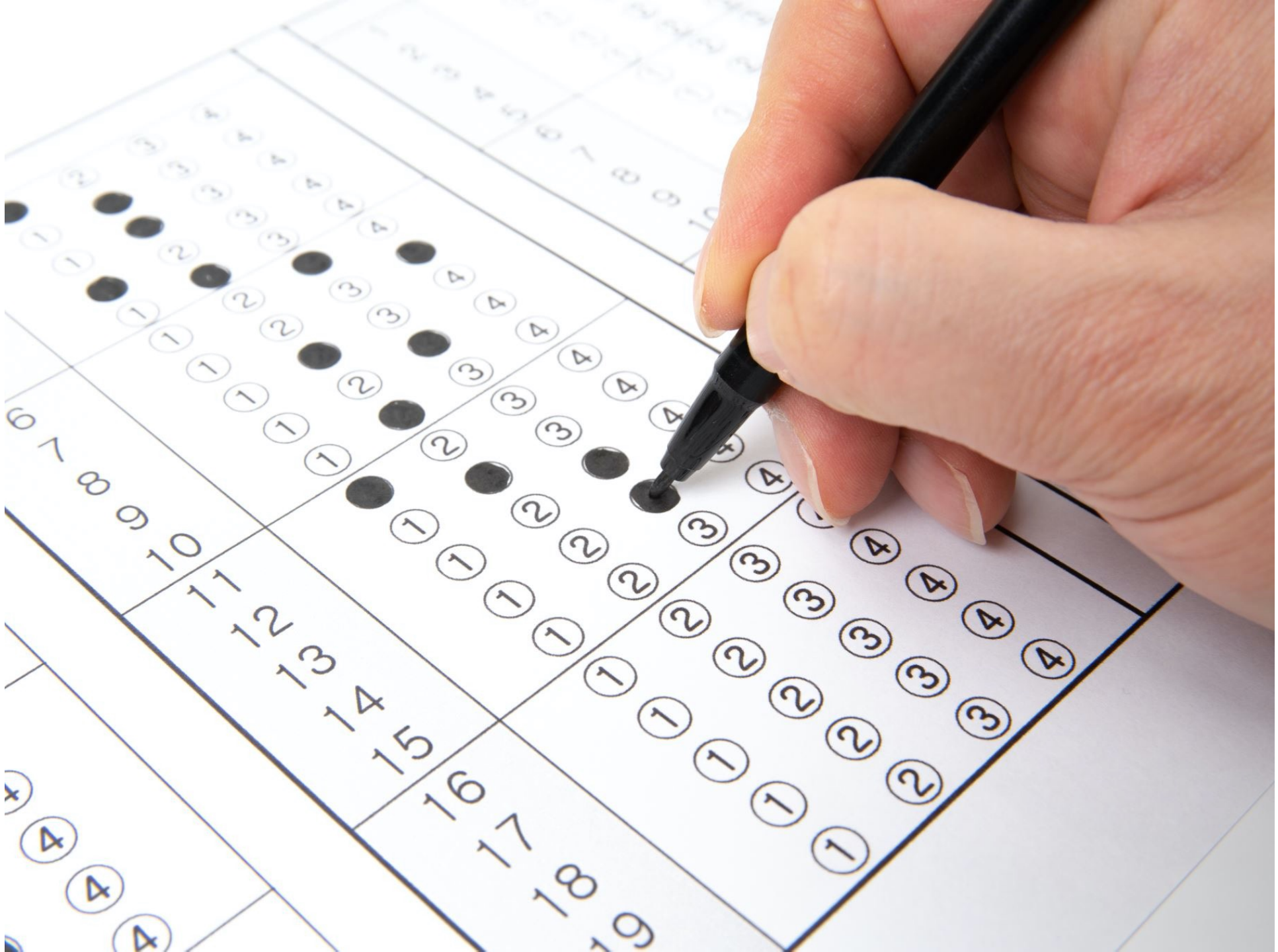


*) Boosting / Gr. Boost. degr.

$D = \{x_i, y_i\}_{i=1}^{n} \rightarrow$ fit a model $F(x)$

Residual $y_i - F(x_i)$

Fit $h(x)$ on $\{x_i, y_i - F(x_i)\}_{i=1}^{n}$

$$\left\{x_i, -\frac{\partial I}{\partial F(x_i)}\right\}$$

$$I = \frac{1}{n}\sum_{i=1}^{n}[y_i - F(x_i)]^2 \quad \left| \quad \frac{1}{n}\sum_{i=1}^{n}|y_i - F(x_i)| \right.$$

Unsupervised learning
Clustering

From MIT and CMU

Before we begin…

- Suppose you got into xyz company. Your manager
- Gives you a data comprising of company' customers like – age, work profile, salary, hobby, recent purchase history, duration of website visit.
- Asks you to come with some interesting pattern using the data about customers.

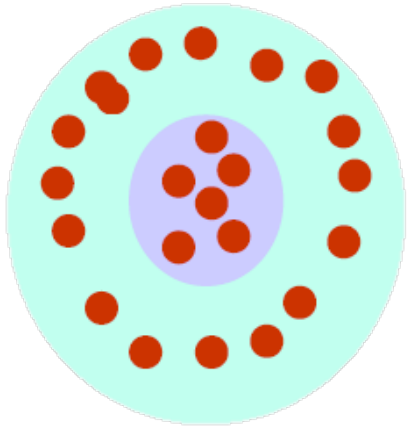
- Do you think if this is a classification/regression problem setup?
- Is there any notion of class/label present?
- Can you apply supervised technique?

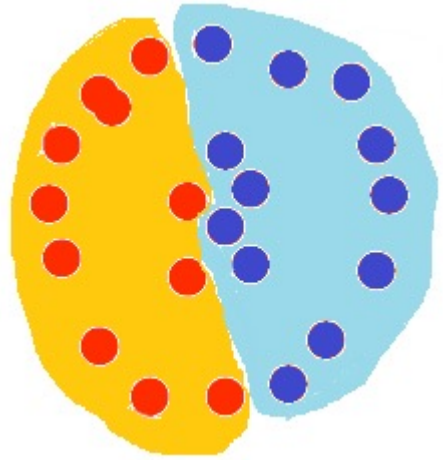
Outline

- Introduction to clustering
- K-means
- ~~Hierarchical clustering~~

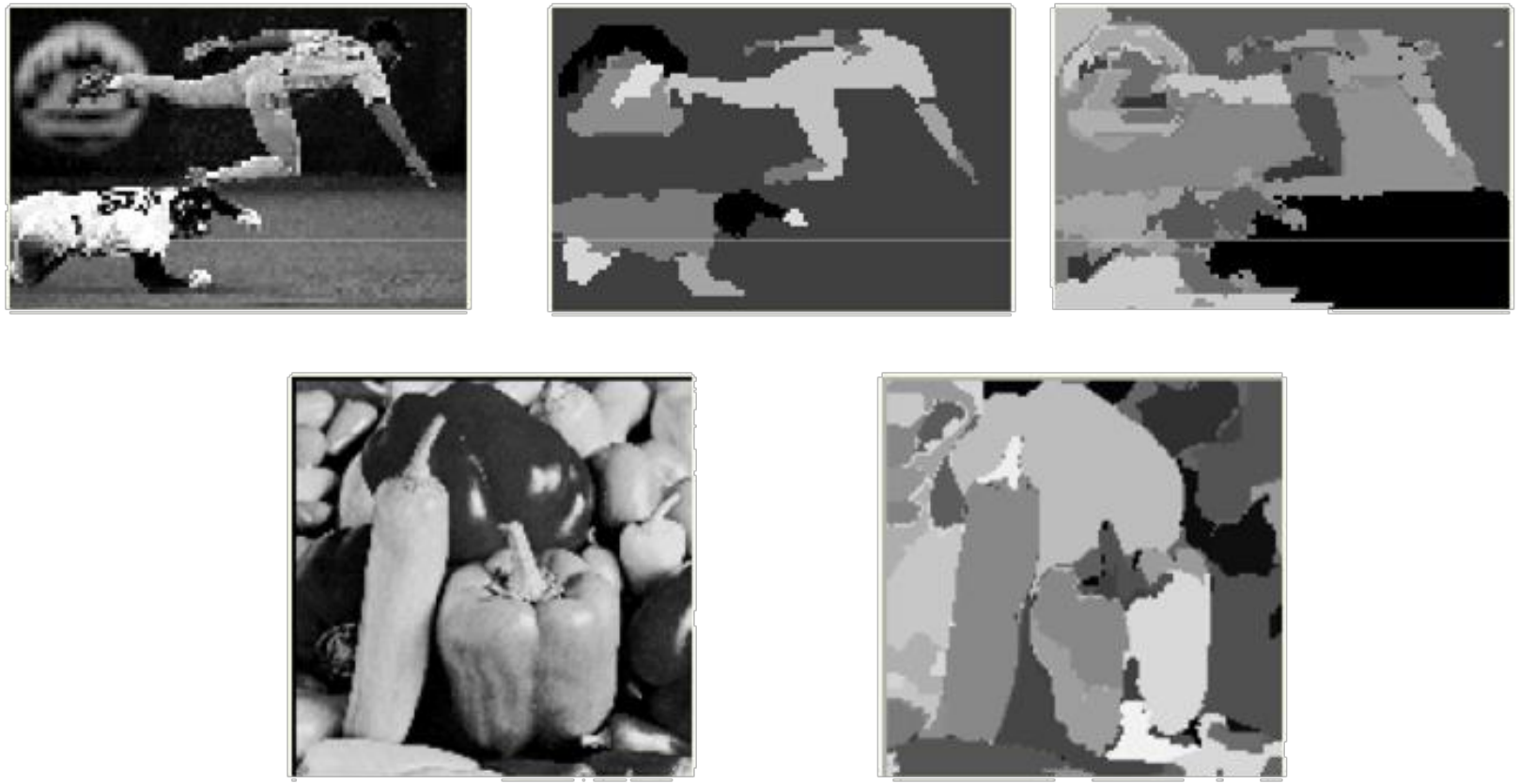
What is clustering?

- The organization of unlabeled data into similarity groups called clusters.
- A cluster is a collection of data items which are “similar” between them, and “dissimilar” to data items in other clusters.





Computer vision application: Image segmentation



From: Image Segmentation by Nested Cuts, O. Veksler, CVPR2000

Other applications

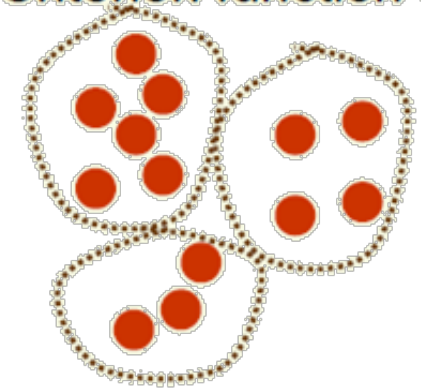
- Google news links
- Biological cell segmentation
- Astronomical image segmentation

- Can you think of other applications?

What do we need for clustering?

1. Proximity measure, *either*
   - similarity measure $s(x_i,x_k)$: large if x_i,x_k are similar
   - dissimilarity(or distance) measure $d(x_i,x_k)$: small if x_i,x_k are similar

2. Criterion function to evaluate a clustering



3. Algorithm to compute clustering
   - For example, by optimizing the criterion function

$p \to \infty \to l_\infty$ distance. $\max\{|x_{jk} - x_{ik}|\}$

Distance (dissimilarity) measures

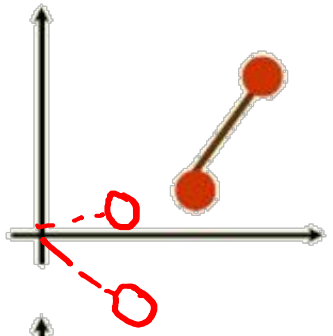
$x_i + d$, $x_j + d$

- Euclidean distance

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{d} (x_i^{(k)} - x_j^{(k)})^2}$$

l_2

  - translation invariant rotation?

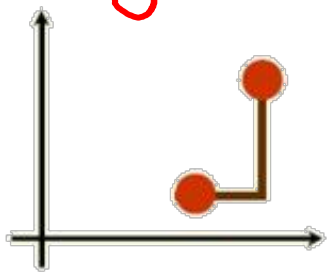


- Manhattan (city block) distance

$$d(x_i, x_j) = \sum_{k=1}^{d} |x_i^{(k)} - x_j^{(k)}|$$

l_1

  - approximation to Euclidean distance, cheaper to compute



- They are special cases of **Minkowski distance**:

$$d_p(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{k=1}^{m} |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}}$$

(p is a positive integer)

$x_i, x_j \in R^m$

$p=2$, l_2 dist.

$=1$, l_1 dist.

Distance definition

So what makes a good distance? There are two aspects to the answer to this question. The first is that it captures the "right" properties of the data, but this is a sometimes ambiguous modeling problem. The second is more well-defined; it is the properties which makes a distance a metric.

A distance $\mathbf{d} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ is a bivariate operator (it takes in two arguments, say $a \in \mathcal{X}$ and $b \in \mathcal{X}$) that maps to $\mathbb{R}^+ = [0, \infty)$. It is a *metric* if

(M1) $\mathbf{d}(a, b) \geq 0$ **(non-negativity)**

(M2) $\mathbf{d}(a, b) = 0$ if and only if $a = b$ **(identity)**

(M3) $\mathbf{d}(a, b) = \mathbf{d}(b, a)$ **(symmetry)**

(M4) $\mathbf{d}(a, b) \leq \mathbf{d}(a, c) + \mathbf{d}(c, b)$ **(triangle inequality)**

A distance that satisfies (M1), (M3), and (M4) (but not necessarily (M2)) is called a *pseudometric*.

A distance that satisfies (M1), (M2), and (M4) (but not necessarily (M3)) is called a *quasimetric*.

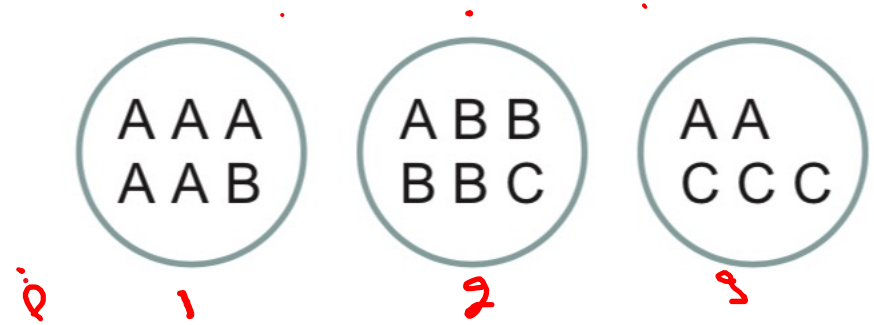
K-L divergence $D_{P||Q}$

https://www.cs.utah.edu/~jeffp/teaching/cs5955/L7-Distances.pdf

Evaluating the output of clustering methods

- Let N_{ij} be the number of objects in cluster i that belong to class j, and
- Give me the no. of object in cluster 2 that belong to class 2?
- $N_{22} = 4, N_{?2} = 0$

- let $N_i = \sum_{j=1}^{C} N_{ij}$ be the total number of objects in cluster i.



Classes – A, B, and C
j 1, 2, and 3

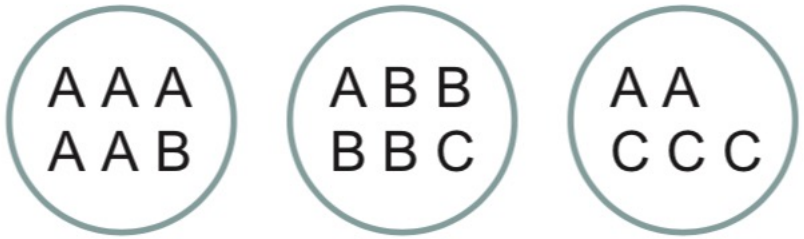
Evaluating the output of clustering methods (contd.)

- Define $p_{ij} = N_{ij} / N_i$; this is the empirical distribution over class labels for cluster i. It denotes proportion of a class.

- We define the **purity** of a cluster as $p_i \triangleq \max_j p_{ij}$, ie proportion of class with max samples

$p_1 \triangleq \max_j p_{1j}$ $p_2 \triangleq \max_j p_{2j} = 4/6$

$p_1 = \max\{p_{11}, p_{12}, p_{13}\} = p_{11} = 5/6$



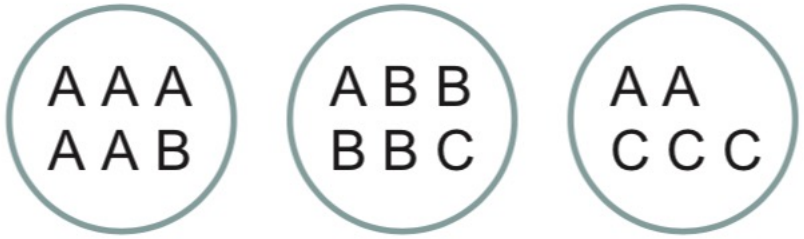
Evaluating the output of clustering methods (contd.)

Purity of clustering = $\frac{N_1}{N} p_1 + \frac{N_2}{N} p_2 + \frac{N_3}{N} p_3$

Purity = (0, 1]

- Purity of cluster 1? $\max(p_{11} = 5/6, p_{12} = 1/6, p_{13} = 0)$

$purity = \sum_{i=1}^{K} \frac{N_i}{N} p_i$



$$\frac{6}{17}\frac{5}{6} + \frac{6}{17}\frac{4}{6} + \frac{5}{17}\frac{3}{5} = \frac{5+4+3}{17} = 0.71$$

- Assume three clusters and three classes, what are the possible min and max purity of any given cluster?

Contd.

- The purity ranges between 0 (bad) and 1 (good).
- However, we can trivially achieve a purity of 1 by putting each object into its own cluster, so this measure does not penalize for the number of clusters.

Normalized Mutual Info

- **mutual information** (**MI**) of two random variables is a measure of the mutual dependence between the two variables.

- More specifically, it quantifies the "amount of information" (in units such as shannons, commonly called bits) obtained about one random variable through observing the other random variable.

- If $\bar{X}$ and $\bar{Y}$ are independent, then knowing $\bar{X}$ does not give any information about $\bar{Y}$ and vice versa, so their mutual information is zero.

MI: $I(X;Y) = 0$ X & Y ind.

If $Y = X$

$I(X;Y) = H(X)$

$H(Y)$

Entropy: $H(X) \Rightarrow$ entropy of r.v X

Suppose $X = \{1, 2, 3\}$

With prob. $p(X=1) = \frac{1}{2}$, $p(X=2) = \frac{1}{2}$, $p(X=3) = 0$

$$H(X) = -\sum_{i=1}^{|X|} p(x_i) \log p(x_i)$$

$|X| \rightarrow$ denoted cardinality of $X \rightarrow$ no. of elements.

$$H(X) = -\left[p(x=1)\log p(x=1) + p(x=2)\log p(x=2) \right.$$

~~$y \log \hat{y}$~~

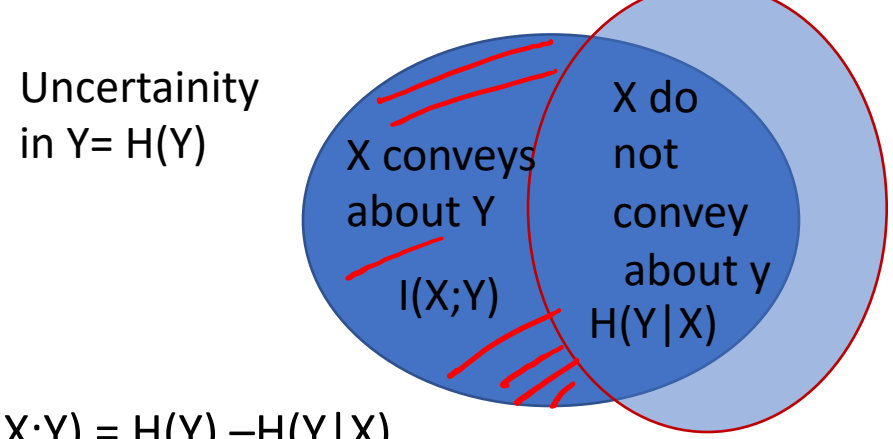
$$\left. + p(x=3)\log p(x=3) \right]$$

$$= -\left[\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2} + \underbrace{0\log 0}_{0} \right]$$

$$= 1 \text{ if log base 2}$$

1 bits

Entropy quantifies how much information/ randomness/ uncertainty is present in X



- I(X;Y) = H(Y) –H(Y|X)

- Intuitively, if entropy H(Y) is regarded as a measure of uncertainty about a random variable, then H(Y|X) is a measure of what X does not say about Y.

- This is "the amount of uncertainty remaining about Y after X is known", and thus the right side of the second of these equalities can be read as "the amount of uncertainty in Y, minus the amount of uncertainty in Y which remains after X is known", which is equivalent to "the amount of uncertainty in Y which is removed by knowing X".

- What is unknown about Y (H(Y))
- **minus**
- What is unknown about Y when X is observed H(Y|X)
- =
- What is known about Y when X is observed = I(X;Y)

Common flu – X and Covid - Y

- X={high fever, body ache, running nose}
- Y={X, respiratory issues}

- Y - cannot be seen but you can measure X using following test.

- Simple tests: for measuring fever- IR thermometer, running nose is visible, body ache by asking question to patient.

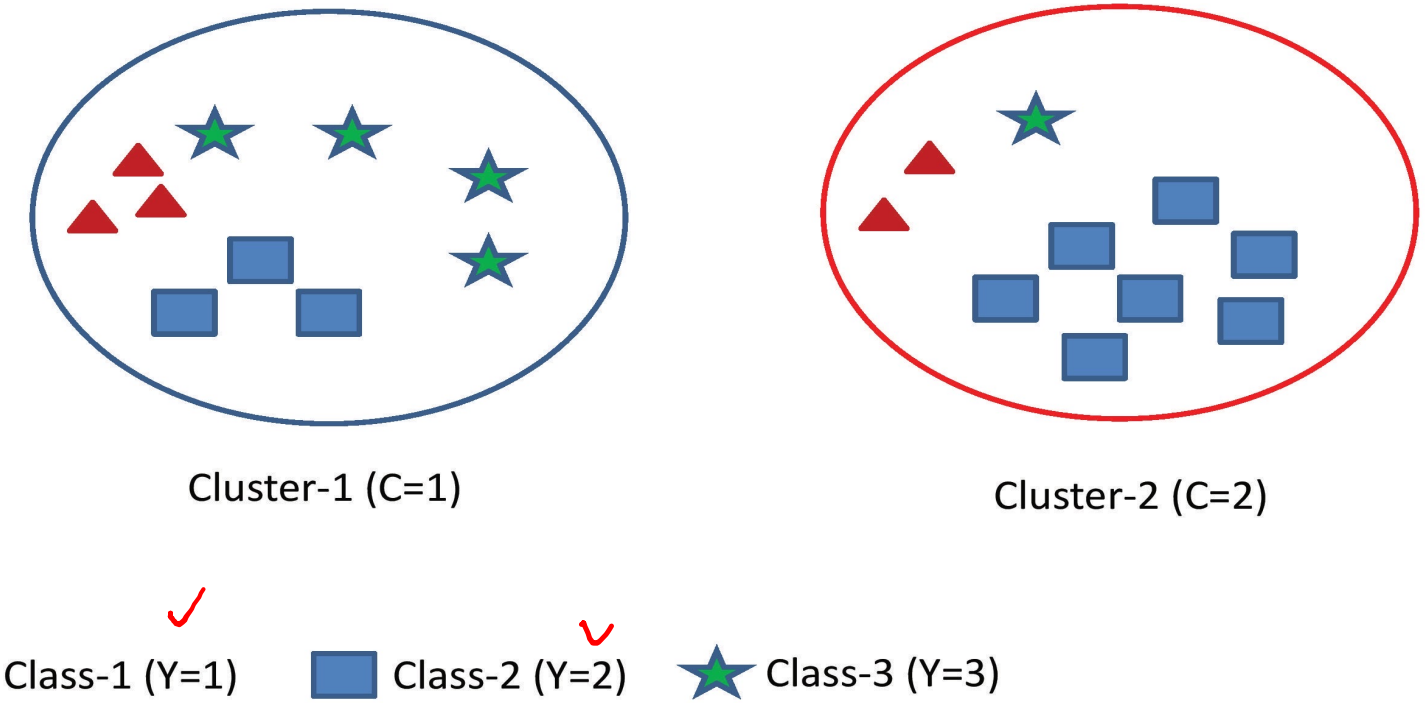
- This test will tell you that there are all common flu symptoms, however, it cannot tell whether covid symptoms are present or not.

- Assuming H(Y|X) is what is compute from the tests, then tests do not say about respiratory issues.

- H(Y) is measured from {X, respiratory issues} and H(Y|X) does not indicate about {respiratory}
- Say you know some covid patients, then see if mutual info is high. If so, you can infer that for a patient having fever, ache issues, test should be done.

Calculating NMI for Clustering

- Assume m=3 classes and k=2 clusters



Task is to compute mutual information between

true labels Y and cluster labels C

$$I(Y;C) = H(Y) - H(Y|C)$$

Higher I would mean clusters are good representative of classes.

H(Y) = Entropy of Class Labels

- P(Y=1) = 5/20 = ¼
- P(Y=2) = 5/20 = ¼
- P(Y=3) = 10/20 = ½
- $H(Y) = -\frac{1}{4}\log\left(\frac{1}{4}\right) - \frac{1}{4}\log\left(\frac{1}{4}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1.5$ bits

This is calculated for the entire dataset and can be calculated prior to clustering, as it will not change depending on the clustering output.

H(C) = Entropy of Cluster Labels

- P(C=1) = 10/20 = 1/2
- P(C=2) = 10/20 = ½
- H(C) $= -\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right) = 1$ bit

This will be calculated every time the clustering changes. You can see from the figure that the clusters are balanced (have equal number of instances).

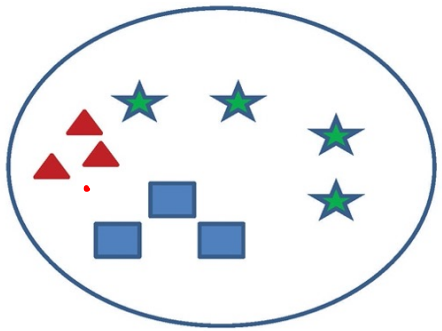
I(Y;C)= Mutual Information

- Mutual information is given as:
  - $I(Y;C) = H(Y) - H(Y|C)$
  - We already know H(Y)
  - H(Y|C) is the entropy of class labels within each cluster, how do we calculate this??

Mutual Information tells us the reduction in the entropy of class labels that we get if we know the cluster labels.

H(Y|C): conditional entropy of class labels for clustering C

- Consider Cluster-1:
  - P(Y=1|C=1)=3/10 (three triangles in cluster-1)
  - P(Y=2|C=1)=3/10 (three rectangles in cluster-1)
  - P(Y=3|C=1)=4/10 (four stars in cluster-1)
  - Calculate conditional entropy as:



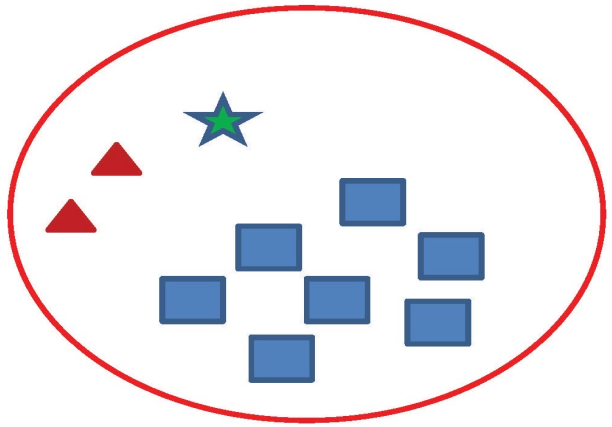
Cluster-1 (C=1)

$$H(Y|C=1) = - \sum_{y \in \{1,2,3\}} P(Y=y|C=1) \log(P(Y=y|C=1))$$

$$= - \left[\frac{3}{10}\log\left(\frac{3}{10}\right) + \frac{3}{10}\log\left(\frac{3}{10}\right) + \frac{4}{10}\log\left(\frac{4}{10}\right)\right]$$

H(Y|C): conditional entropy of class labels for clustering C

- Now, consider Cluster-2:
  - P(Y=1|C=2)=2/10 (two triangles in cluster-1)
  - P(Y=2|C=2)=7/10 (seven rectangles in cluster-1
  - P(Y=3|C=2)=1/10 (one star in cluster-1)
  - Calculate conditional entropy as:



Cluster-2 (C=2)

$$H(Y|C=2) = - \sum_{y\in\{1,2,3\}} P(Y=y|C=2)\log(P(Y=y|C=2))$$

$$= - [\frac{2}{10}\log\left(\frac{2}{10}\right) + \frac{7}{10}\log(\frac{7}{10}) + \frac{1}{10}\log(\frac{1}{10})]$$

$$H(Y|C) = p(C=1)H(Y|C=1) + p(C=2)H(Y|C=2) + \ldots.$$

0.7855

I(Y;C) were high, say, 1.4, this would convey a lot about Y

I(Y;C)

- Finally the mutual information is:

$$I(Y;C) = H(Y) - H(Y|C)$$
$$= 1.5 - [0.7855 + 0.5784]$$
$$= 0.1361$$

$I(Y;C) \in [0, H(Y)]$

The NMI is therefore,

$$NMI(Y,C) = \frac{2 \times I(Y;C)}{[H(Y) + H(C)]}$$

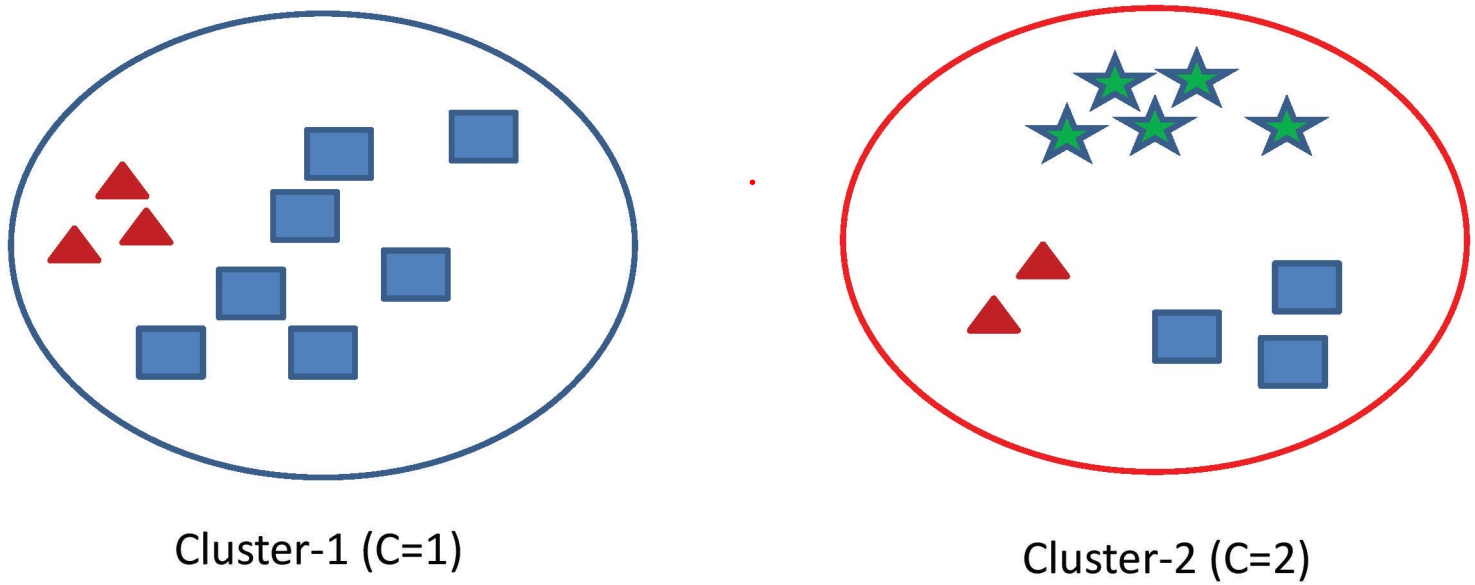
$$NMI(Y,C) = \frac{2 \times 0.1361}{[1.5 + 1]} = 0.1089$$

NMI

- NMI is a good measure for determining the quality of clustering.
- It is an external measure because we need the class labels of the instances to determine the NMI.
- Since it’s normalized we can measure and compare the NMI between different clusterings having different number of clusters.

NMI for Clustering

- Calculate the NMI:



H(Y|C): conditional entropy of class labels for clustering C

- Consider Cluster-1:
  - P(Y=1|C=1)=3/10 (three triangles in cluster-1)
  - P(Y=2|C=1)=7/10 (seven rectangles in cluster-1)
  - P(Y=3|C=1)=0/10 (no stars in cluster-1)
  - Calculate conditional entropy as:

$$H(Y|C=1) = -P(C=1)\sum_{y\in\{1,2,3\}} P(Y=y|C=1)\log(P(Y=y|C=1))$$

$$= -\frac{1}{2}\times\left[\frac{3}{10}\log\left(\frac{3}{10}\right)+\frac{0}{10}\log\left(\frac{0}{10}\right)+\frac{7}{10}\log\left(\frac{7}{10}\right)\right] = 0.4406$$

We used 0log(0)=0

H(Y|C): conditional entropy of class labels for clustering C

- Now, consider Cluster-2:
  - P(Y=1|C=2)=2/10 (two triangles in cluster-1)
  - P(Y=2|C=2)=3/10 (three rectangles in cluster-1)
  - P(Y=3|C=2)=5/10 (five stars in cluster-1)
  - Calculate conditional entropy as:

$$H(Y|C=2) = -P(C=2) \sum_{y\in\{1,2,3\}} P(Y=y|C=2)\log(P(Y=y|C=2))$$

$$= -\frac{1}{2} \times \left[\frac{2}{10}\log\left(\frac{2}{10}\right) + \frac{3}{10}\log\left(\frac{3}{10}\right) + \frac{5}{10}\log\left(\frac{5}{10}\right)\right] = 0.7427$$

I(Y;C)

- Finally the mutual information is:

$$I(Y;C) = H(Y) - H(Y|C)$$
$$= 1.5 - [0.4406 + 0.7427]$$
$$= 0.3167$$

The NMI is therefore,

$$NMI(Y,C) = \frac{2 \times I(Y;C)}{[H(Y) + H(C)]}$$

$$NMI(Y,C) = \frac{2 \times 0.3167}{[1.5 + 1]} = 0.2533$$

Normalized Mutual Information

- Normalized Mutual Information:

$$NMI(Y, C) = \frac{2 \times I(Y; C)}{[H(Y) + H(C)]}$$

where,

1) Y = class labels
2) C = cluster labels
3) H(.) = Entropy
4) I(Y;C) = Mutual Information b/w Y and C

Note: All logs are base-2.

Comments

- NMI for the second clustering is higher than the first clustering. It means we would prefer the second clustering over the first.
  - You can see that one of the clusters in the second case contains all instances of class-3 (stars).
- If we have to compare two clustering that have different number of clusters we can still use NMI.

Cluster evaluation (a hard problem)

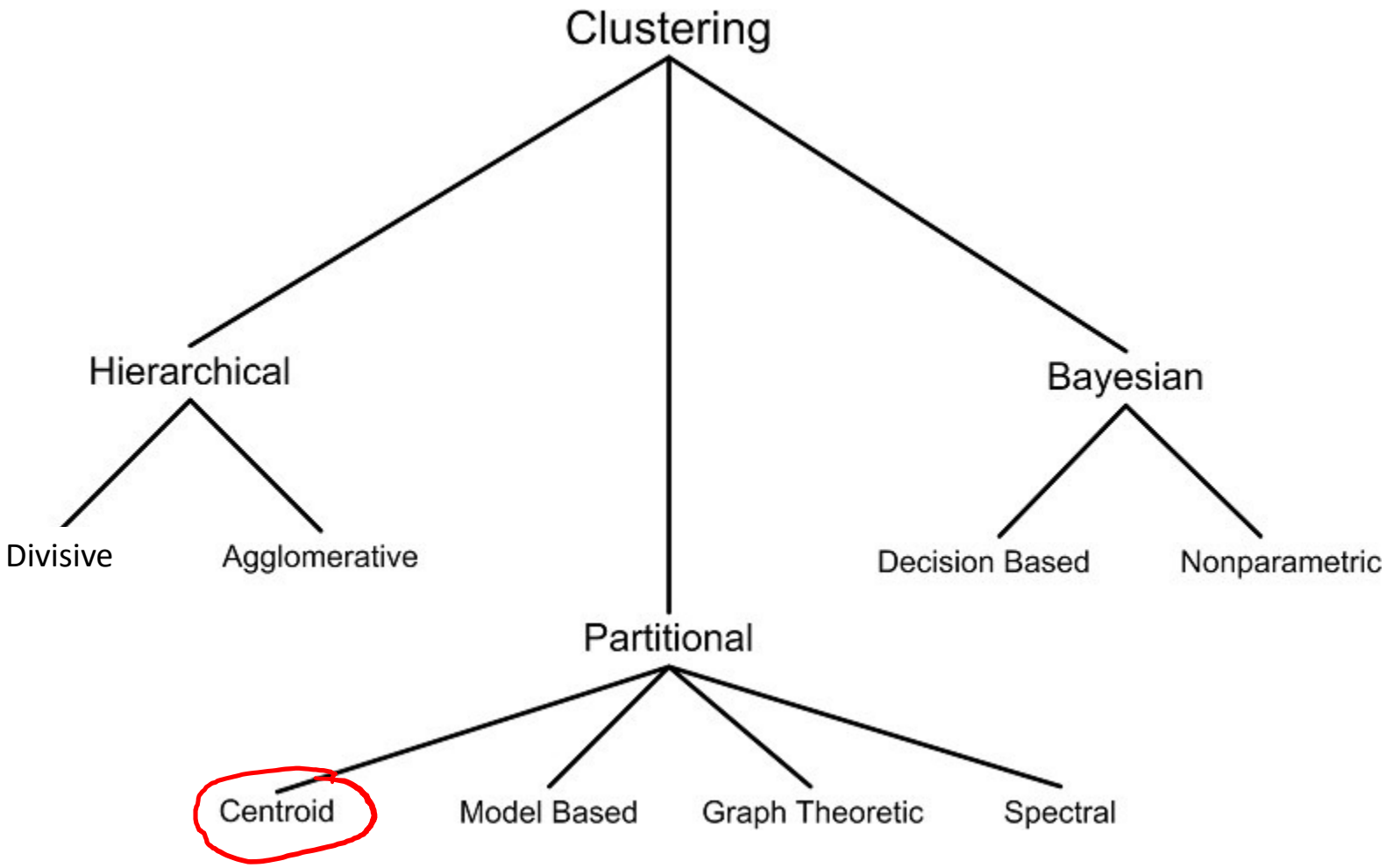
- **Intra-cluster cohesion** (compactness):
  - Cohesion measures how near the data points in a cluster are to the cluster centroid.
  - Sum of squared error (SSE) is a commonly used measure.
- **Inter-cluster separation** (isolation):
  - Separation means that different cluster centroids should be far away from one another.
- In most applications, expert judgments are still the key

How many clusters?

3 clusters or 2 clusters?

- Possible approaches
  1. fix the number of clusters to ***k***
  2. find the best clustering according to the criterion function (number of clusters may vary)

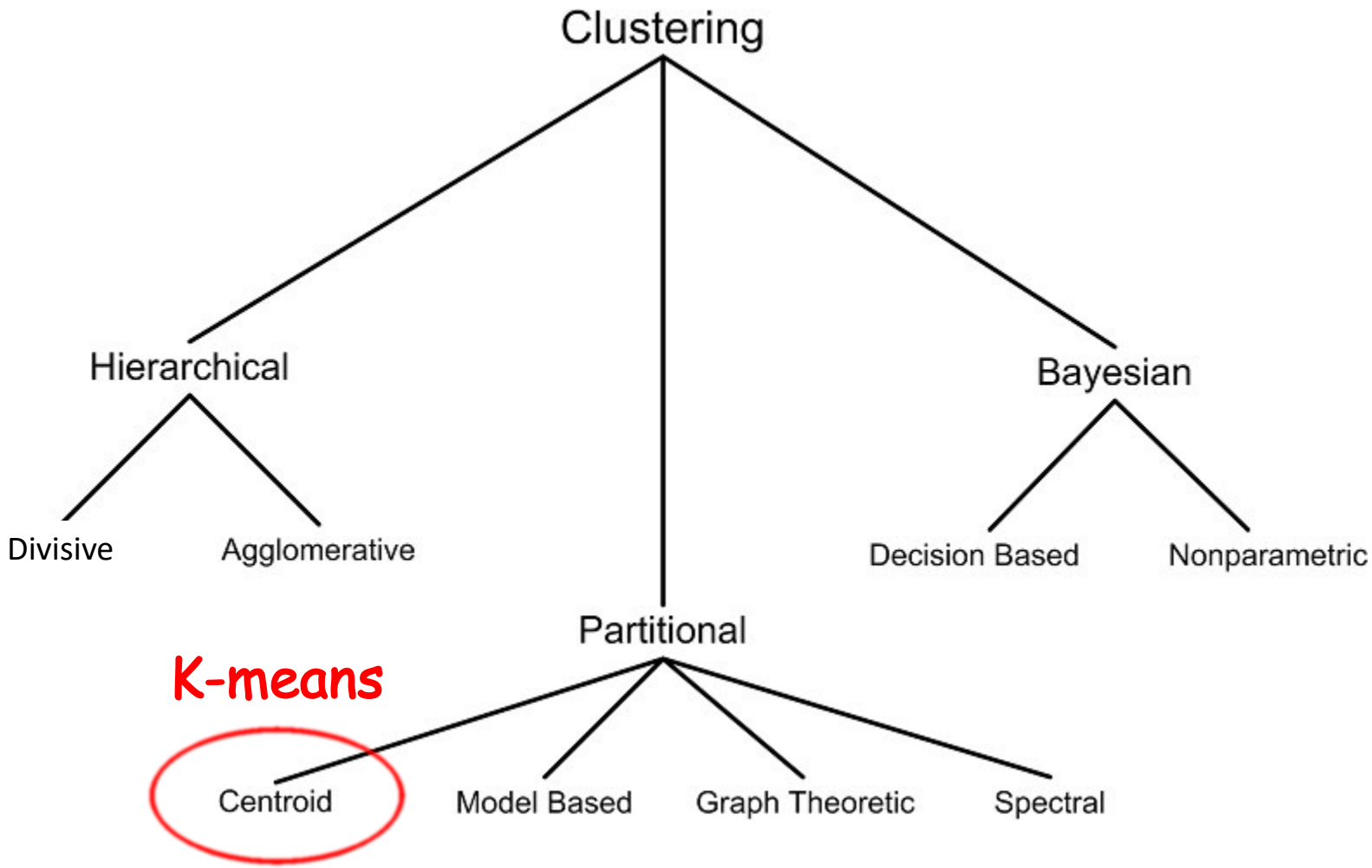
Clustering techniques



Clustering techniques

- **Hierarchical** algorithms find successive clusters using previously established clusters. These algorithms can be either agglomerative ("*bottom-up*") or divisive ("*top-down*"):
  1. Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters;
  2. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.
- **Partitional** algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering.
- **Bayesian** algorithms try to generate a *posteriori distribution* over the collection of all partitions of the data.

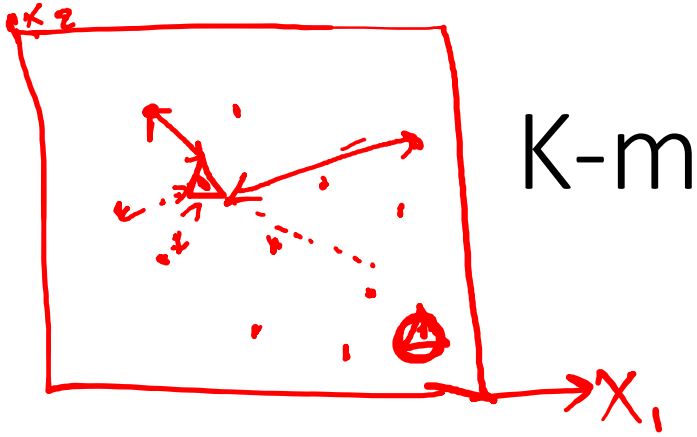
Clustering techniques



K-Means clustering

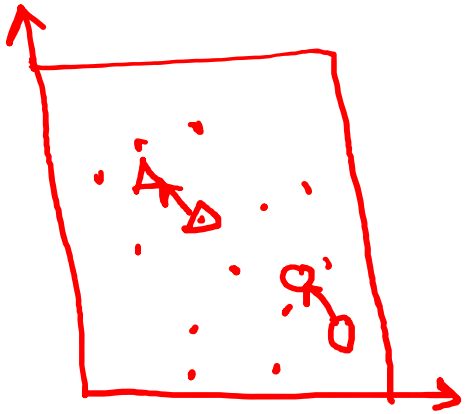
- K-means (MacQueen, 1967) is a partitional clustering algorithm
- Let the set of data points D be $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$,

  where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector in $X \subseteq R^r$, and r is the number of dimensions.
- The *k*-means algorithm partitions the given data into *k* clusters:
  - Each cluster has a cluster **center**, called **centroid**.
  - *k* is specified by the user



K-means algorithm

K = 2 clusters.



- Given *k*, the *k-means* algorithm works as follows:
  1. Choose *k* (random) data points (seeds) to be the initial centroids, cluster centers
  2. Assign each data point to the closest centroid
  3. Re-compute the centroids using the current cluster memberships
  4. If a convergence criterion is not met, repeat steps 2 and 3

K-means convergence (stopping) criterion

- no (or minimum) re-assignments of data points to different clusters, *or*
- no (or minimum) change of centroids, *or*
- minimum decrease in the **sum of squared error** (SSE),

$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} d(\mathbf{x}, \mathbf{m}_j)^2$$

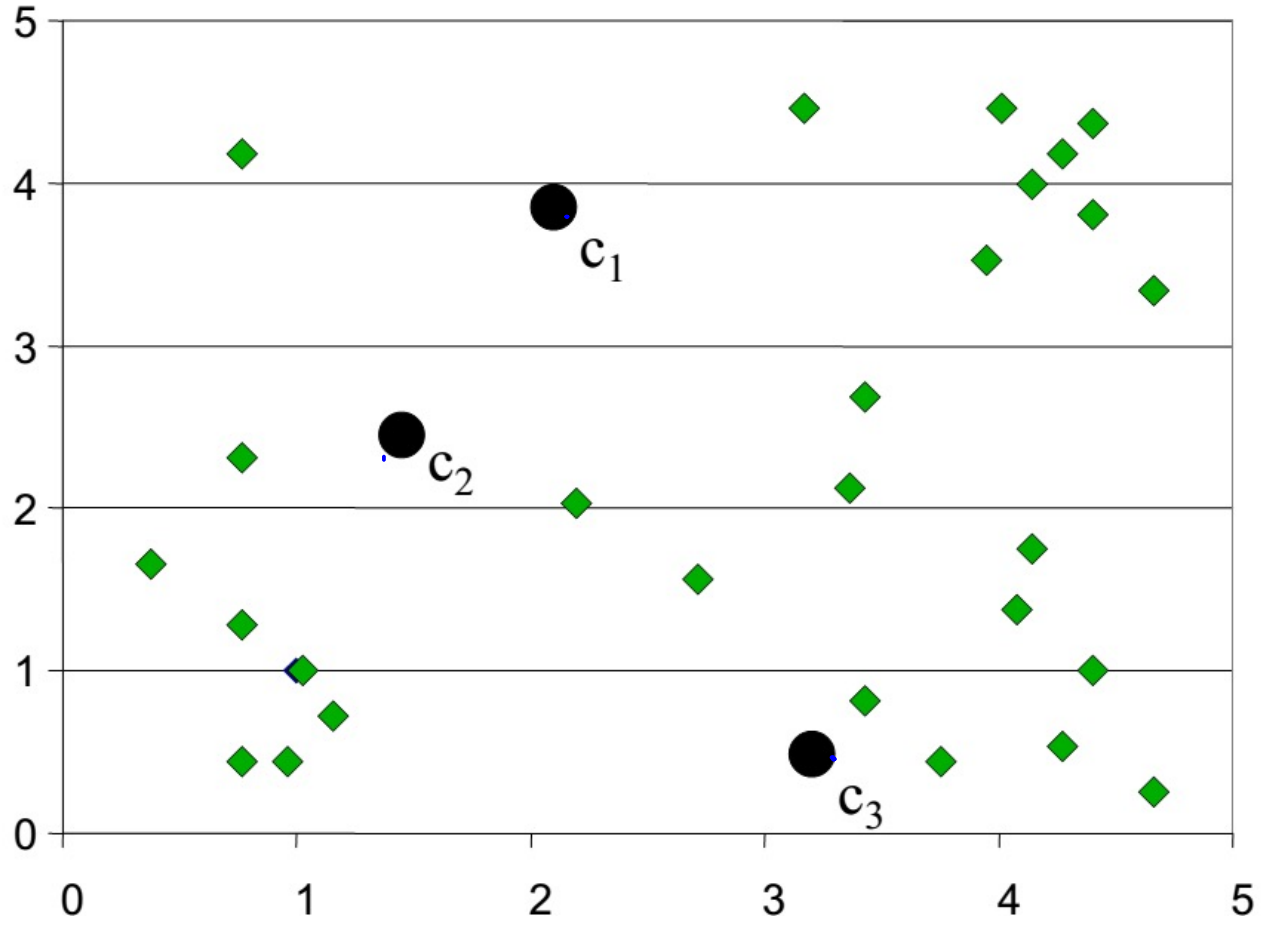
  - C_j is the jth cluster,
  - $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j),
  - $d(\mathbf{x}, \mathbf{m}_j)$ is the (Eucledian) distance between data point $\mathbf{x}$ and centroid $\mathbf{m}_j$.

K-means algorithm

- Given *k*, the *k-means* algorithm works as follows:
  1. Choose *k* (random) data points (seeds) to be the initial centroids, cluster centers

K-means clustering example: step 1

Randomly initialize the cluster centers



	x_1	x_2	...	x_n
c_1	$d(x_1, c_1)$	$d(x_2, c_1)$		$d(x_n, c_1)$
c_2	$d(x_1, c_2)$			
c_3	$d(x_1, c_3)$			$d(x_n, c_3)$

Compute $k \times n$ matrix distance.

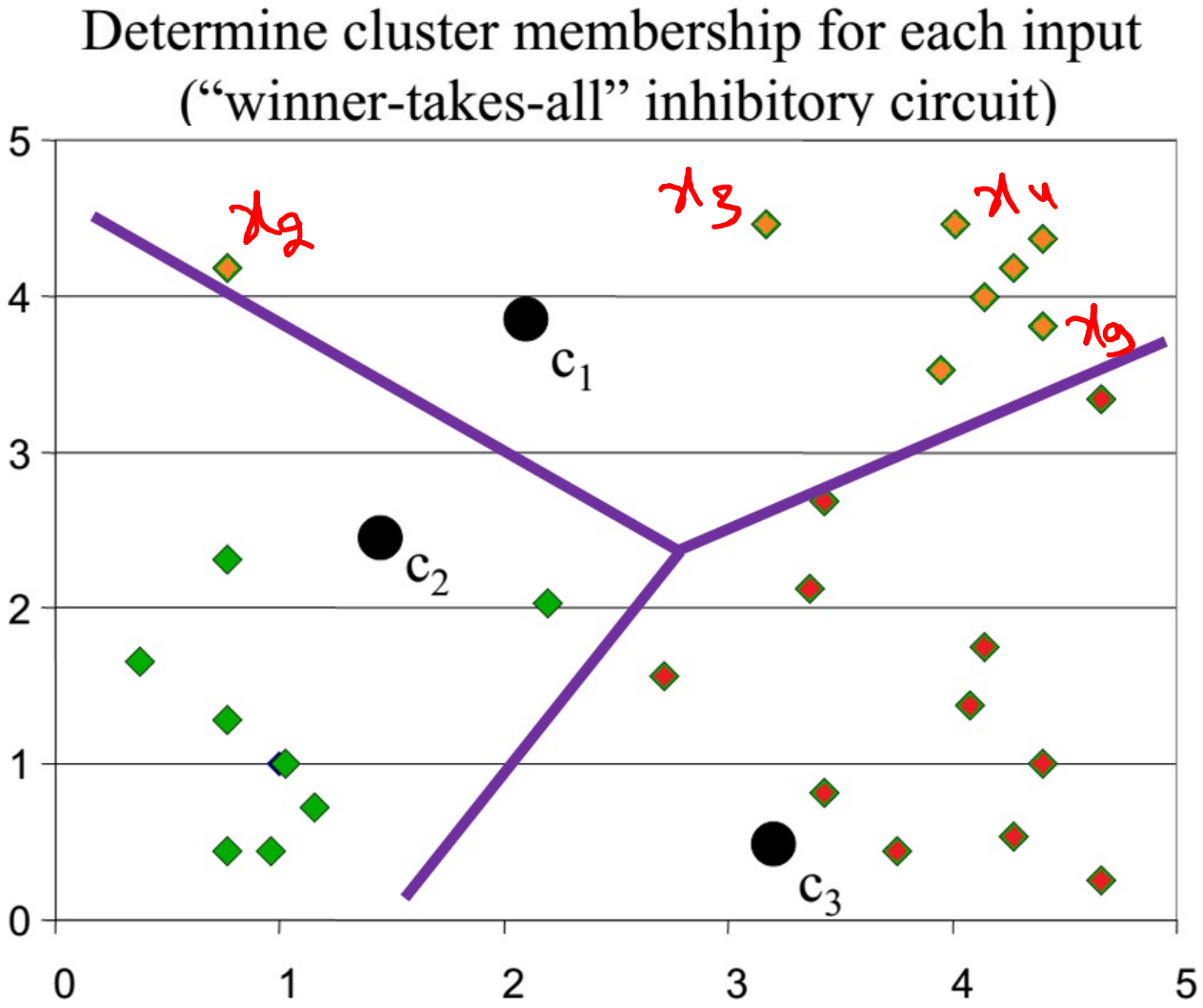
for x_1, $\min\{d(x_1, c_1), d(x_1, c_2), d(x_1, c_3)\}$

$= d(x_1, c_2) \rightarrow$ assign x_1 to c_2

K-means algorithm

- Given *k*, the *k-means* algorithm works as follows:
  1. Assign each data point to the closest centroid

K-means clustering example – step 2



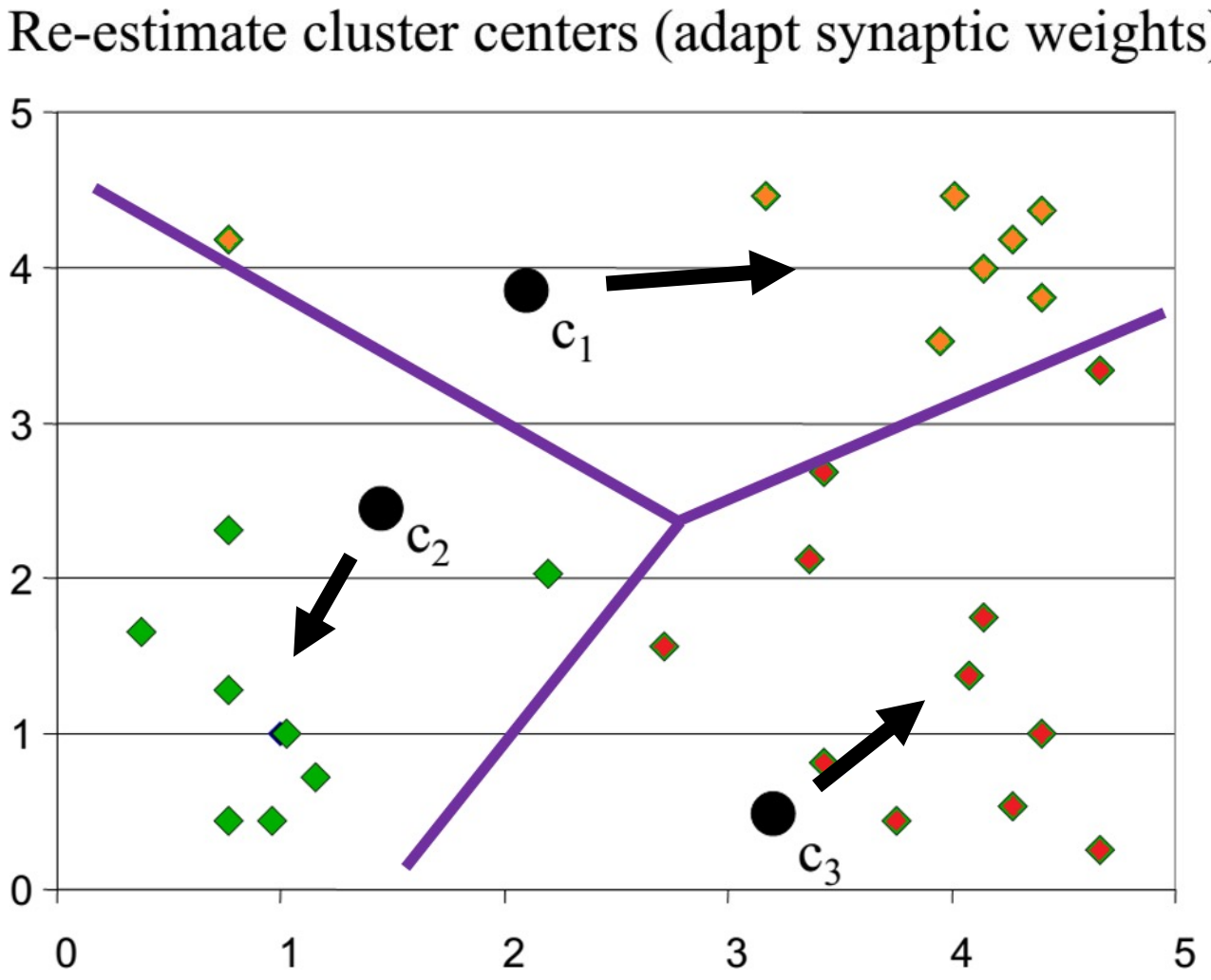
New centre → $C_1^{new} = \frac{1}{8} \sum_{i=2}^{8} x_i$

assuming 8 points are assigned to C_1

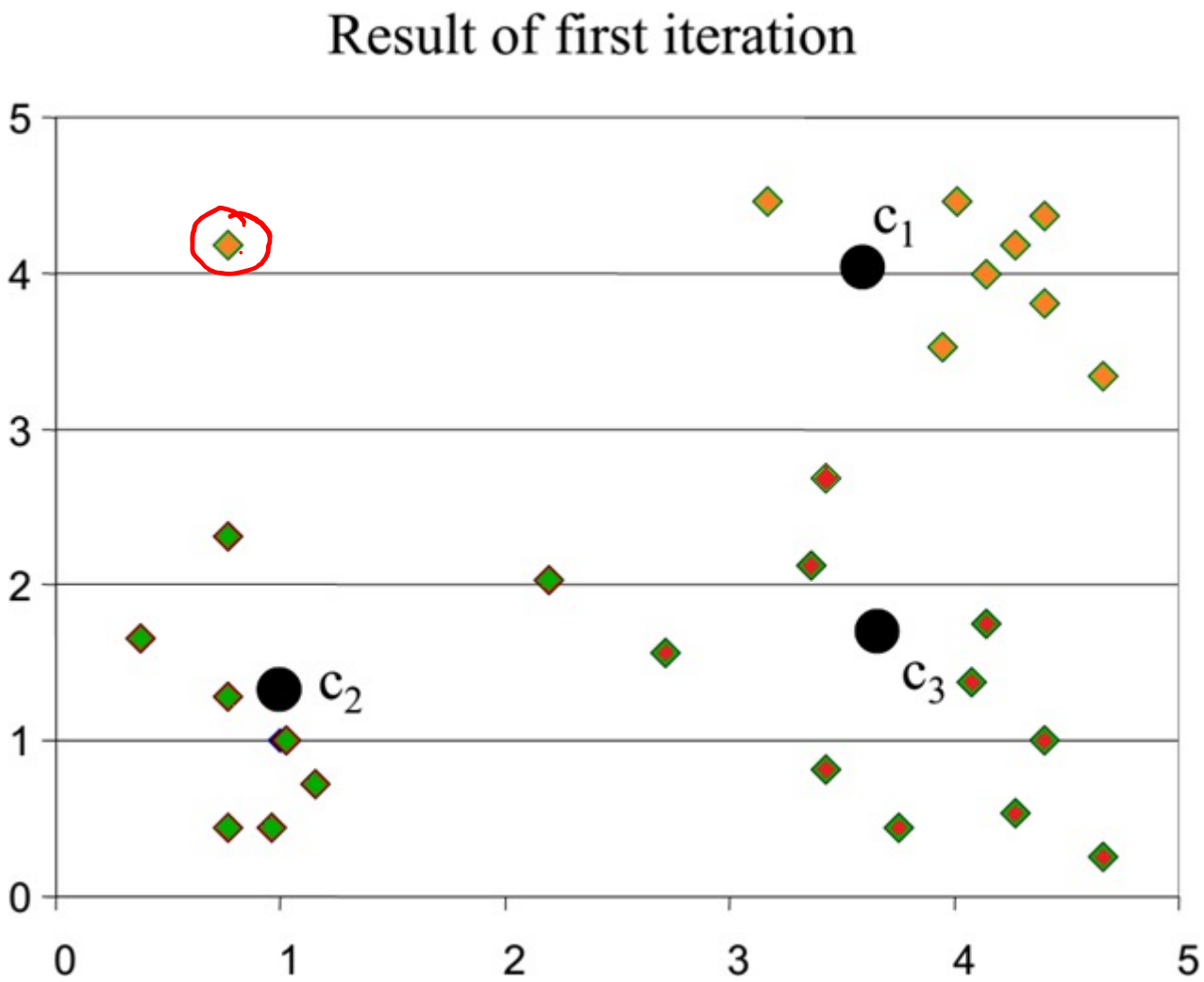
K-means algorithm

- Given *k*, the *k-means* algorithm works as follows:
  1. Re-compute the centroids using the current cluster memberships
  2. If a convergence criterion is not met, repeat steps 2 and 3

K-means clustering example – step 3



K-means clustering example

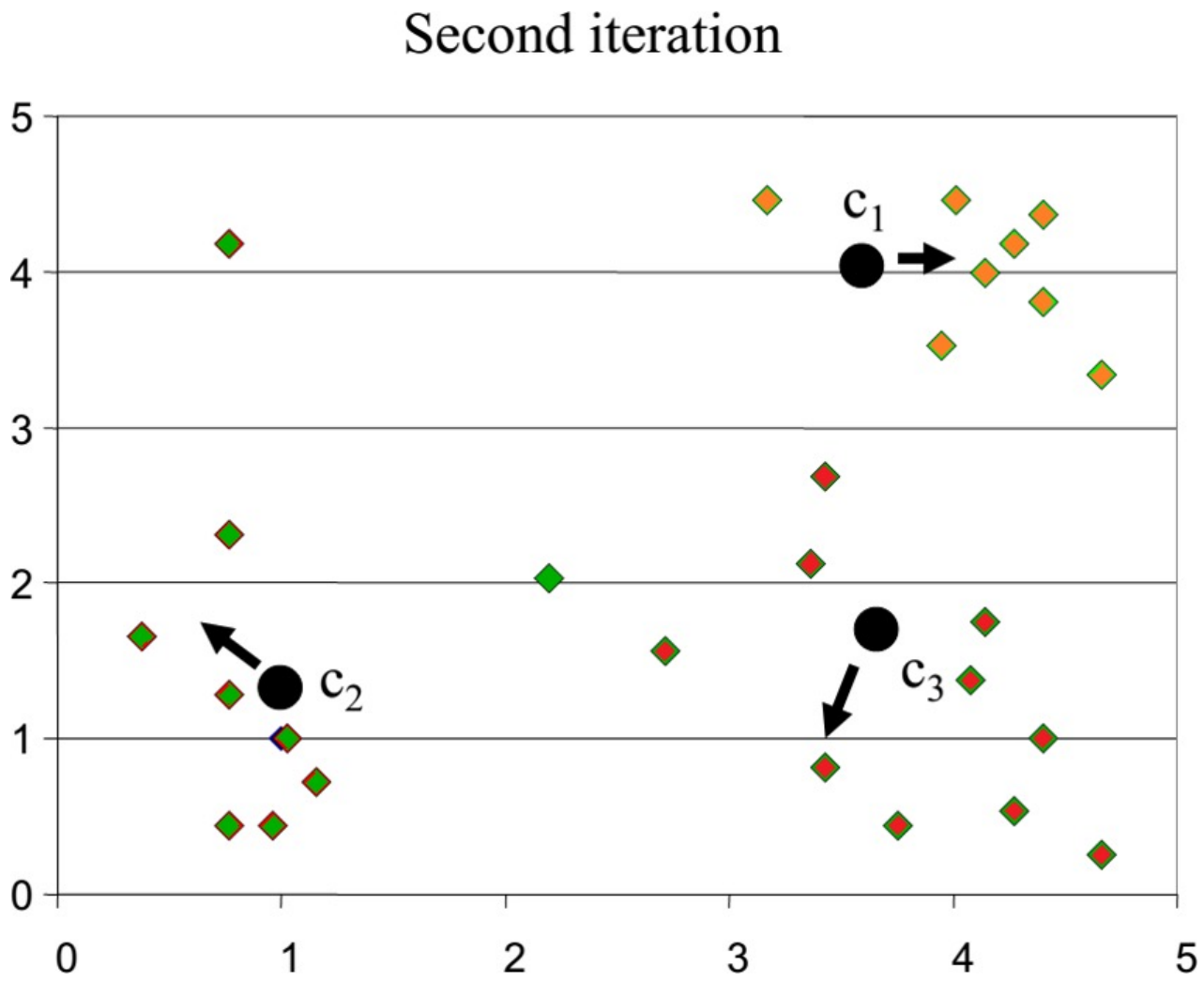


K-means algorithm

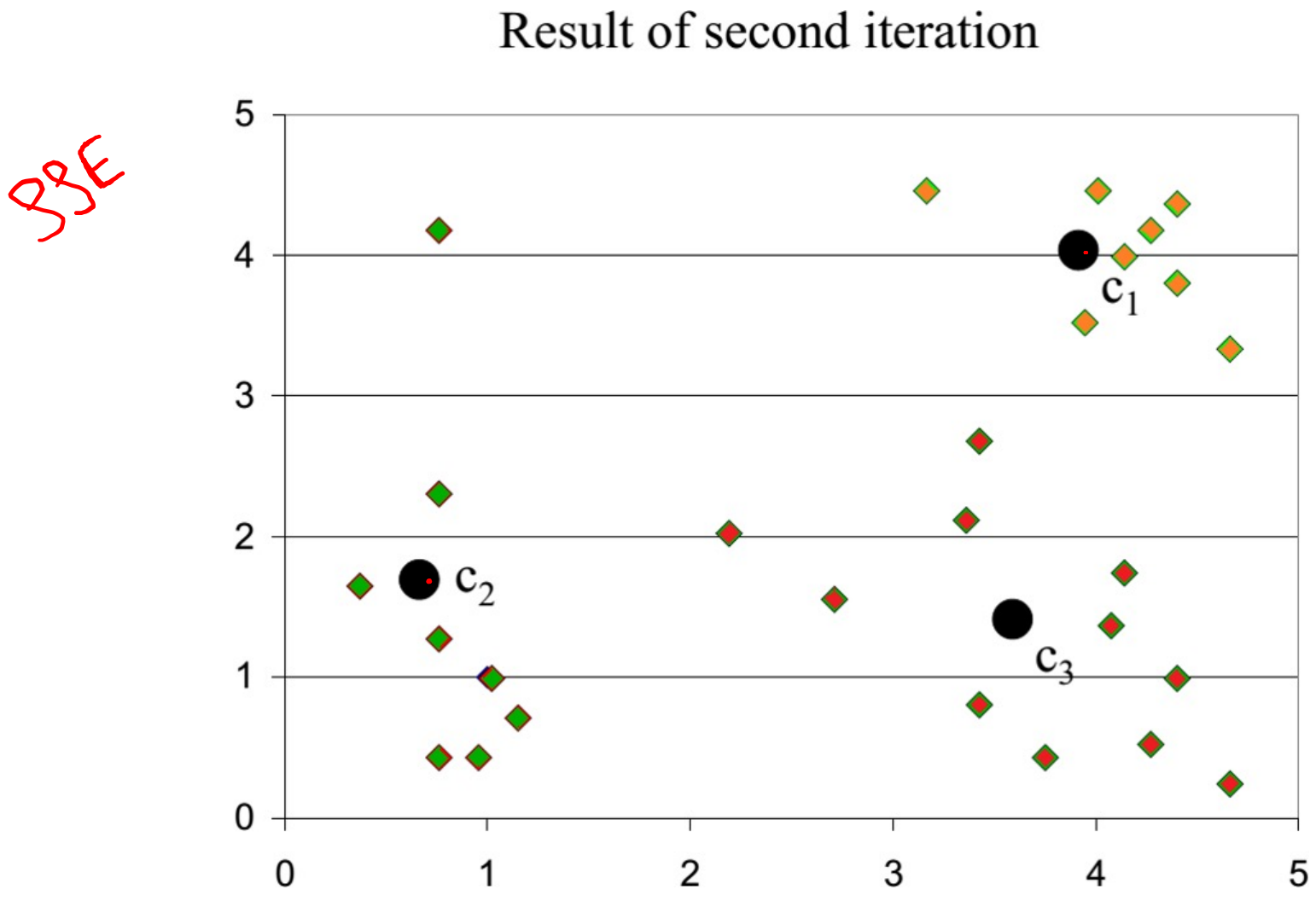
- Given *k*, the *k-means* algorithm works as follows:

  If a convergence criterion is not met, repeat steps 2 and 3

K-means clustering example



K-means clustering example



At iteration 1 → SSE → 10

iter. 2 → SSE → 5 → $|10-5| <$ thres.

3 → SSE → 4 → $|5-4| <$ thres.

4 → SSE → 3.9 → $|4-3.9| <$ thres.

5 → SSE → 3.85

thres = ~~0.2~~

= 0.02

Convergence. $|SSE_t - SSE_{t-1}| <$ threshold

exit.

Why use K-means?

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient: Time complexity: $O(tkn)$,

    where n is the number of data points,

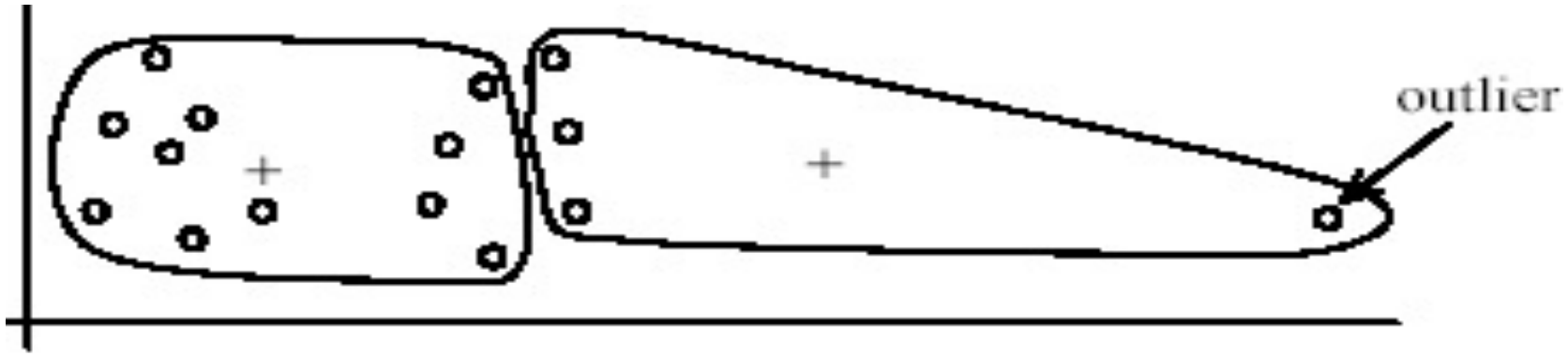
    k is the number of clusters, and

    t is the number of iterations.
  - Since both k and t are small. k-means is considered a linear algorithm.
- K-means is the most popular clustering algorithm.
- Note that: it terminates at a local optimum if SSE is used. The global optimum is hard to find due to complexity.

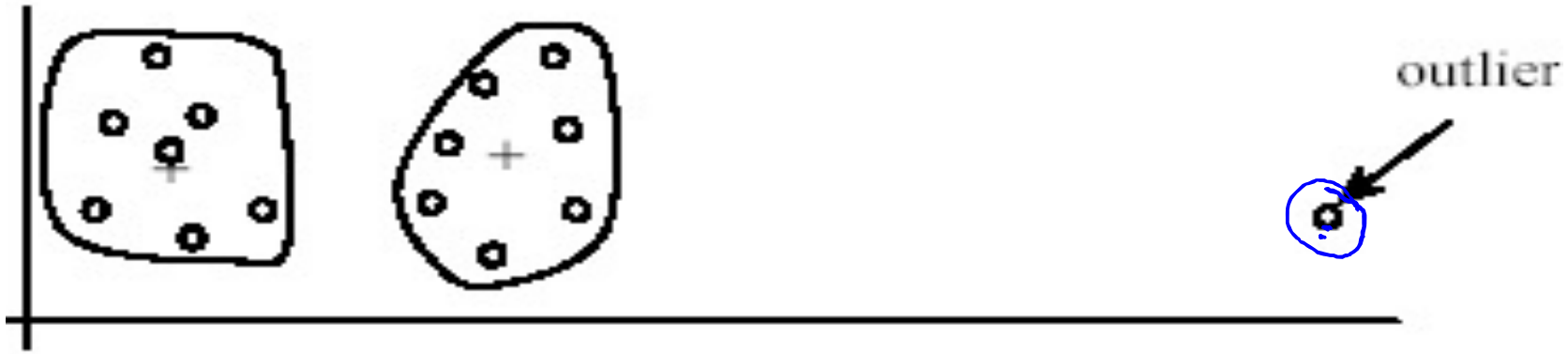
Weaknesses of K-means

- The user needs to specify *k*.
- The algorithm is sensitive to **outliers**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Outliers



(A): Undesirable clusters



(B): Ideal clusters

Dealing with outliers

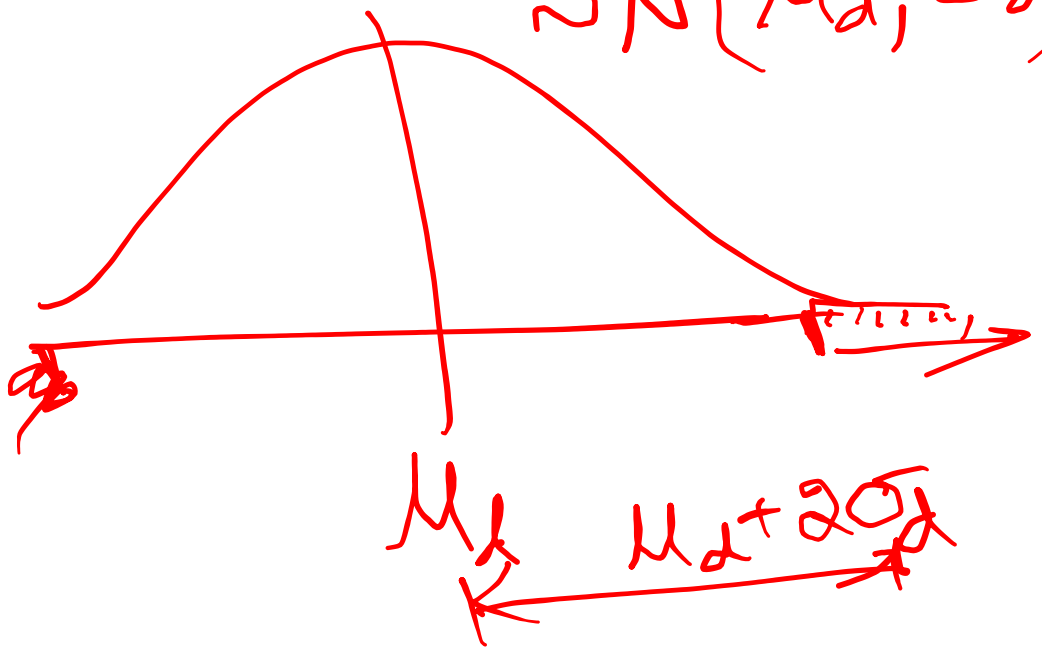
- Remove some data points that are much further away from the centroids than other data points
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- Perform random sampling: by choosing a small subset of the data points, the chance of selecting an outlier is much smaller
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

Outlier detection

distances, $d_1, d_2, d_3 \cdots d_n$

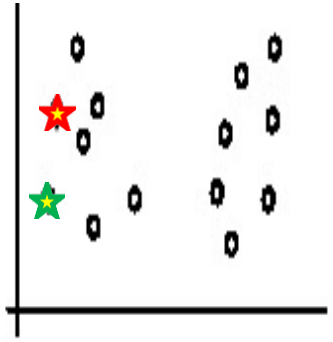
$\mu_d = \frac{1}{n}\sum_{i=1}^{n} d_i$, $\sigma_d^2 =$

$\sim N(\mu_d, \sigma_d^2)$

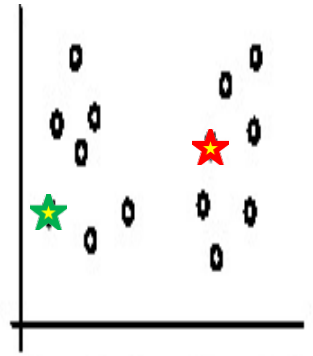


if $d_i > \mu + 2\sigma$
→ outlier

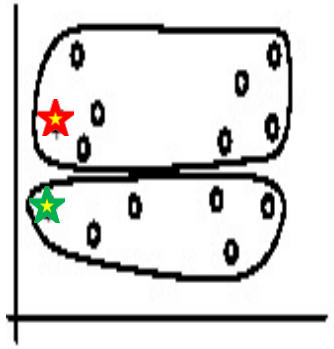
Sensitivity to initial seeds



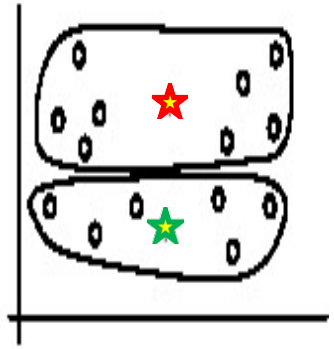
Random selection of seeds (centroids)



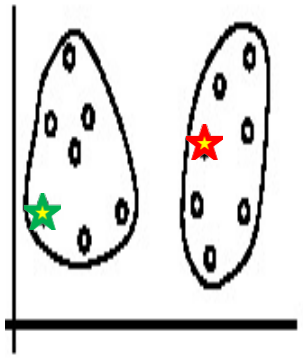
Random selection of seeds (centroids)



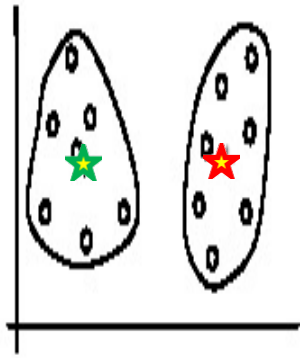
Iteration 1



Iteration 2



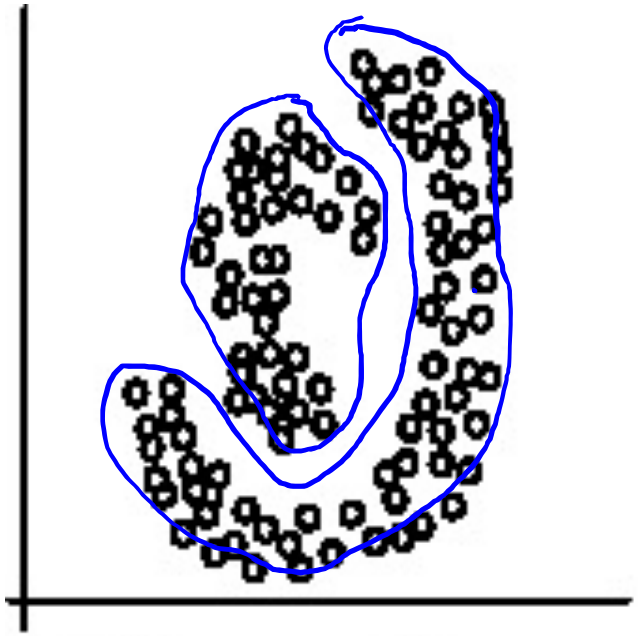
Iteration 1



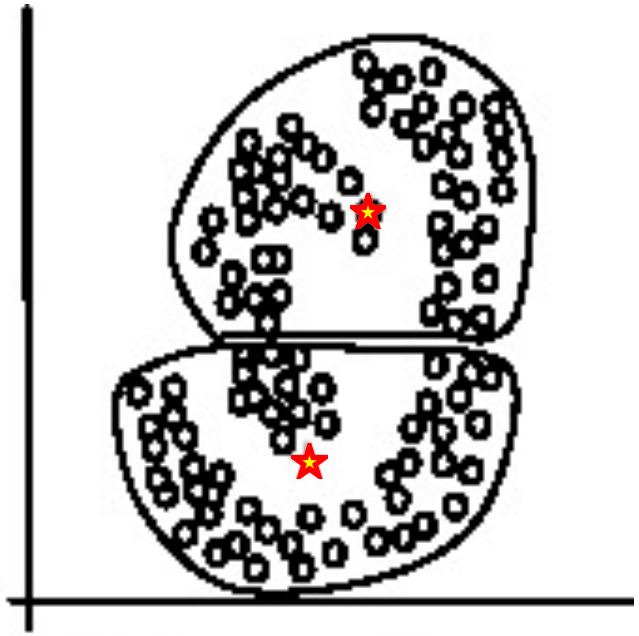
Iteration 2

Special data structures

- The *k*-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).



(A): Two natural clusters



(B): *k*-means clusters

K-means summary

- Despite weaknesses, *k*-means is still the most popular algorithm due to its simplicity and efficiency
- No clear evidence that any other clustering algorithm performs better in general
- Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!