Lecture 07
Clustering & Dimension Reduction

Oran Kittithreerapronchai[1]

[1]Department of Industrial Engineering, Chulalongkorn University
Bangkok 10330 THAILAND

last updated: August 7, 2023

Outline

source: General references [NC20, TSK16, BG19, Pat14]

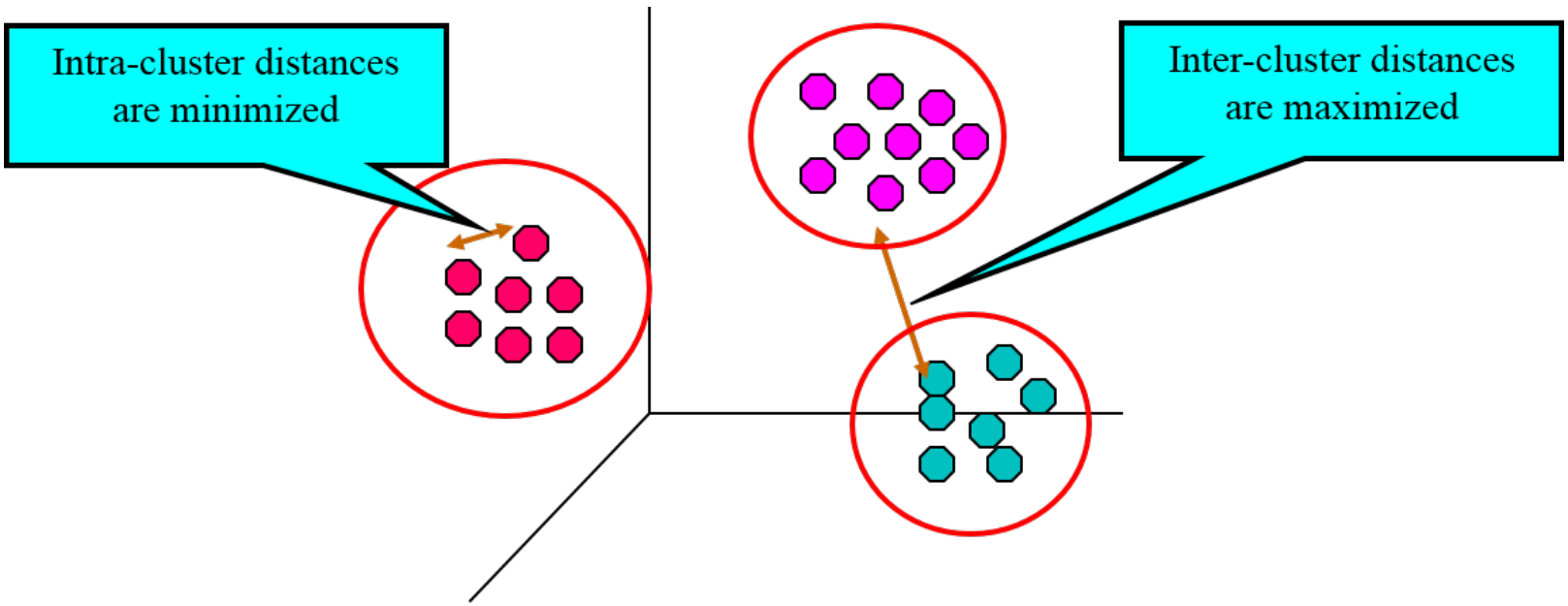
What is Clustering?

- **What:** labeling similar data points into groups
- **Important:** find features, screen outlier, generalize knowledge, verify knowledge
- **Terms:**
  - **Agglomerative** series of clusters linked to create large cluster
  - **Divisive (Partitioning)** exclusive clusters devising into independent group
  - **Distance (Similarity)** value indicating similarity of each point
  - **Hierarchical** points connected into clusters

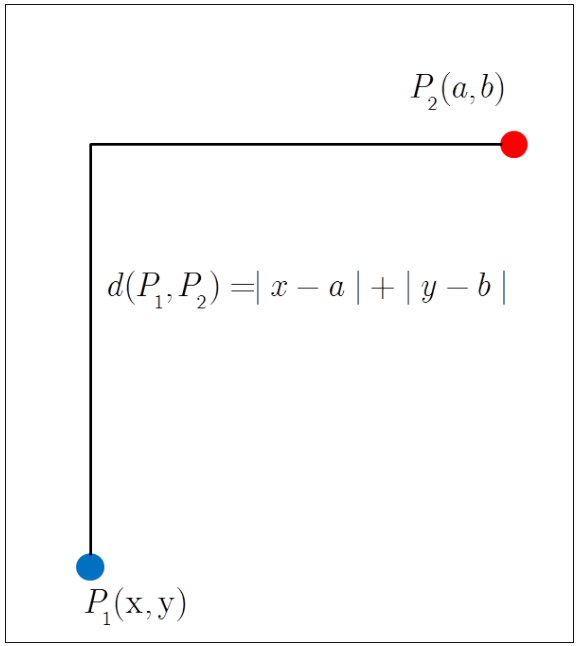
Clustering Task

Definition

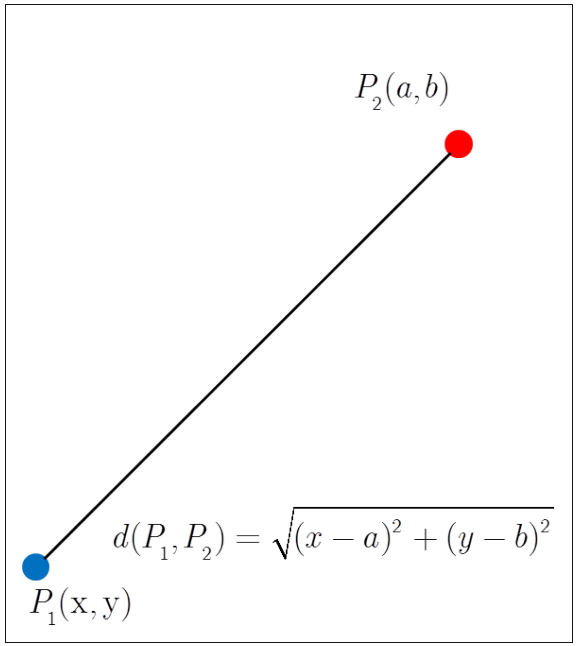
Find groups such that the one in a group are similar another in the group, but different from those in other groups



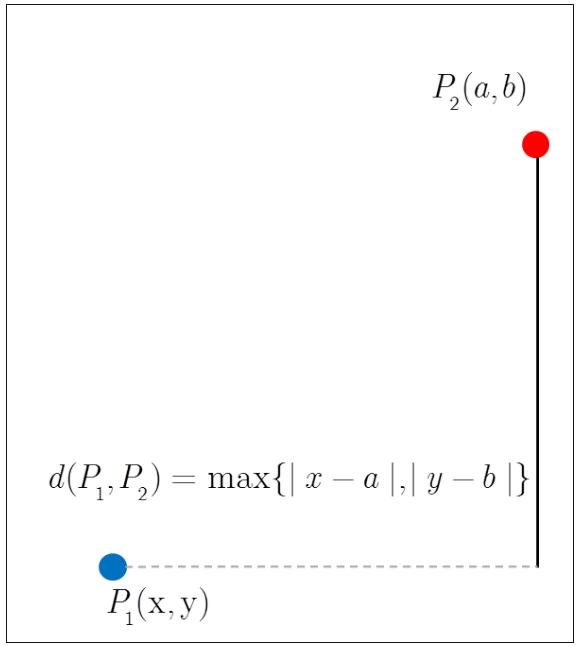
Distance Measurement



Rectilinear (L_∞)



Straight Line (L_2)



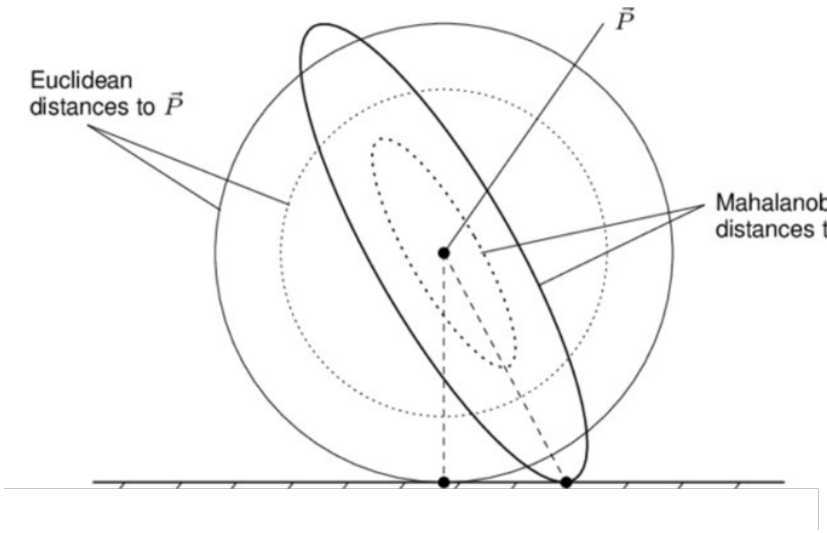
Chebyshev (L_1)

Minkoski distance (p)

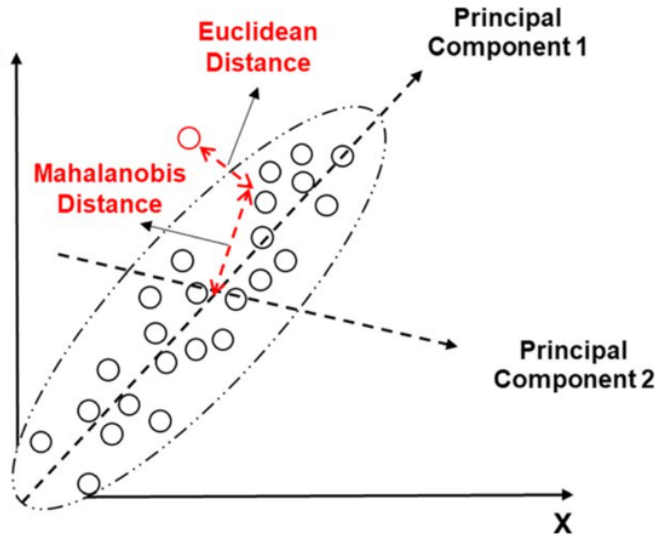
$$L_p(O_1, P_2) =^p \sqrt{(x-a)^p + (y-b)^p}$$

Scaling and Mahalanobis Distance

- **What:** Generalization of Euclidian distance in multivariate that normalizes and consider covariance



Comparison with Euclidian distance



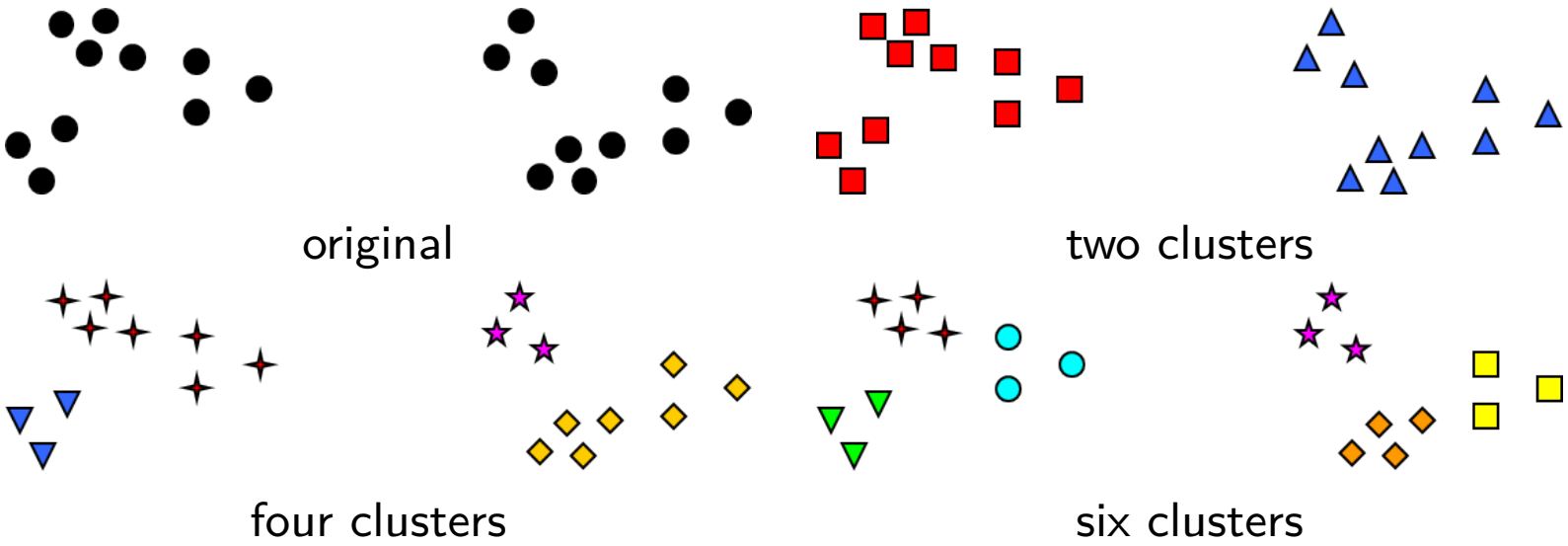
Relationship with PCA

Mahalanobis distance (d_m)

$$d_m(\mathbf{x}, \mathbf{Q}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \mathbb{S}^{-1} (\mathbf{x} - \boldsymbol{\mu})}$$

where $\boldsymbol{\mu}$ and $\mathbb{S}^{-1}$ are mean and inverse of covariance matrix $\mathbb{Q}$

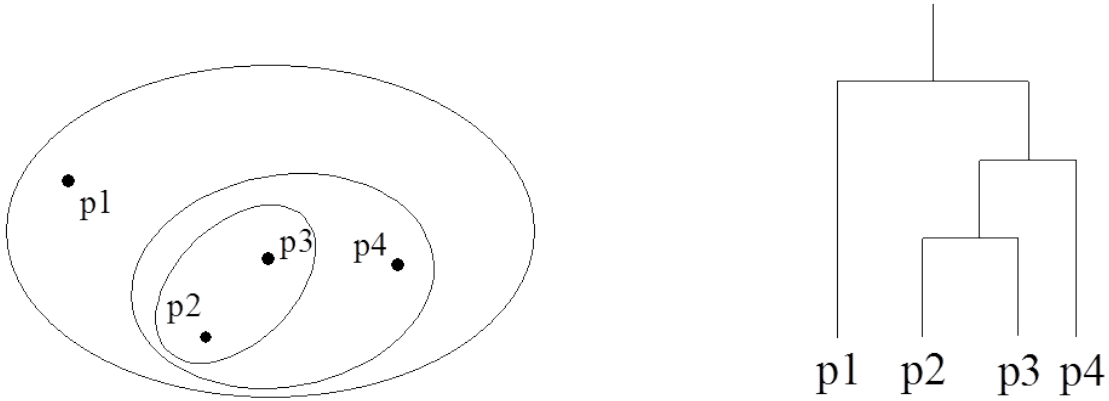
How many cluster?



Types of Clustering

Partition clustering: one object is in exactly one cluster

Hierarchical clustering: one object is in many clusters (organized as tree)

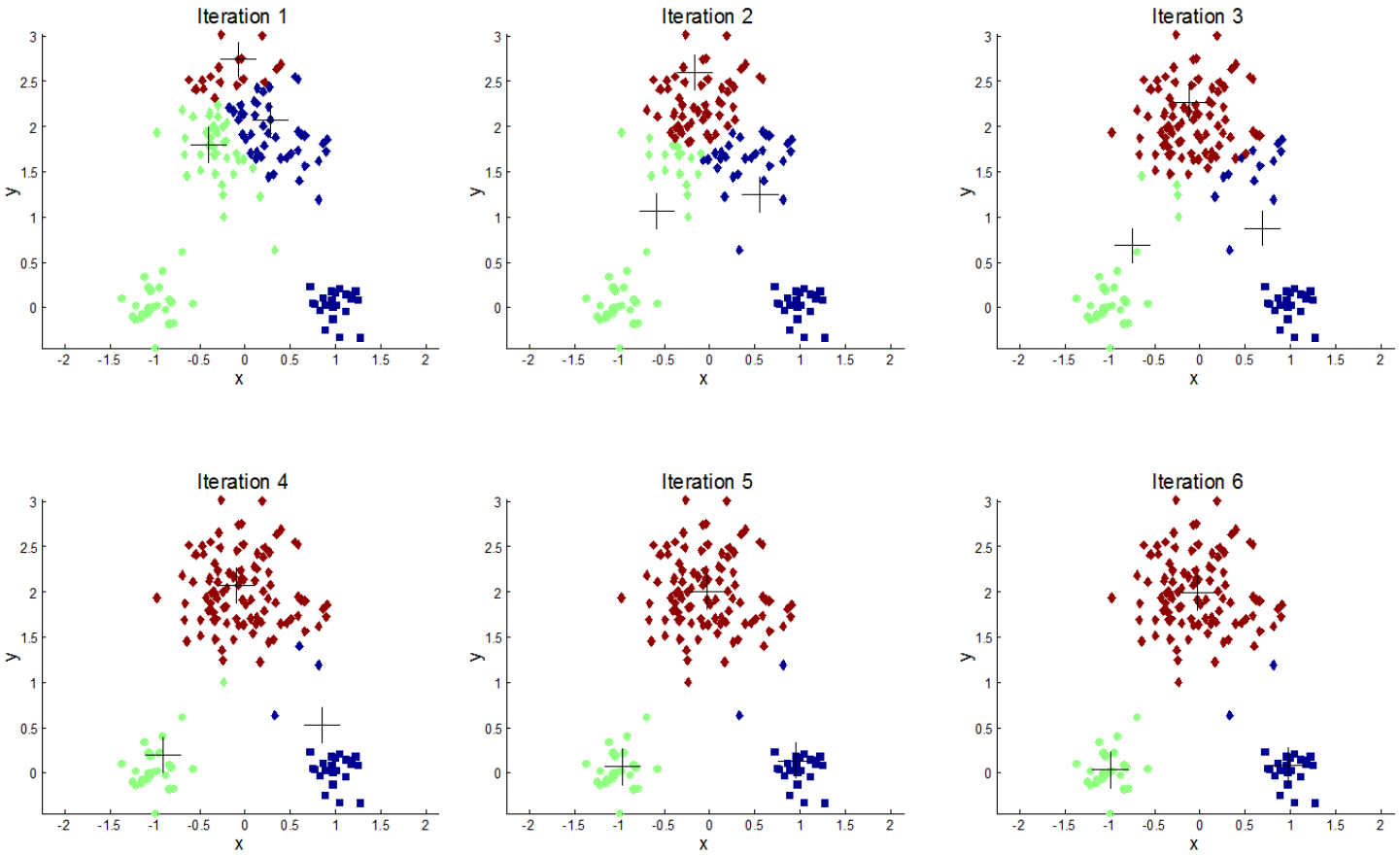


Partition clustering: k-means

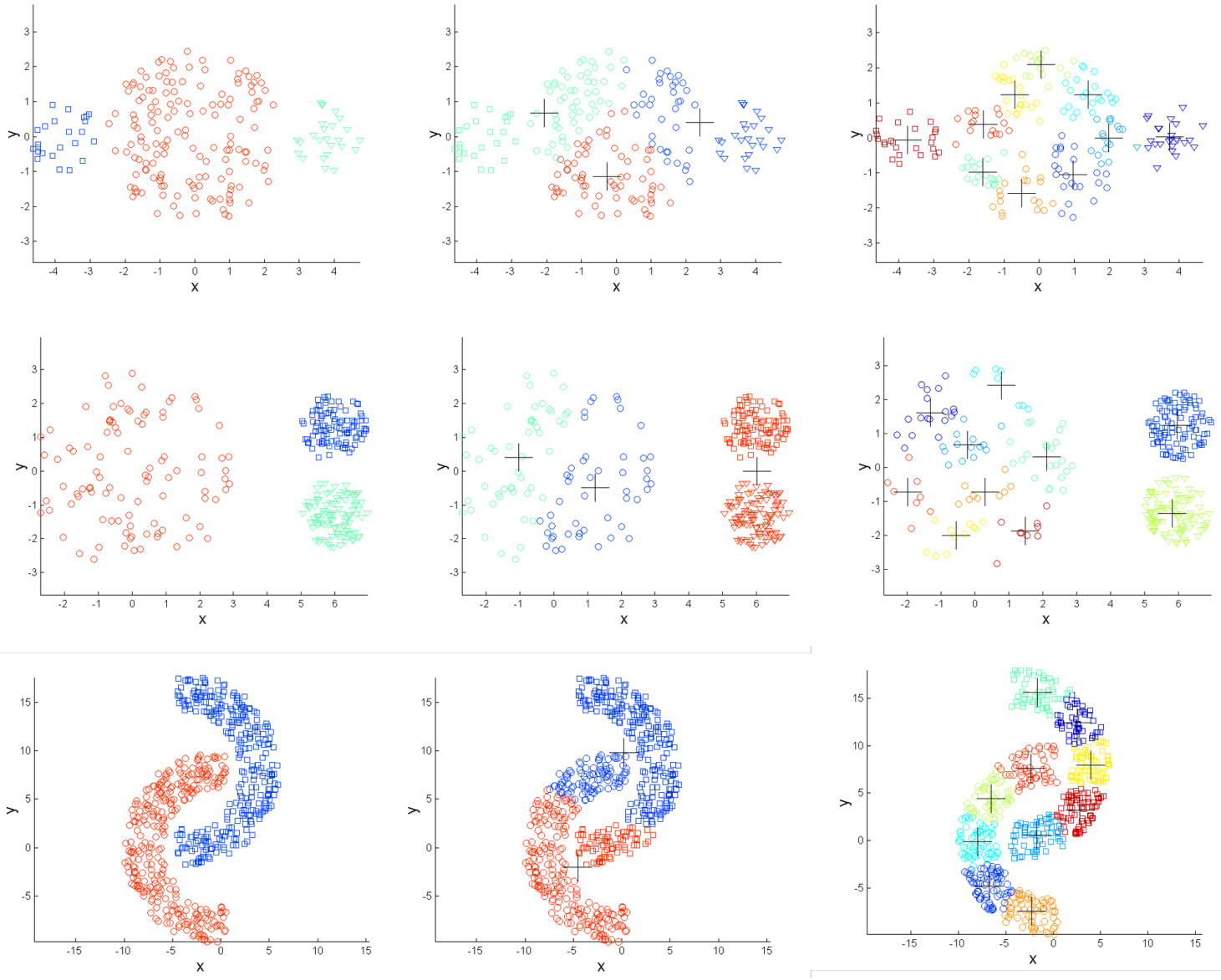
- **Important:** a simple and easy implement clustering class
- **Explanation:** Given k clusters representing by centroid (center point), assign each point to cluster with the closest centroid
- k-**mean algorithm**

  randomly select k distinguish centroid points
  repeat
    assign data points to the "nearest" centroid
    re-calculate all centroid points
  until no centroid point change
- **Quality of cluster:** sum of distances from its cluster
- **Benefits:** simple algorithm $\rightarrow$ fast & easy
- **Issue:** distance function; how to initially select

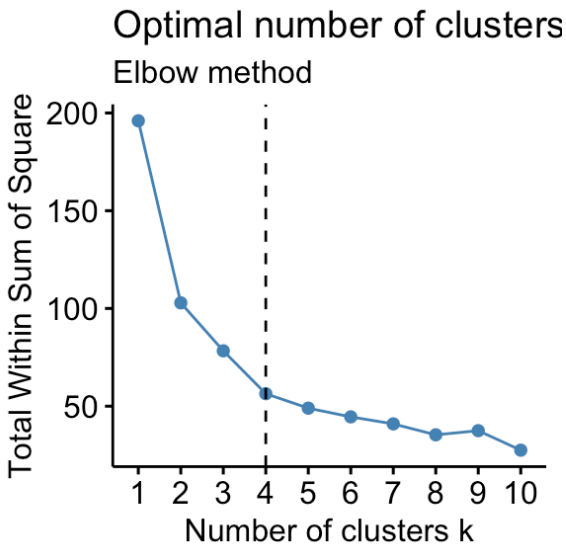
Example of k-mean



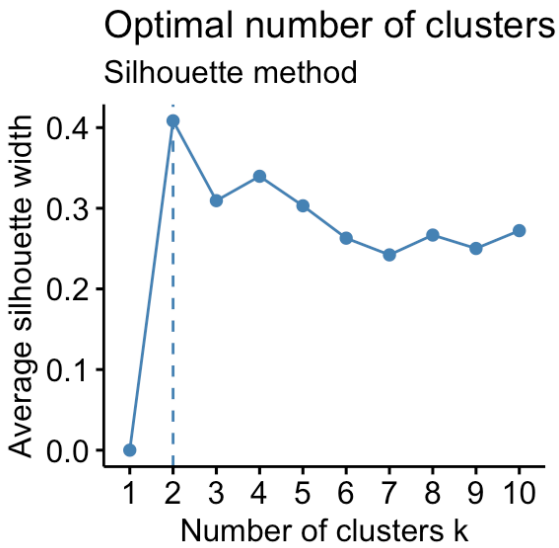
Problems with k-mean



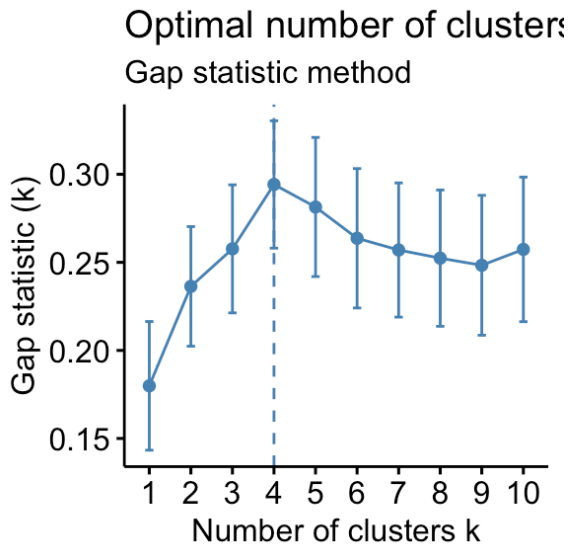
What is a suitable k?



Elbow method



Silhouette method



Gap statistic method

Minimizing Intra-Cluster Variance

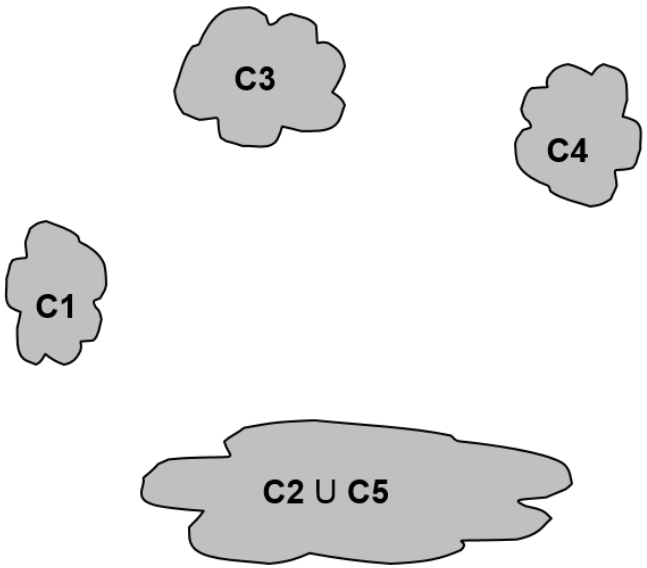
- **Elbow method:** minimizing within-cluster sum of square (WSS)
- **Silhouette method:** maximizing quality of clustering of dataset within cluster
- **Gap statistic method:** maximizing difference between cluster k and NULL reference distribution

Hierarchical Clustering: Agglomerative

- **Explanation:** Start with individual point and merge proximity points to form cluster
- **Agglomerative algorithm**

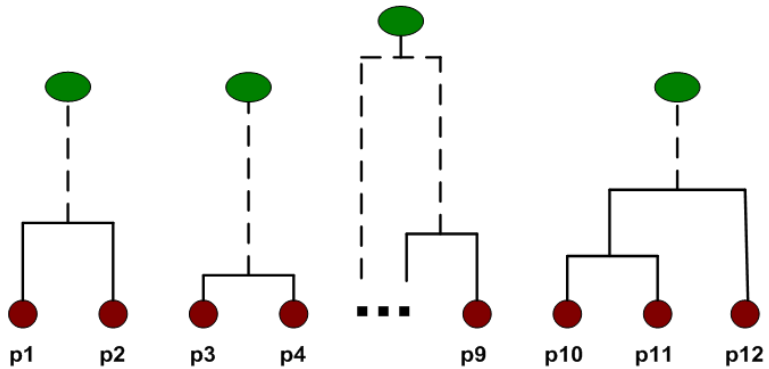
  compute proximity matrix
  each point = individual cluster
  repeat
    merge the two closest clusters
    update the proximity matrix
  until only a single cluster remains
- **Benefits:** no assumption on k; meaningful taxonomies
- **Issue:** how to define similarity

Example of Agglomerative



	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



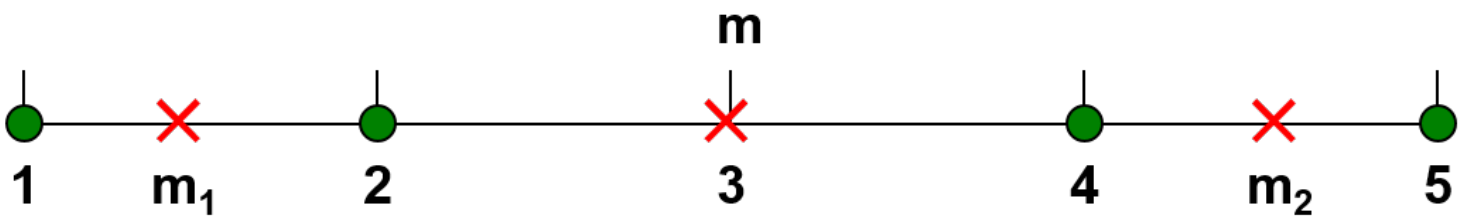
Cluster Validation

- **Cluster Cohesion:** measures how closely related are objects in a cluster, such as

$$WSS = \sum_k \sum_{i \in \mathcal{S}(k)} d(p_i, m_k)^2$$

- **Cluster Separation:** measure how distinct or well-separated a cluster is from other clusters

$$BSS = \sum_k \sum_{i \in \mathcal{S}(k)} |\mathcal{S}(k)| d(m, m_k)^2$$



Example

k	$\mathcal{S}(k)$	m_k	BSS
1	$\{1, 2, 4, 5\}$	$\{3\}$	$\|4\|(3-3)^2 = 0$
2	$\{1, 2\}$, $\{4, 5\}$	$\{1.5, 4.5\}$	$\|2\|(1.5-3)^2 + \|2\|(4.5-3)^2 = 9$

Time of R: zoo from UCI

- **Base:**

```
iris.kmean1 <- kmeans(iris[,1:4],centers = 4)
iris.kmean2 <- kmeans(scale(iris[,1:4]),centers = 4)
xtabs(~iris.kmean1$cluster+iris[,5])
xtabs(~iris.kmean2$cluster+iris[,5])
```

- **Find *k*:**

```
require(factoextra)
require(NbClust)

fviz_nbclust(iris[,1:4], kmeans, method = "wss")        ## elbow method
fviz_nbclust(iris[,1:4], kmeans, method = "silhouette")
fviz_nbclust(iris[,1:4], kmeans, nstart = 25,  method = "gap_stat", nboot = 50)
```

- pam

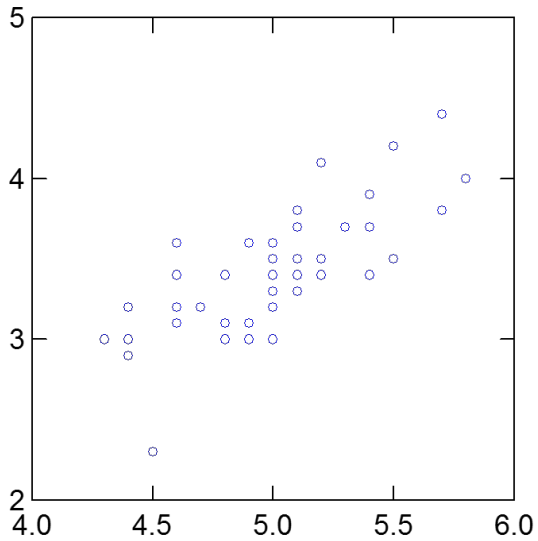
```
require(cluster)
iris.pam <- pam (iris[,1:4],k = 4)
plot(iris.pam,which.plots = 1)
```

- agnes

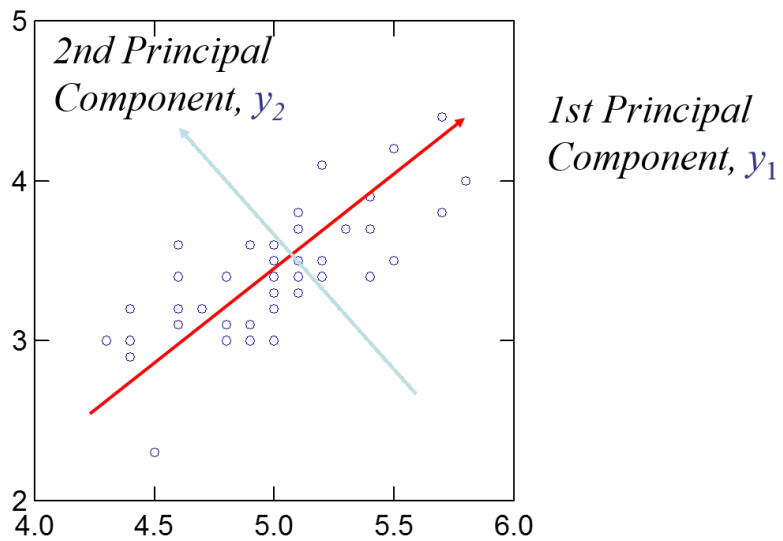
```
require(cluster)
iris.agnes <- agnes (iris[,1:4])
cutree(iris.agnes,k=4)
plot(iris.agnes)

exam.mona <- mona(exam)  ### for binary like our exam
plot(exam.mona)
```

Principal Component Analysis

Principal component analysis (PCA) is a transformation procedure that convert correlated data into a set of linearly uncorrelated terms

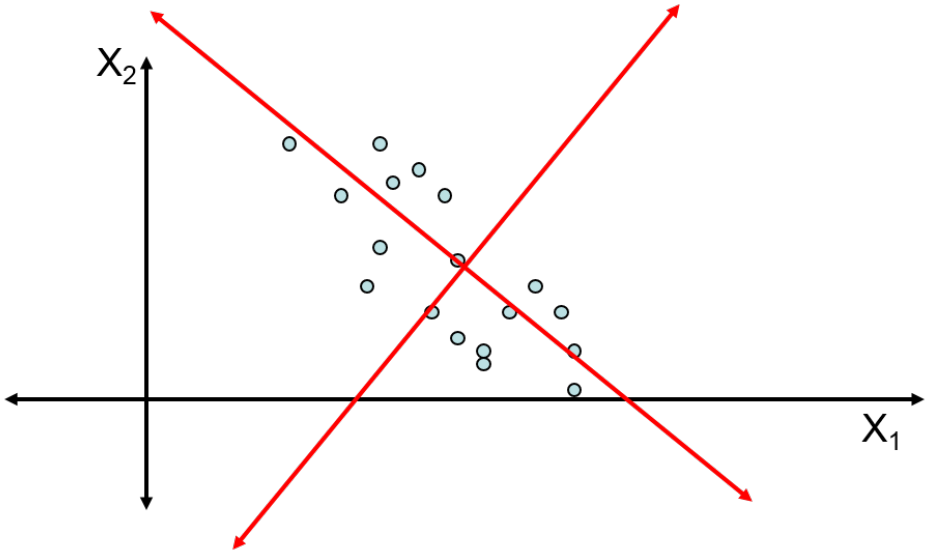


origin



pca

Linear Transformation of PCA



Desired Property of PCA

- **shift:** re-center values of attributes $\rightarrow$ **0**
- **rotate:** first components explains a lot of information
- **orthogonal:** each component independence of others
- **linear subspace:** minimal sum square error of projection
- **max** 1^{st} **Var:** $Var(\mathbf{X}) = Var(\mathbf{Y})$ Eigenvalue Decomposition

PCA: Matrix Algebra

- **Notation:**
  - $\mathbf{X}$ matrix of original data $p \times n$ that center at $\mathbf{0}$
  - $\mathbf{w}$ unit vector for projection each component
  - $\mathbf{Y}$ matrix of orthogonal projection, $\mathbf{Y} \equiv \mathbf{X} \cdot \mathbf{w}$

  Consider 1^{st} component:
- **Max Var. of Project:** $\max \mathbf{Y^T Y} = \max (\mathbf{X} \cdot \mathbf{w})^{\mathbf{T}} (\mathbf{X} \cdot \mathbf{w})$
- **Subject to:** $\mathbf{w^T} \cdot \mathbf{w} = 1$

apply FOC;

$$(\mathbf{X^T} \cdot \mathbf{X})\mathbf{w} - \lambda \mathbf{w} = \mathbf{0}$$

Hence,
$\mathbf{w}$ is 'principle' (max eigenvector) eigenvector

princomp() VS prcomp()

- **Two Commands:** different default
  - `princomp(x,cor=F)`: not centering and not scaling
  - `prcomp(x,cor=F,centering=T,scale=F)`: do centering, but not scaling| not report center

Output of PCA

prcomp()	princomp()	description
sdev	sdev	sd of PCA
rotation	loadings	convert vector (eigenvectors)
center	center	means subtracted
scale	scale	sd scaled
x	scores	new coordinate

Time of R: iris example

- **Verify:**

```
iris.temp <- apply(iris[,1:4],2,function(o) o-mean(o) )
sum(iris.temp[,4])

iris.eigen <- eigen(cov(iris.temp))
iris.princ <- princomp (iris.temp)
iris.princ$loadings

head(iris.temp %*% iris.eigen$vector)
head(iris.princ$scores)
```

- **Alternative:**

```
iris.prcom <- prcomp(iris[,1:4],scale = F)
iris.prcom$rotation
```

- **Visual:**

```
plot(iris.princ)
biplot(iris.princ,cex=0.5)
points(iris.princ$scores[,1],iris.princ$scores[,2], cex=0.5,pch=16,col=iris[,5])
```

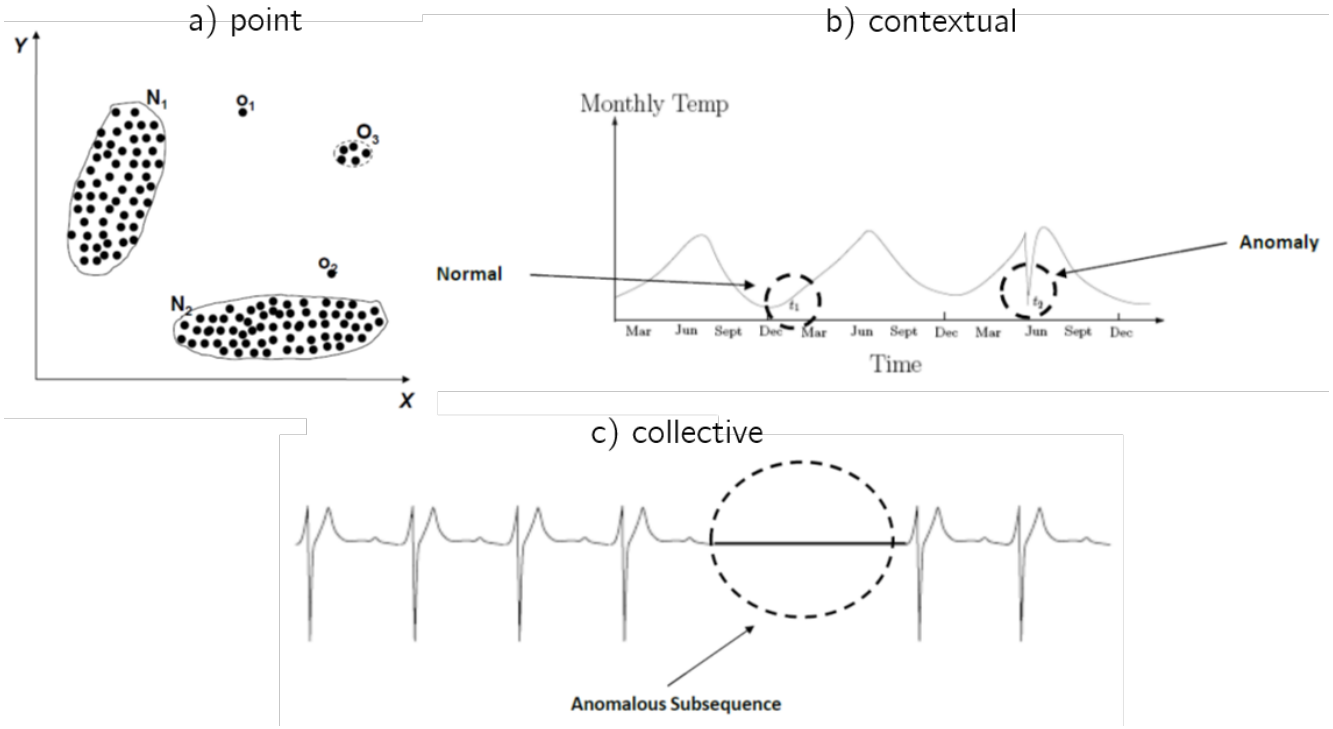
What is Anomaly Detection?

The truly creative people tends to be outliers

source: Nolan Bushnell

- **What:** identify rare events that different to normal
- **Important:** find outliers & uncommon features/exception
- **Example:**
  - **Credit Card Fraud:** abnormally purchase of value, item, location
  - **Cyber Intrusions:** computer virus spread over Internet
- **Label:** How to know that data anomaly or normal
  - **Supervised:** have labels for both; known anomaly $\rightarrow$ classification
  - **Unsupervised:** no labels assumed; rare anomaly $\rightarrow$ clustering
  - **Semi-Supervised** limited labels; ??? anomaly $\rightarrow$ classification & clustering

Type of Anomaly

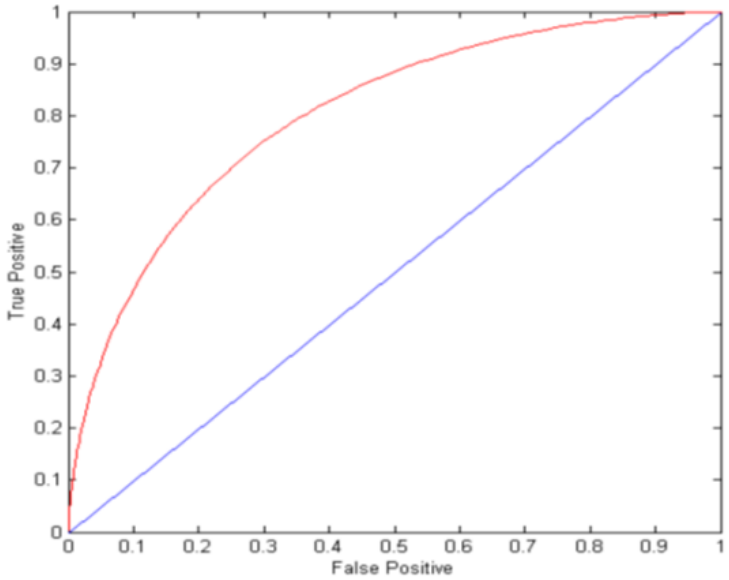


- **Point:** foreign w.r.t static data
- **Condition/Context:** foreign to condition (required context)
- **Collective:** not foreign by itself, (require pattern)

Detecting Technic

- **Define Normal Profile:** profile = pattern, conditions of normal data
- **Apply Profile:** anomalies are observations that significantly differ from normal profile
- **Scheme Method**
  - **Statistical-based :** normal = distribution, e.g., Benford's law,
  - **Distance-based :** normal $\in$ group, e.g. Local Outlier $\ll$**to-be-research** $\gg$
  - **Model-based :** normal $\approx$ predicted model $\ll$**to-be-research** $\gg$

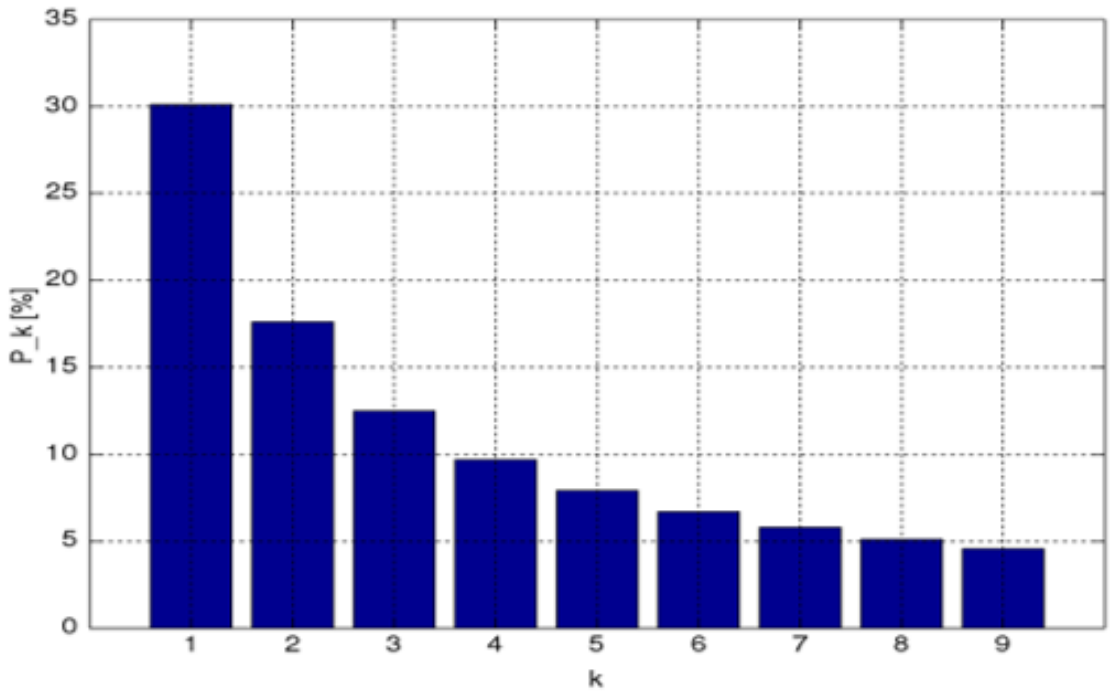
ROC: Evaluating Detection Technique



- **ROC:** a plot of TPR (y-axis) VS FPR (x-axis) at different threshold
- **Importance** effect of changing threshold (point)/algorithm(line) to detect anomaly
- **Uses:**
  - **point** $(0,0)$ all anomaly ; $(1,1)$ all normal ; $(1,0)$ ideal
  - **Diagonal Line:** random guessing
  - **Note :** True-Positive = correctly detect anomaly

Benford's law: first digit law

- **What:** counter intuitive observation on frequency of first leading digit
- **Detail:** frequency of each digit $\neq$ uniform.
- **Particularly** '1' (.30) $>$ '2' (.17) $> \cdots >$ '6' (0.05) $\approx$ '9' (0.05)
- **Important:** detect random/frabicated data
- **Rational:** At constant power/scale, frequency at each digit are not equal
- **example:** Fibonacci, Expo seq, log-normal distribution



Reference

[BG19] Brad Boehmke and Brandon M Greenwell.
Hands-on machine learning with R.
CRC press, 2019.

[NC20] Fred Nwanganga and Mike Chapple.
Practical machine learning in R.
John Wiley & Sons, 2020.

[Pat14] Manas A Pathak.
Beginning data science with R.
Springer, 2014.

[TSK16] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar.
Introduction to data mining.
Pearson Education India, 2016.