# Lecture 13
# Midterm Review

Mahdi Roozbahani
Georgia Tech

These slides are adopted based on slides from Le Song.


me brushing over
a boring topic
sky atlantic
me in the exam
first question of
the exam

# Linear Algebra Basics

- Norms
  - Vector nom, matrix norm
- Multiplications
  - Vector dot product, matrix-vector multiplication, matrix-matrix multiplication
- Matrix Inversion
  - Linear dependence, rank, matrix inverse, invertibility
- Trace and Determinant
- Eigen Values and Eigen Vectors
- Singular Value Decomposition
- Matrix Calculus

$$\begin{bmatrix} 1 & 0 & 2 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix}$$

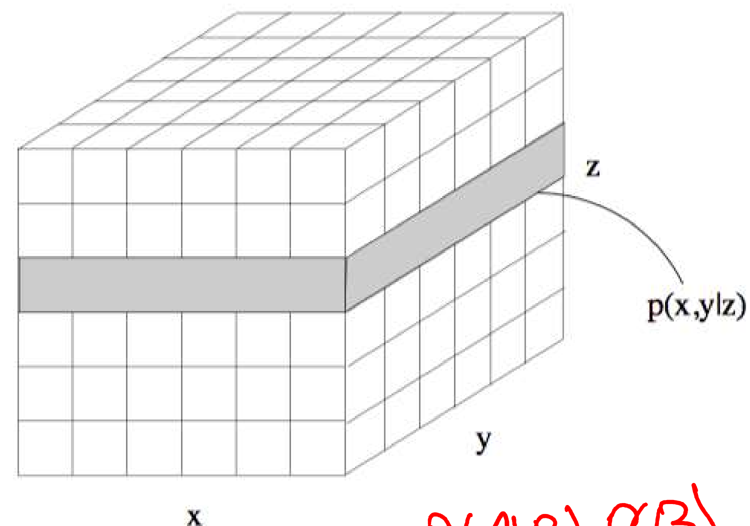
# Basic Probability Theory and Statistics

$$P(x) = \sum_y \underbrace{P(x,y)} = \sum_y P(x|y)\,P(y)$$

- Probability Distributions
  - Random variable, sample space, probability density, discrete vs continuous

$$P(A|B) = \frac{P(A,B)}{\cancel{P(B)}} = \frac{P(B|A)\,P(A)}{P(A|B)\,\cancel{P(B)}}$$

- Joint and Conditional Probability Distributions
  - Joint dist., marginal dist., conditional dist., i




- Bayes' Rule

$$\mu = \frac{\sum x_i}{n}$$

- Mean and Variance
- Properties of Gaussian Distribution
- Maximum Likelihood Estimation
  - Inferring parameters with MLE, optimization

$$P(B|A) = \frac{P(A|B)\,P(B)}{P(A)}$$

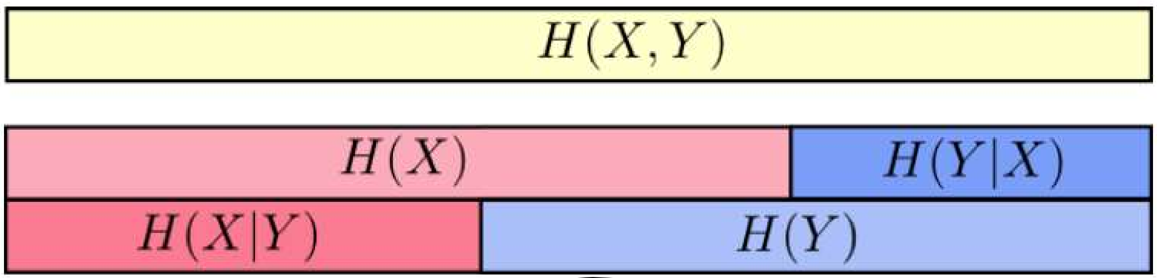
# Basic Information Theory

Expected [illegible] of information $H(X) = \sum p(x)\log_2 \frac{1}{p(x)}$

$I(x) = \log_2 \frac{1}{p(x)}$

- Entropy
- Conditional Entropy and Mutual Information
- Cross-Entropy and KL-Divergence

$I(x) = \log_2 \frac{1}{p(x)}$




$E(x) = \sum p(x) f(x)$

$f(x)$

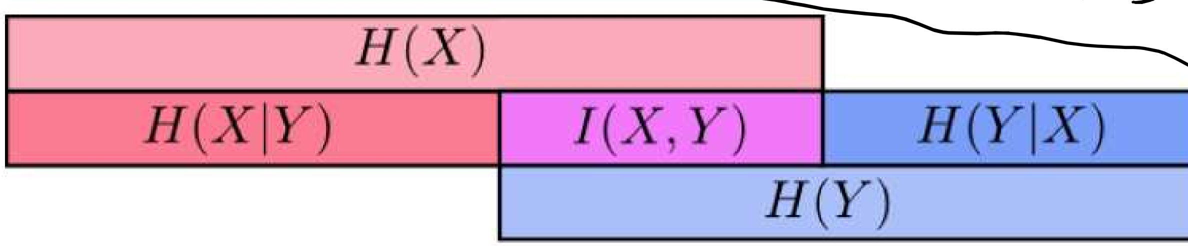



Image credit: Chris Olah

# Clustering Analysis

- You pick your similarity/dissimilarity function
- The algorithm figures out the grouping of objects based on the chosen similarity/dissimilarity function
  - Points within a cluster is similar
  - Points across clusters are not so similar
- Issues for clustering
  - How to represent objects? (Vector space? Normalization?)
  - What is a similarity/dissimilarity function for your data?
  - What are the algorithm steps?

# K-Means Algorithm

- Initialize $k$ cluster centers, $\{c^1, c^2, \ldots, c^k\}$, randomly

- Do

  - Decide the cluster memberships of each data point, $x^i$, by assigning it to the nearest cluster center (cluster assignment)

$$\pi(i) = argmin_{j=1,\ldots,k} \left\|x^i - c^j\right\|^2$$

  - Adjust the cluster centers (center adjustment)

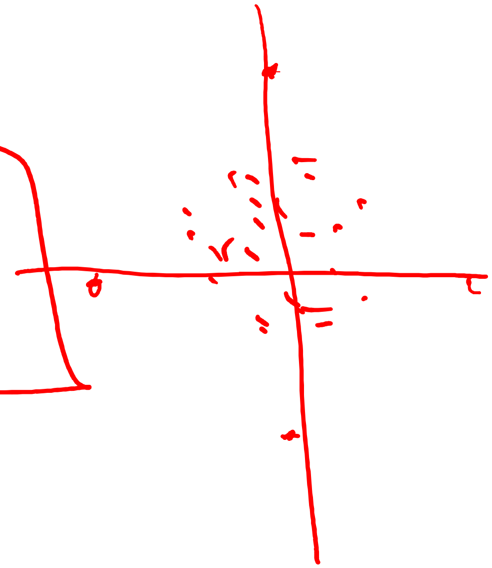
$$c^j = \frac{1}{|\{i: \pi(i) = j\}|} \sum_{i:\pi(i)=j} x^i$$

- While any cluster center has been changed

# Convergence of K-Means

- Will kmeans objective oscillate?

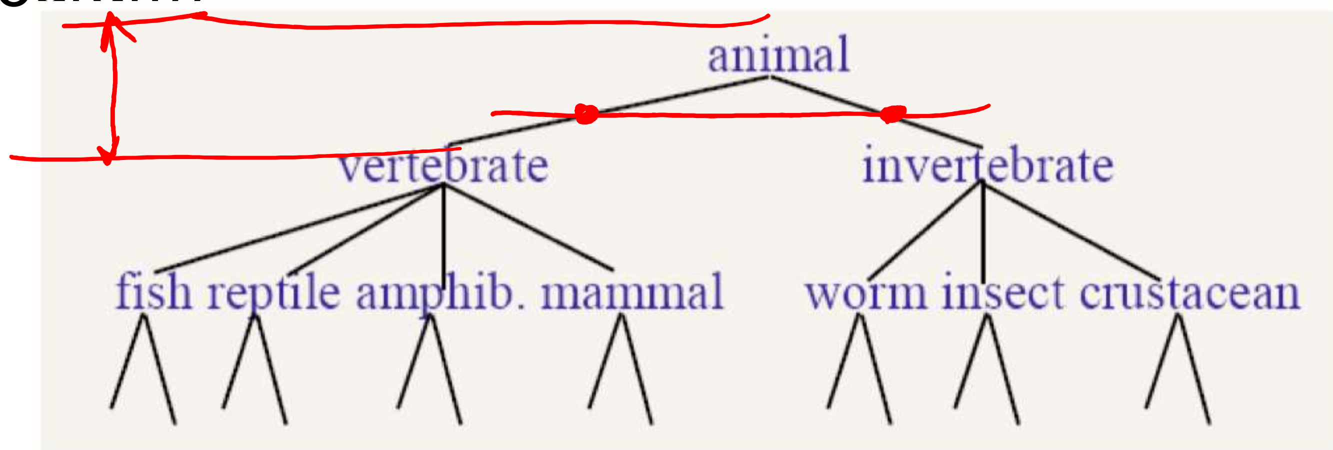
$$\frac{1}{m}\sum_{i=1}^{m}\left\|x^i - c^{\pi(i)}\right\|^2$$



- The minimum value of the objective is finite
- Each iteration of kmeans algorithm decrease the objective
  - Cluster assignment step decreases objective
    - $\pi(i) = argmin_{j=1,\dots,k} \left\|x^i - c^j\right\|^2$ for each data point $i$
  - Center adjustment step decreases objective
    - $c^j = \frac{1}{|\{i:\pi(i)=j\}|}\sum_{i:\pi(i)=j} x^i = argmin_c \sum_{i:\pi(i)=j}\left\|x^i - c\right\|^2$

# Hierarchical Clustering

- Organize objects into a tree-based hierarchical taxonomy (dendrogram)




- Many applications in the real world
  - Web pages
  - News articles
  - Scientific papers

# Two Paradigms for Hierarchical Clustering

- Bottom-up Agglomerative Clustering
  - Start by considering each object as a separate cluster
  - Repeatedly join the closest pair of clusters
  - Stop when there is only one cluster left

- Top-Down Divisive Clustering
  - Start by considering all objects as one large cluster
  - Recursively divide each cluster into two sub-clusters
  - Stop when each cluster contains only one object

# Distance Between Two Clusters

## Single-Link

- Nearest Neighbor: their closest members.

## Complete-Link

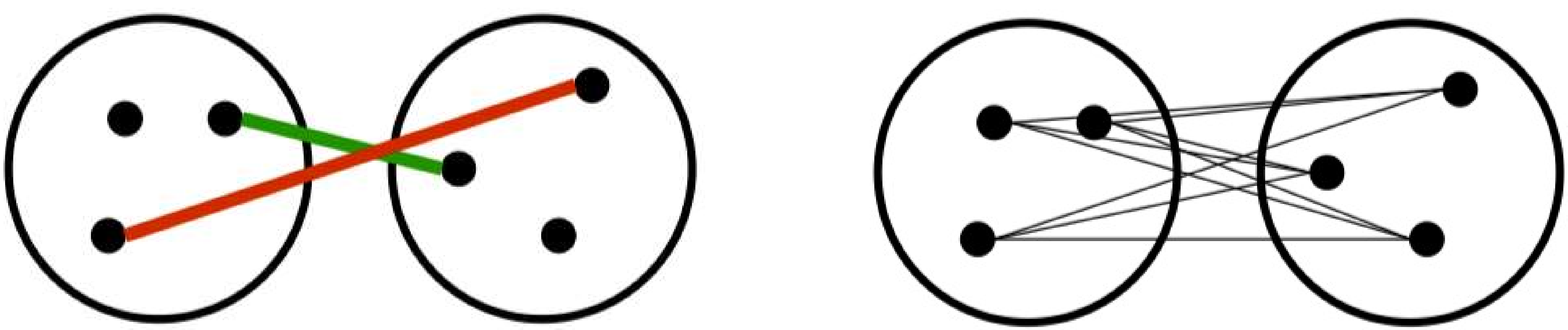
- Furthest Neighbor: their furthest members.

## Centroid:

- Clusters whose centroids (centers of gravity) are the most cosine-similar

## Average:

- average of all cross-cluster pairs.

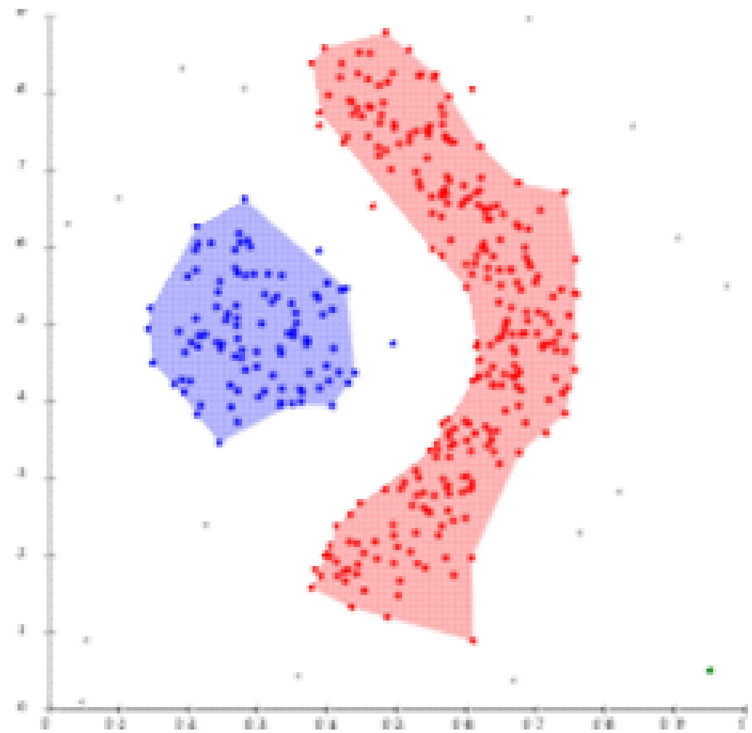


# Density-Based Clustering

MinPts

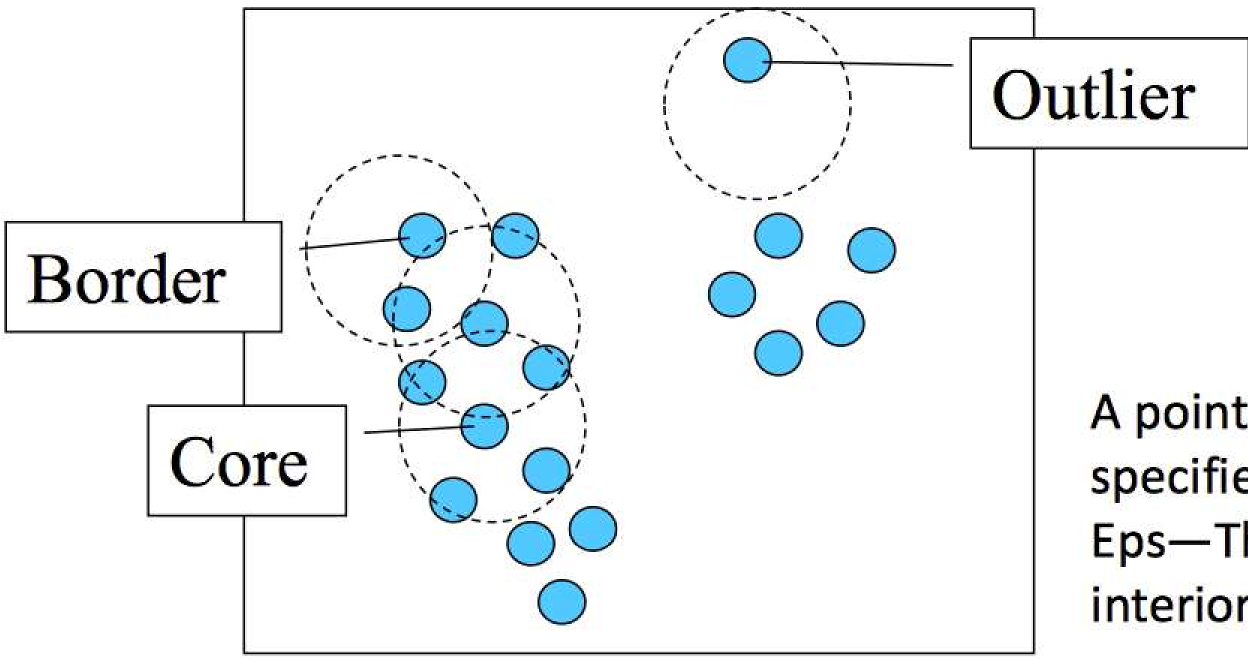
$\varepsilon$

- Basic Idea
  - Clusters are dense regions in the data space, separated by regions of lower density
  - A cluster is defined as a maximal set of density-connected points
  - Detect arbitrarily shaped clusters



- Method
  - DBSCAN (Density-Based Spatial Clustering of Applications with Noise )

# Core Points, Border Points, and Outliers




$\varepsilon$ = 1unit, MinPts = 5

Given $\varepsilon$ and *MinPts*, categorize the objects into three exclusive groups.

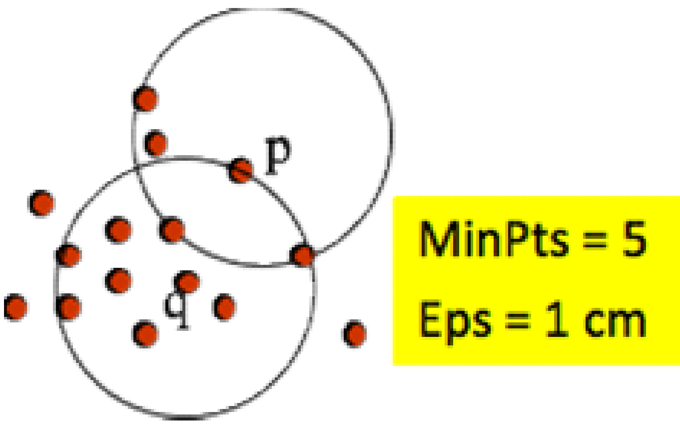
A point is a core point if it has more than a specified number of points (MinPts) within Eps—These are points that are at the interior of a cluster.

A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point.

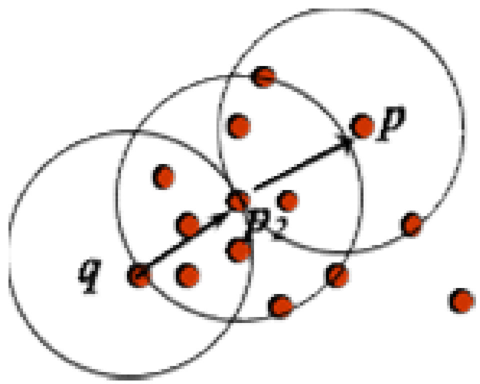
A noise point is any point that is not a core point nor a border point.

# Density Reachability

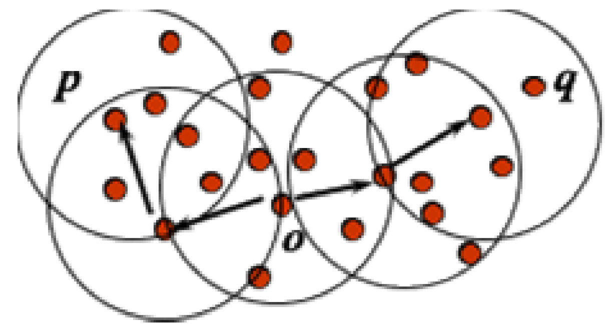
- Density reachability:
  - A point ***p*** is density-reachable from a point ***q*** if there is a chain of points $\boldsymbol{p_1, \ldots, p_n}$, $\boldsymbol{p_1 = q}$, $\boldsymbol{p_n = p}$ such that $\boldsymbol{p_{i+1}}$ is directly density-reachable from $\boldsymbol{p_i}$
  - $\boldsymbol{p_1 = q \rightarrow p_2 \rightarrow \ldots \rightarrow p_n = q}$




Directly Density-Reachable




Density-Reachable




Density-Connected

# Gaussian Mixture Model for Soft Clustering

$P(y|x) = \frac{P(y,x)}{P(x)}$

- **K-means**
  - **hard assignment:** each object belongs to only one cluster

$P(y|x) = f(\theta_i)$

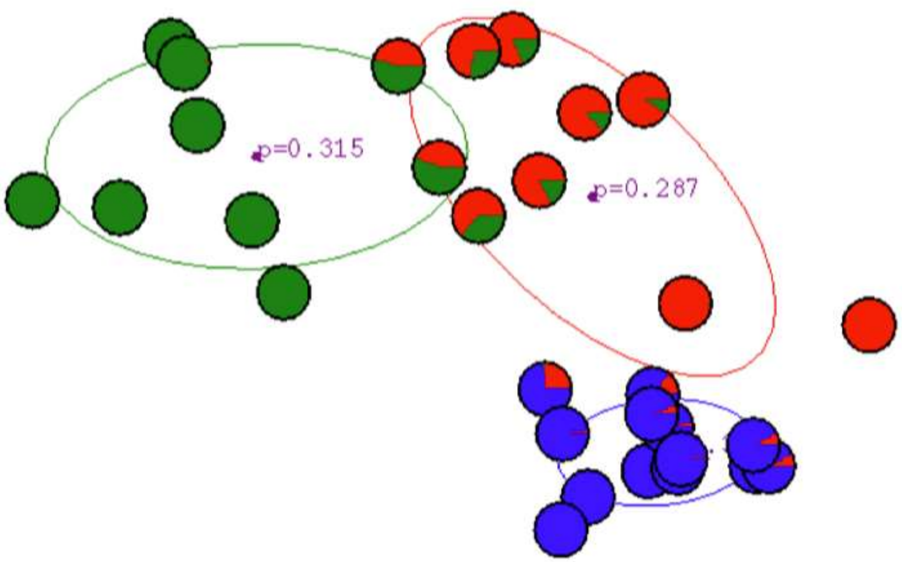
$$\theta_i \in \{\theta_1, \ldots, \theta_K\}$$

- **Mixture modeling**
  - **soft assignment:** probability that an object belongs to a cluster

$$(\pi_1, \ldots, \pi_K),\ \pi_i \geq 0,\ \sum_{i=1}^{K} \pi_i = 1$$

$k=3 \quad N=100$

$P(x) = \sum_z P(x,z)$



# Mixture Models

- Formally a Mixture Model is the weighted sum of a number of pdfs where the weights are determined by a distribution, $\pi$

$$\int p(x)\,d(x) = \int \pi_0 f_0(x) + \pi_1 f_1(x) + \pi_2 f_2(x) + \ldots + \pi_k f_k(x)\,dx$$

where $\sum_{i=0}^{k} \pi_i = 1$

$$\int p(x)\,dx = 1$$

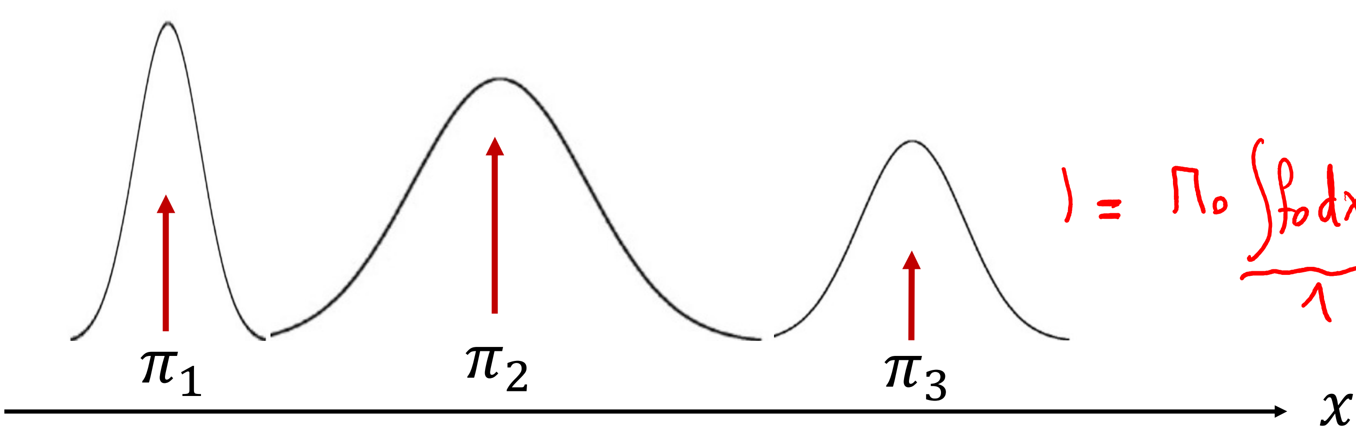
$$\int f_0(x)\,dx = 1$$

$$\boxed{p(x) = \sum_{i=0}^{k} \pi_i f_i(x)}$$

What is **f** in GMM?

$$1 = \pi_0 \underbrace{\int f_0\,dx}_{1} + \pi_1 \underbrace{\int f_1\,dx}_{1} + \cdots$$

$$1 = \sum_k \pi_k$$



Start with parameters describing each cluster:

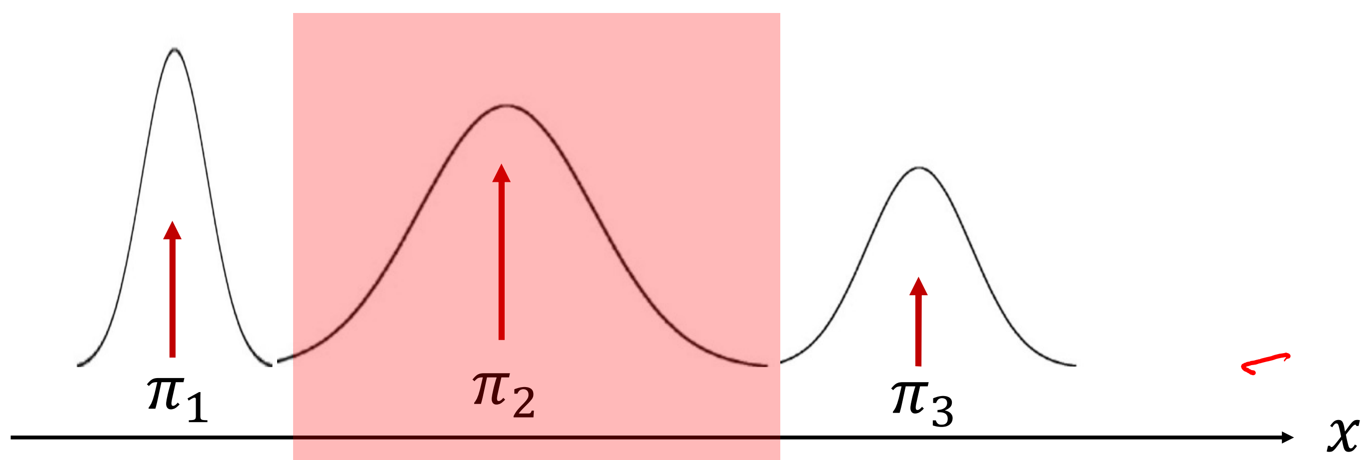
Mean $\mu_k$ Variance $\sigma_k$ Size $\pi_k$

Marginal probability distribution

$$\mathrm{p}(\mathrm{x}|\theta) = \sum_k p(x, z_{nk}|\theta) = \sum_k \underbrace{p(x|z_{nk}, \theta)}_{f_k(x)} \underbrace{p(z_{nk}|\theta\ )}_{\pi_k} = \sum_k N(x|\mu_k, \sigma_k)\pi_k$$

$p(z_{nk}|\theta) = \pi_k$ Select a mixture component with probability $\pi$

$p(x|z_{nk}, \theta) = N(x|\mu_k, \sigma_k)$ Sample from that component's Gaussian



# Inferring Cluster Membership

- We have representations of the joint $p(x, z_{nk}|\theta)$ and the marginal, $p(x|\theta)$
- The conditional of $p(z_{nk}|x, \theta)$ can be derived using Bayes rule.
  - The **responsibility** that **a** mixture component takes for explaining an observation x.

$\boxed{p(x) = f_1 \pi_1 + f_2 \pi_2}$

$$\frac{f_1 \pi_1}{p(x) = f_1 \pi_1 + f_2 \pi_2} =$$

$$\tau(z_{nk}) = p(z_{nk}|x) = \frac{p(z_{nk})p(x|z_{nk}, \theta)}{\sum_{j=1}^{K} p(z_{nj})p(x|z_{nj})}$$

$$= \frac{\pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x|\mu_j, \Sigma_j)}$$

$P(z_{nk}|x)$

$$= \frac{N(x|\mu_1, \sigma_1)\, \pi_1}{p(x) = \sum_k N(x|\mu_k, \sigma_k)\, \pi_k} = \frac{P(x|z_{n1})\, P(z_{n1})}{\sum P(x|z_{nk})\, P(z_{nk})} =$$

# Well, we don't know $\pi_k, \mu_k, \Sigma_\mathrm{k}$
# What should we do?

We use a method called "Maximum Likelihood Estimation" (MLE) to solve the problem.

$$\mathrm{p}(\mathrm{x}|\theta) = \sum_k p(x, z_{nk}|\theta) = \sum_k p(z_{nk}|\theta)p(x|z_{nk}, \theta) = \sum_{k=0}^{K} \pi_k N(x|\mu_k, \Sigma_k)$$

Let's identify a likelihood function :

$$P(x) = P(x_1, \ldots, x_n) = \prod_{i=1}^{N} \underline{P(x_i)}$$

Now, let's find the missing parameters that maximizes the likelihood:

$$\arg\max p(x|\theta) = p(x|\pi, \mu, \Sigma) = \prod_{n=1}^{N} p(x_n|\theta) = \prod_{n=1}^{N} \sum_{k=0}^{K} \pi_k N(x_n|\mu_k, \Sigma_k)$$

$$\arg\max p(x) = p(x|\pi,\mu,\Sigma) = \prod_{n=1}^{N} p(x_n|\theta) = \prod_{n=1}^{N}\sum_{k=0}^{K} \pi_k N(x_n|\mu_k,\Sigma_k)$$

$$\ln[p(x)] = \ln[p(x|\pi,\mu,\Sigma)]$$

- As usual: Identify a likelihood function

$$\ln p(x|\pi,\mu,\Sigma) = \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{K} \pi_k N(x_n|\mu_k,\Sigma_k)\right\}$$

- And set partials to zero…

# MLE of a GMM

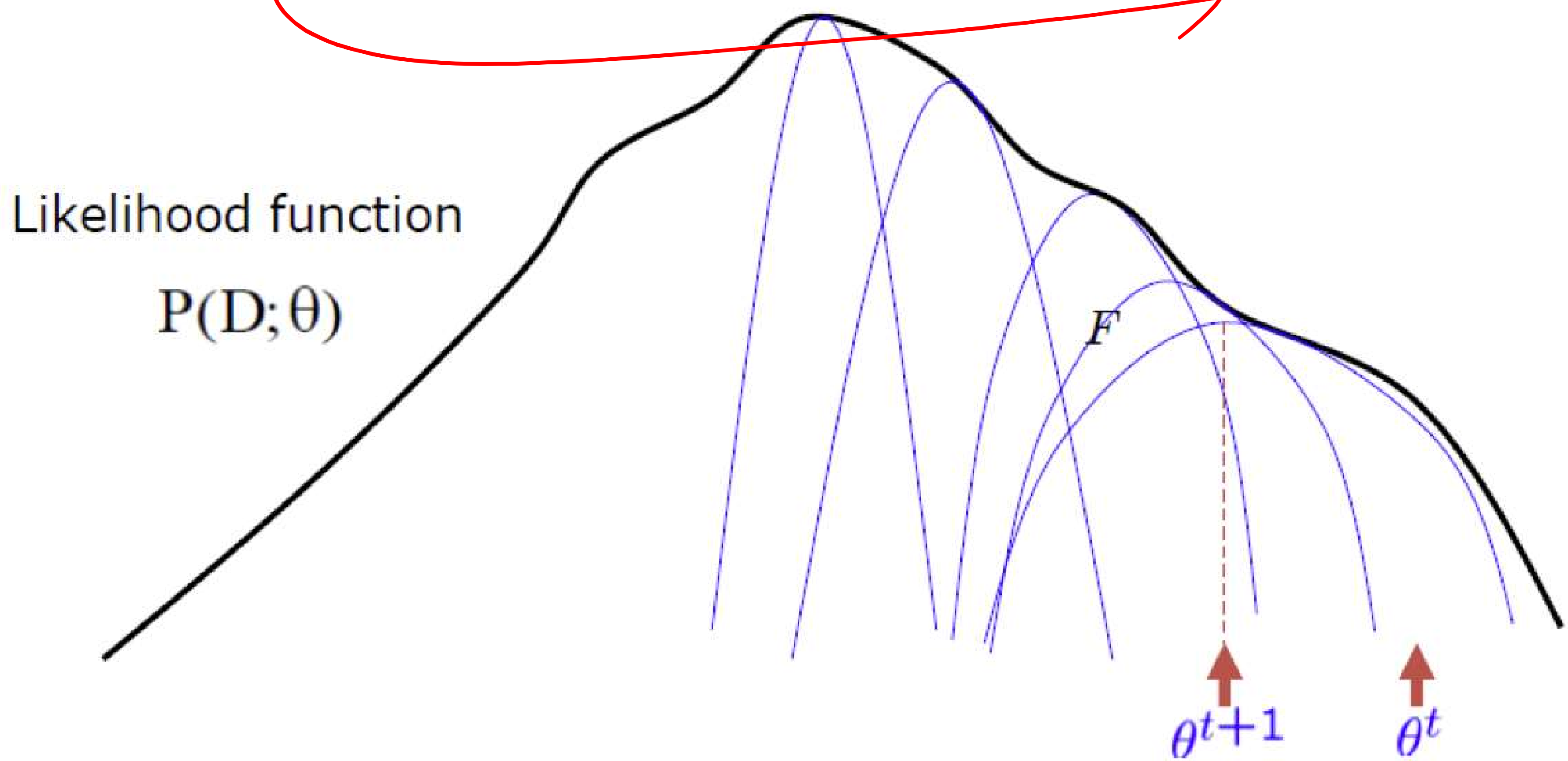
$$\boxed{\mu_k = \frac{\sum_{n=1}^{N} \tau(z_{nk}) x_n}{N_k}}$$

$$\boxed{\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \tau(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}$$

$$\boxed{\pi_k = \frac{N_k}{N}}$$

$$N_k = \sum_{n=1}^{N} \tau(z_{nk})$$

# EM Always Converges

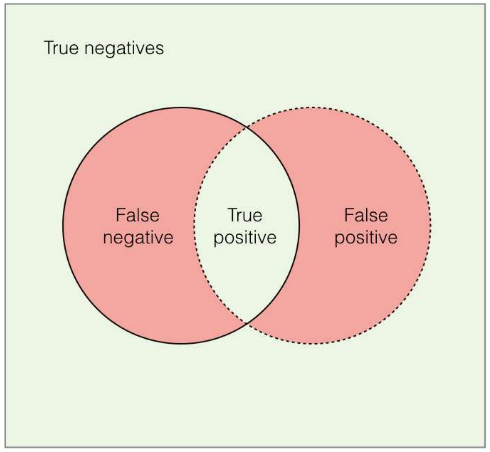



Sequence of EM lower bound F-functions

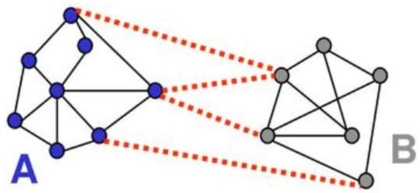
**EM monotonically converges to a local maximum of likelihood**

# Clustering Evaluation

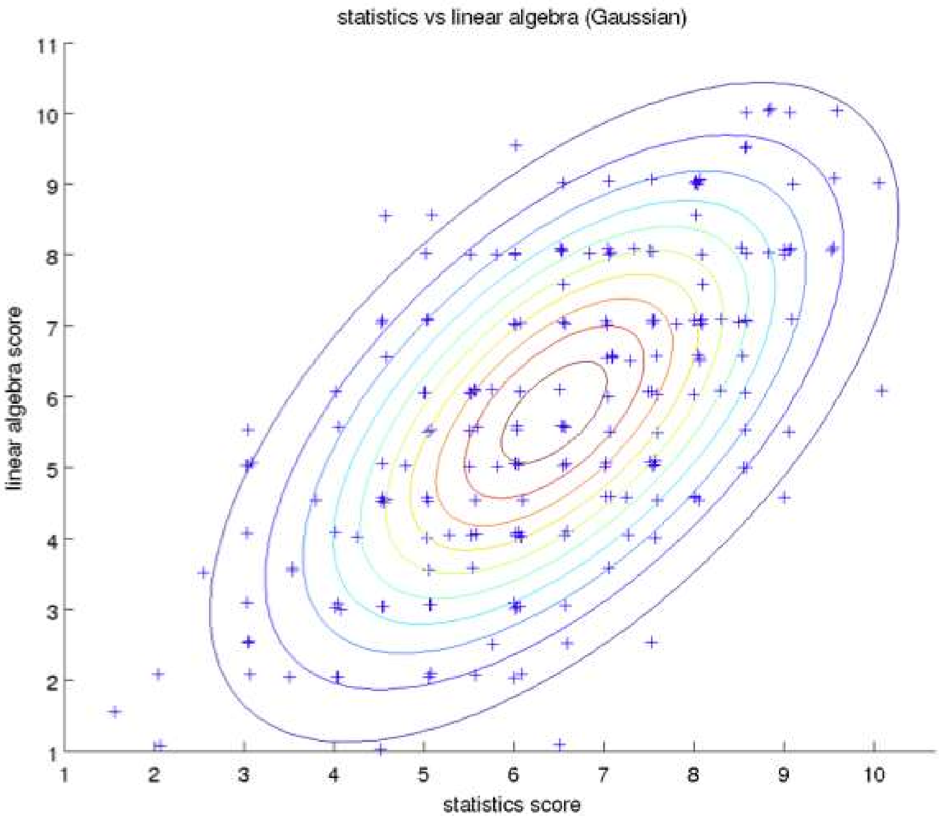
- External measures for clustering evaluation
  - Matching-based measures: Purity, Max Matching, Precision, Recall, F-1
  - Entropy-based measures: Conditional Entropy, Mutual Information
  - Pairwise measures: TP, TN, FP, FN, Jaccard
- Internal measures for clustering evaluation
  - Graph-based measures: Beta-CV, normalized cut
  - Davies-Bouldin Index
  - Silhouette Coefficient

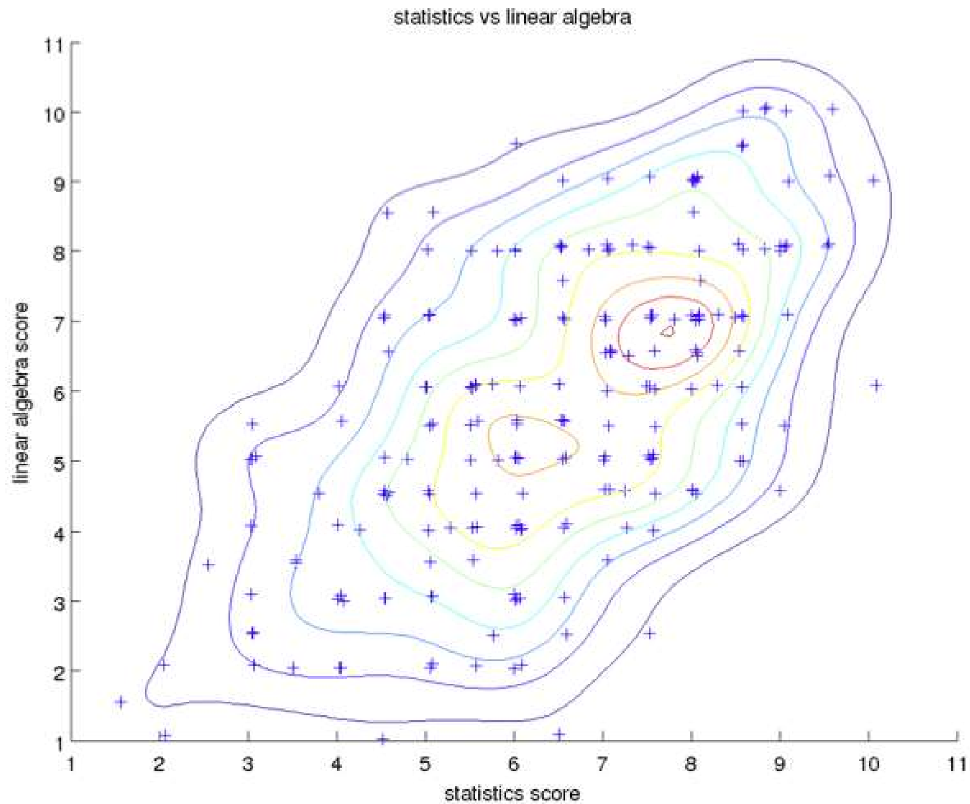



| C\T | $T_1$ | $T_2$ | $T_3$ | Sum |
|---|---|---|---|---|
| $C_1$ | 0 | 30 | 20 | 50 |
| $C_2$ | 0 | 20 | 5 | 25 |
| $C_3$ | 25 | 0 | 0 | 25 |



# Parametric v.s. Nonparametric Density Estimation



# Parametric Density Estimation

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

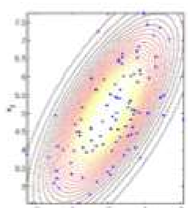
- Discrete case: eg. Bernoulli distribution

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

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 $R^n$

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



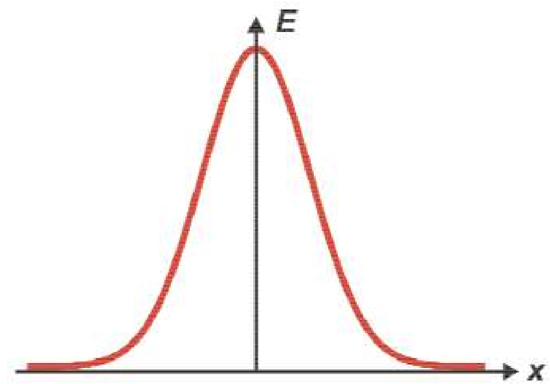
Two sets of parameters $\{\mu, \Sigma\}$, which again generate a family of models, $\mathcal{F} = \{p(x|\mu,\Sigma) \mid \mu \in R^n, \Sigma \in R^{n\times n}\ and\ PSD\}$ ,

# Estimating Gaussian Distributions

- 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 $\mu, \sigma$




- Given $m$ iid samples

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

- Likelihood of one data point:

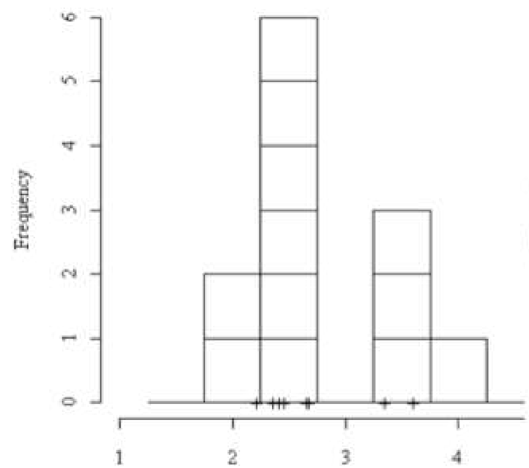
$$p(x^i|\mu,\sigma) \propto exp\left(-\frac{1}{2\sigma^2}(x^i-\mu)^2\right)$$

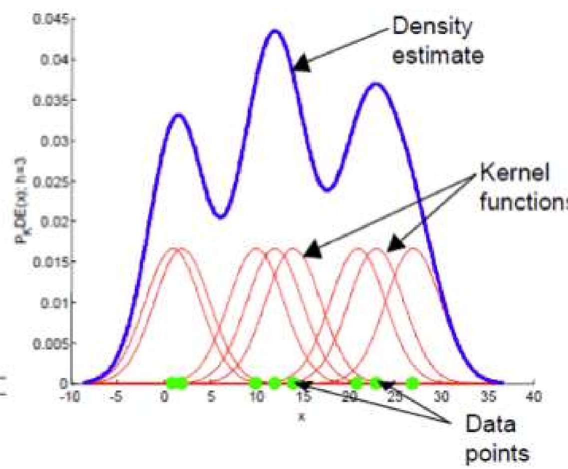
# Nonparametric Density Estimation

- 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 parameters

- Eg. Histogram

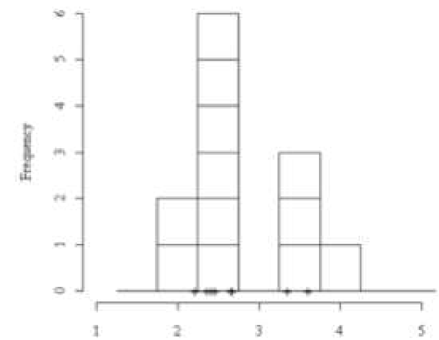
- Eg. Kernel density estimator



# Histograms

- 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 $x$

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

# Kernel Density Estimation

- Kernel density estimator

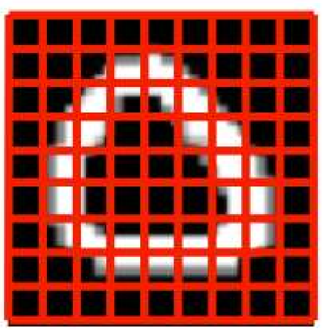
$$p(x) = \frac{1}{m}\sum_{i}^{m}\frac{1}{h}K\left(\frac{x^i - x}{h}\right)$$

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

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

# Dimension Reduction

- The process of reducing the number of random variables under consideration
  - One can combine, transform or select variables
  - One can use linear or nonlinear operations



$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

vector in $R^d$

Original data point

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix}$$

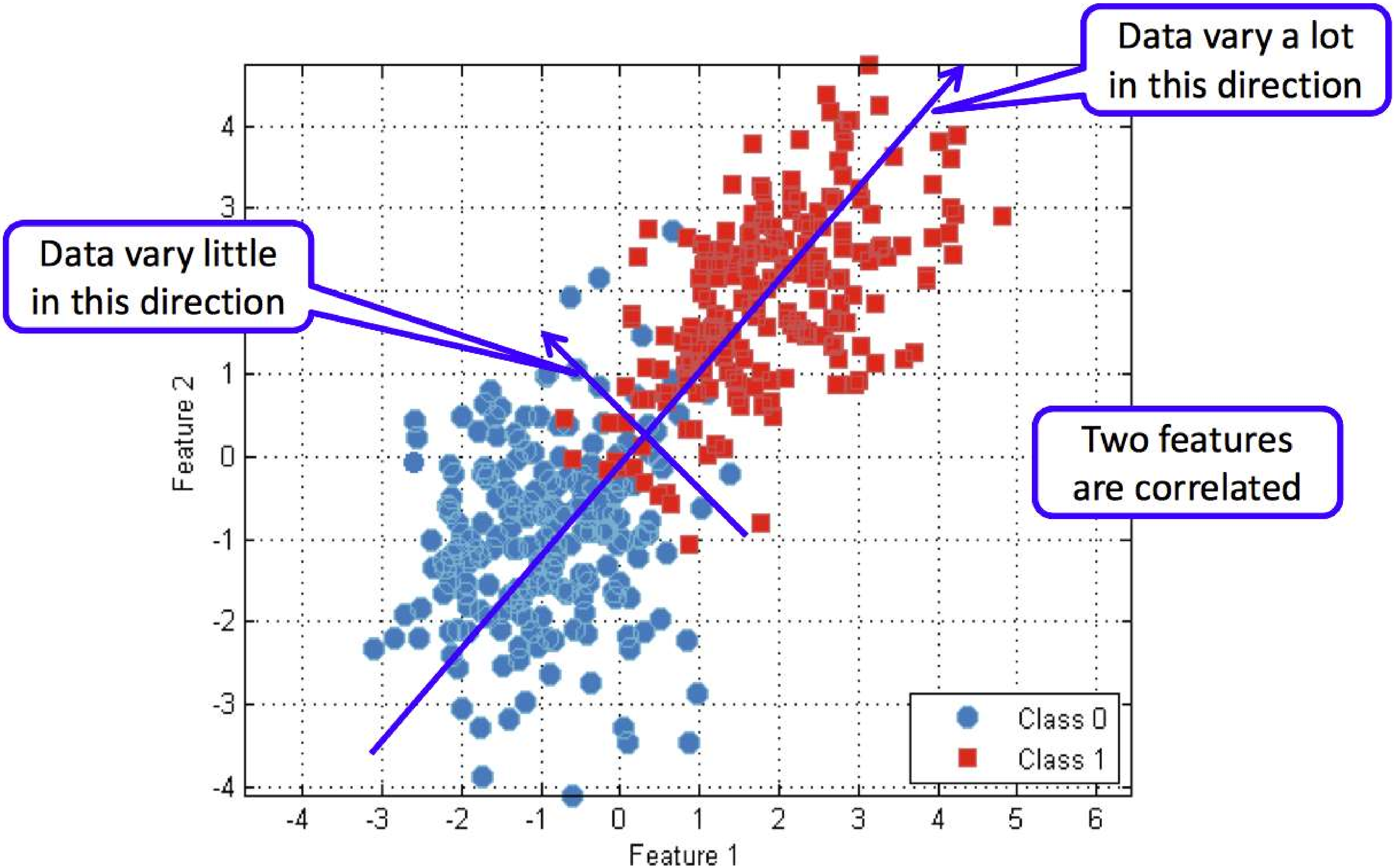
$$f(x): R^d \mapsto R^k$$

$$k \ll d$$

Reduced representation

$$z = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \end{pmatrix}$$

# Principal Component Analysis



# Formulating the Problem

- Given $m$ data points, $\{x^1, x^2, \dots x^m\} \in R^n$, with their mean $\mu = \frac{1}{m}\sum_{i=1}^{m} x^i$

- Find a direction $w \in R^n$ where $\|w\| \leq 1$

- Such that the variance (or variation) of the data along direction $w$ is maximized

$$\max_{w:\|w\|\leq 1} \underbrace{\frac{1}{m}\sum_{i=1}^{m}\left(w^\top x^i - w^\top \mu\right)^2}_{\text{variance}}$$

# The PCA Algorithm

- Given $m$ data points, $\{x^1, x^2, \dots x^m\} \in R^d$, with mean

- Step 1: Estimate the mean and covariance matrix from data

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^i \quad and \quad C = \frac{1}{m}\sum_{i=1}^{m}(x^i - \mu)(x^i - \mu)^\top$$

Principal directions

- Step 2: Take the eigenvectors $w^1, w^2, \dots$ of $C$ corresponding to the largest eigenvalue $\lambda_1$, the second largest eigenvalue $\lambda_2$ ...

- Step 3: Compute reduced representation

$$z^i = \begin{pmatrix} {w^1}^\top (x^i - \mu)/\sqrt{\lambda_1} \\ {w^2}^\top (x^i - \mu)/\sqrt{\lambda_2} \\ \vdots \end{pmatrix}$$

Normalize by standard deviation

# PCA and SVD

- $M = [u_1 \quad u_2 \ldots \quad u_n] \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1n} \\ \vdots & \ddots & \vdots \\ \Sigma_{m1} & \cdots & \Sigma_{mn} \end{bmatrix} [v_1 \quad v_2 \ldots \quad v_n]^\top$

principal directions

Scaling factor

Projection in principal directions

- Singular value decomposition is related to eigenvalue decomposition
  - Suppose $X = [x_1 - u \quad x_2 - u \ \ldots \quad x_m - u] \in \mathbb{R}^{m \times n}$
  - Then covariance matrix is $C = \frac{1}{m} X X^\top$
  - Starting from singular vector pair
    - $M^\top \mathrm{u} = \sigma v$
    - $\Rightarrow MM^\top \mathrm{u} = \sigma M v$
    - $\Rightarrow MM^\top \mathrm{u} = \sigma^2 u$
    - $\Rightarrow Cu = \lambda u$