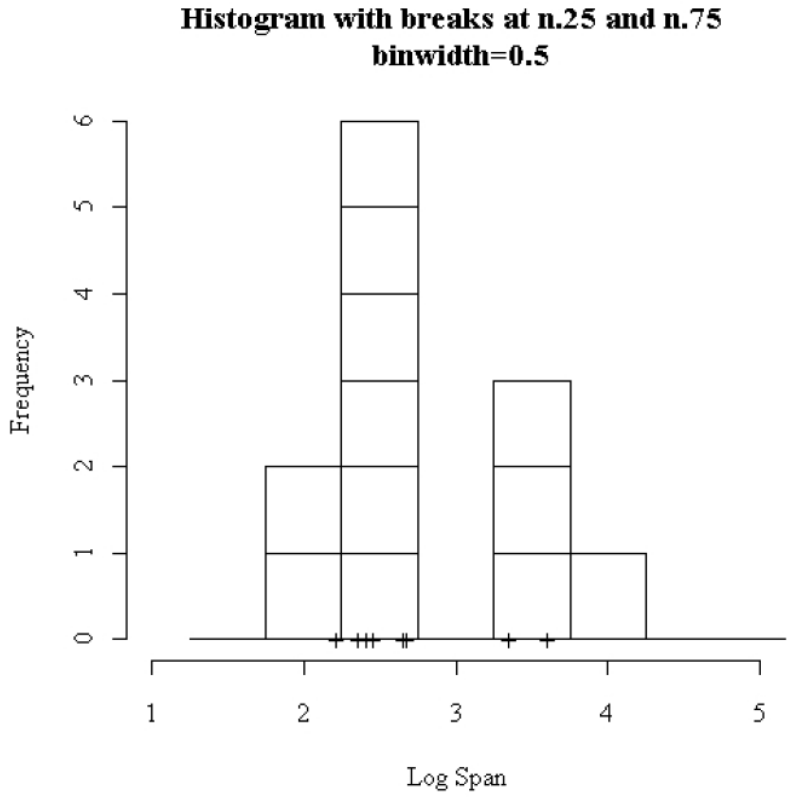
Class scores



Output depends on where you put the bins

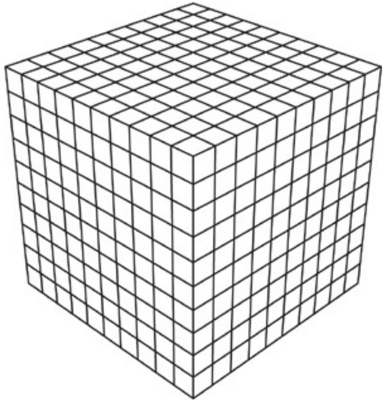


Output depends on where you put the bins



Computation and statistical considerations

- Problem I: too many bin! Not good for high dimensional data
  - If n^d is larger than m, most bins are empty
  - Eg. $n = 10, d = 6$, need ~1 million data points



- Problem II: statistically histogram is not the best
  – Integrated risk:
  $$r(\hat{p}, p) := \int_R \mathbb{E}_X\left[\left(\hat{p}(x) - p(x)\right)^2\right] dx$$
  – Histogram (with bin size $h \sim m^{-1/3}$)
  $$r(\hat{p}, p) \sim \frac{C}{m^{2/3}}$$
  – Kernel density estimator (with bandwidth $h \sim m^{-1/5}$)
  $$r(\hat{p}, p) \sim \frac{C}{m^{4/5}}$$
  – Difference even big for high dimensional data

Kernel density estimation

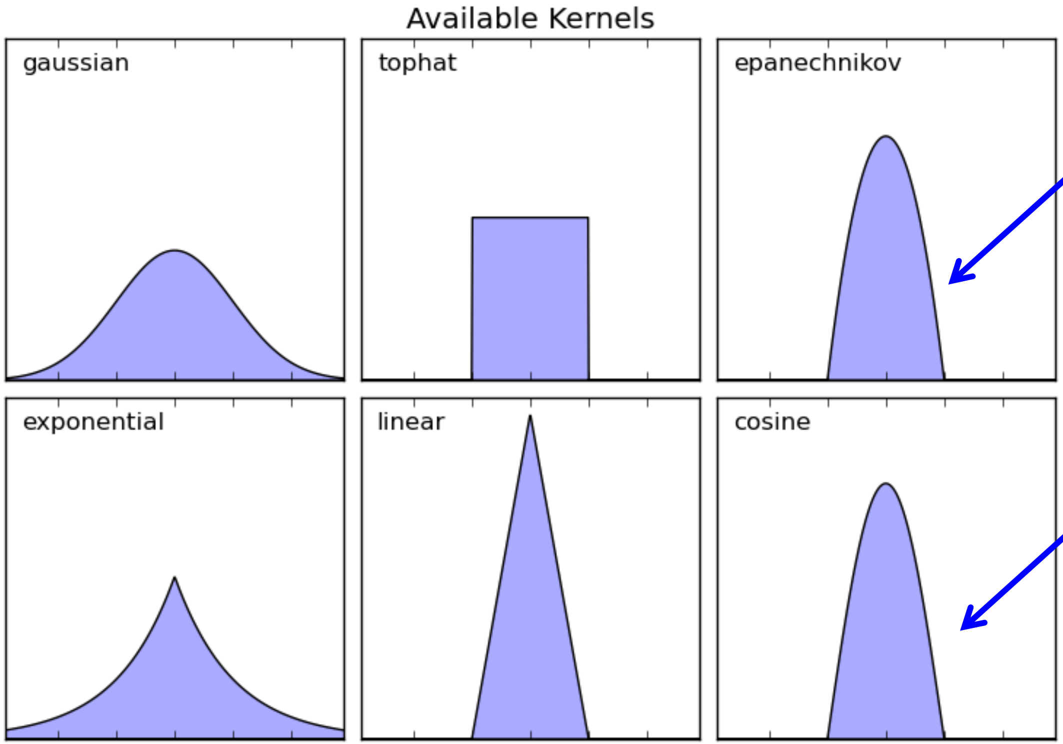
- Kernel density estimator
- $p(x) = \frac{1}{m} \sum_i^m \frac{1}{h} K\left(\frac{x^i - x}{h}\right)$

- Smoothing kernel function
  - $K(u) \geq 0,$
  - $\int K(u) du = 1,$
  - $\int uK(u) = 0,$
  - $\int u^2 K(u) du \leq \infty$

- An example: Gaussian kernel $K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$

Smoothing kernel functions

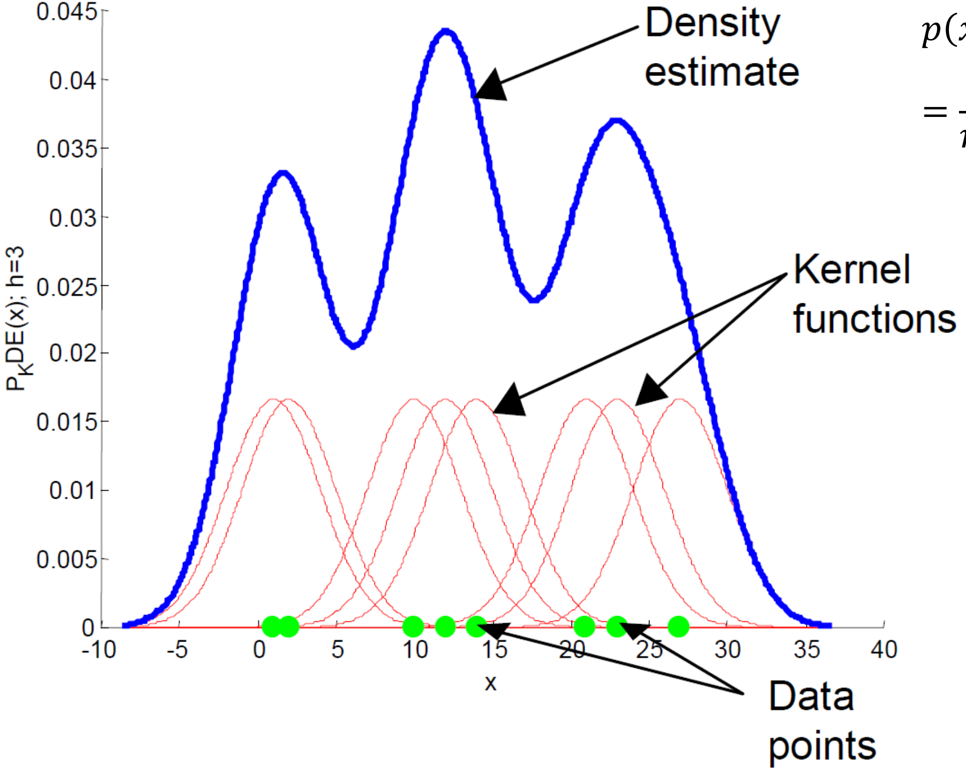
- An example: Gaussian kernel $K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$



$K(u) = \frac{3}{4}(1 - u^2) I(|u| \leq 1)$

$K(u) = \frac{\pi}{4} \cos\left(\frac{\pi}{2} u\right) I(|u| \leq 1)$

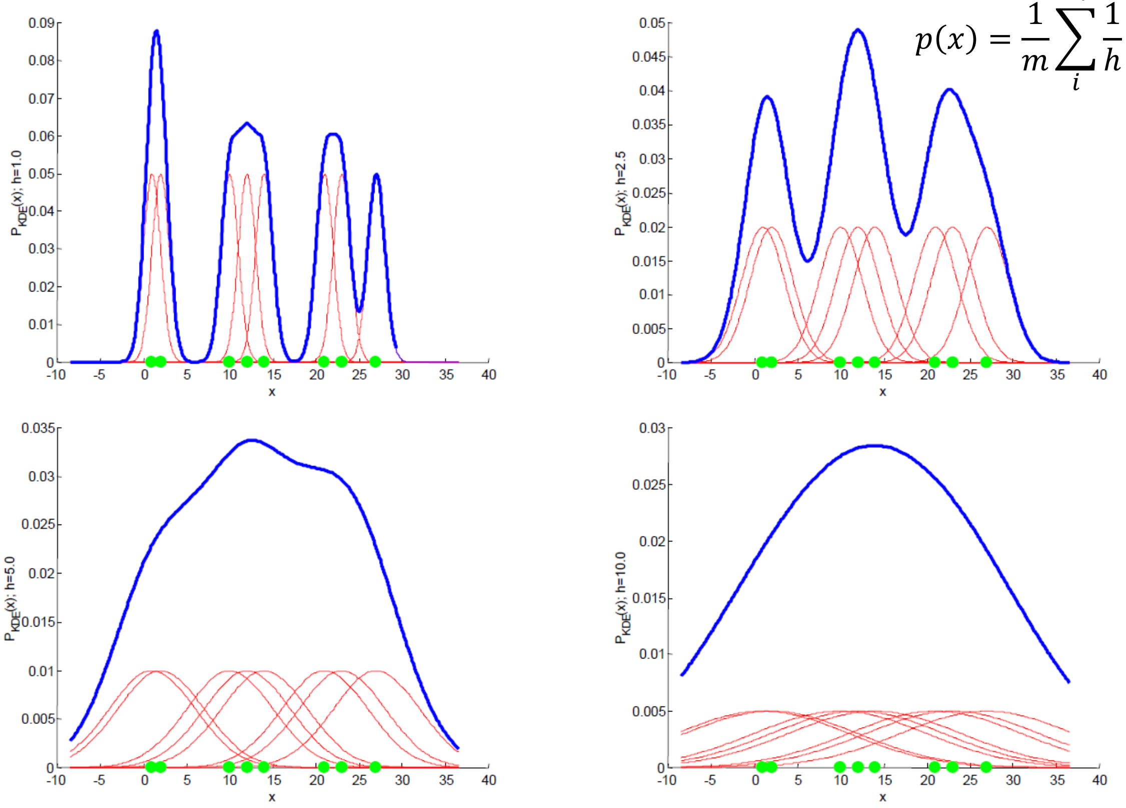
Example



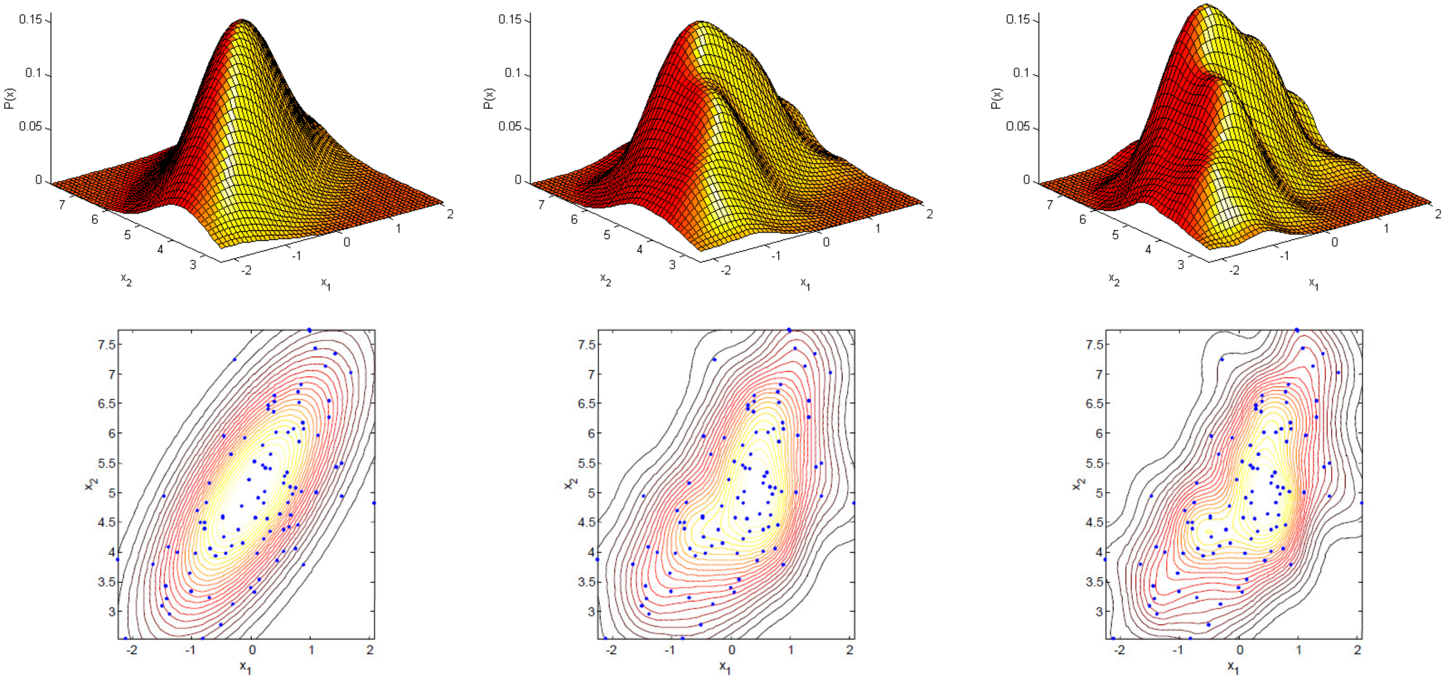
$$p(x) = \frac{1}{m}\sum_{i}^{m}\frac{1}{h}K\left(\frac{x^i - x}{h}\right)$$

Effect of the kernel bandwidth h

$$p(x) = \frac{1}{m}\sum_{i}^{m}\frac{1}{h}K\left(\frac{x^i - x}{h}\right)$$



Two-dimensional example



Wine data example

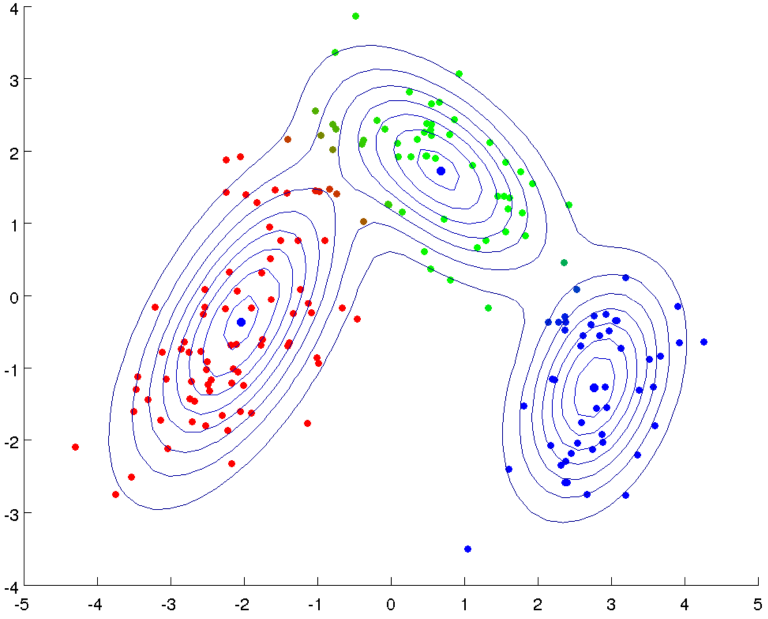
- The wine data set was introduced by Forina et al. (1986).
- It originally included the results of 27 chemical measurements on 178 wines made in the same region in Italy but derived from three different cultivars: Barolo, Grignolino and Barbera.
- We extract the first two principle components of the data, and aim to fit a density distribution

Wine data set (http://archive.ics.uci.edu/ml/datasets/Wine+Quality)

- These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. Feature include
  - 1) Alcohol
  - 2) Malic acid
  - 3) Ash
  - 4) Alcalinity of ash
  - 5) Magnesium
  - 6) Total phenols
  - 7) Flavanoids
  - 8) Nonflavanoid phenols
  - 9) Proanthocyanins
  - 10)Color intensity
  - 11)Hue
  - 12)OD280/OD315 of diluted wines
  - 13)Proline

Demo: test_wine.m

- Chemical analysis of wines grown in three different places
- Clear cluster structure, can we fit 3 Gaussians?



What is the best kernel bandwidth?

Silverman’s rule of thumb: If using the Gaussian kernel, a good choice for is

$$h \approx 1.06\,\hat{\sigma}\,m^{-1/5}$$

where is $\hat{\sigma}$ the standard deviation of the samples

A better but more computationally intensive approach:

- Randomly split the data into two sets
- Obtain a kernel density estimate for the first
- Measure the likelihood of the second set
- Repeat over many random splits and average

Parametric vs. nonparametric

- Data $x \in R^d$ with fixed dimension d
- Given m training data points $\{x^1, x^2 \dots, x^m\}$
- Partition n bin in each dimension

Aspects	Gaussian	Histogram	KDE
Flexible	Not	Yes	Yes
Assumption	Strong	Not	Not
Parameter number	Fixed	Increase with n	Increase with m
Memory requirement	$d + d^2$	n^d	md
Training computation	Closed form	Binning and Counting	nothing
Test computation	Plug in formula	Find the bin	Evaluate m functions
Statistical guarantee	only Gaussian case	Arbitrary (worse)	Arbitrary (better)

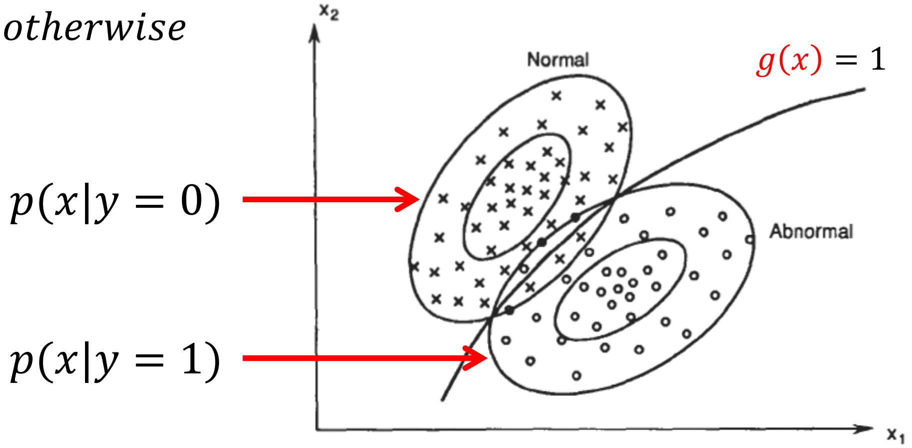
Classification using density estimation

- Simple binary classifier for input $x^i \in R^d$ and label $y^i \in \{0,1\}$
  - Step I: use label to estimate $p(y=0)$ and $p(y=1)$
  - Step II: divide your data according to the value of y, and estimate

$$p(x|y=0) \text{ and } p(x|y=1)$$

  - Step III: Classify a new test point x as

$$\begin{cases} 1, & g(x) \coloneqq \dfrac{p(x|y=1)p(y=1)}{p(x|y=0)p(y=0)} > 1 \\ 0, & otherwise \end{cases}$$




SUMMARY