# Chapter 4 Machine Learning

## COMP 6721 Introduction of AI

*Russell & Norvig – Section 18.1 & 18.2*

# Topics

- Evaluation of Learning Approach
- Unsupervised Learning

# Evaluation of Learning Approach

- how do you know if what you learned is correct?
- You run your classifier on a data set of unseen examples (that you did not use for training) for which you know the correct classification
- Split data set into 3 sub-sets

  **Training** set (~80%)

  - **Actual training set (~80%)**
  - **Validation set (~20%)**
  - **Test** set (~20%)

# Evaluation Methodology

- Standard methodology:
  1. Collect a large set of examples (all with correct classifications)
  2. Divide collection into training, validation and test sets

  Loop:

  3. Apply learning algorithm to training set
  4. Measure performance with the validation set, and adjust hyper-parameters* to improve performance
  5. Measure performance with the test set
- DO NOT LOOK AT THE TEST SET until step 5.
- Hyper-parameters: parameters used to set up your ML algorithm. eg.
  - for NB: value of delta for smoothing,
  - for DTs, pruning level.

# Metrics

- Accuracy
  - % of instances of the test set the algorithm correctly classifies
  - when all classes are equally important and represented
- Recall, Precision & F-measure
  - when one class is more important and the others

# Accuracy

- % of instances of the test set the algorithm correctly classifies
- problem:
  - when one class C is more important and the others
  - eg. when data set is unbalanced

| | *Target* | *system 1* |
|---|---|---|
| | X1 √ | X1 × |
| | X2 √ | X2 × |
| | X3 √ | X3 × |
| | X4 √ | X4 × |
| | X5 √ | X5 × |
| | X6 × | X6 × |
| | X7 × | X7 × |
| | … | … |
| | … | … |
| | X500 × | X500 × |
| *Accuracy* | | 495/500 = 99% ! |

# Recall, Precision

- ***Recall:*** What proportion of the instances in class C are labelled correctly?
- ***Precision:*** What proportion of instances labeled with the class C are actually correct?

| Model says… | In reality, the instance is… in class C | Is not in class C |
|---|---|---|
| instance is in class C | A | B |
| instance is NOT in class C | C | D |

$$\text{Precision} = \frac{A}{A+B}$$

*instances that are in class C and that the model identified as class C*

*instances that the model labelled as class C*

$$\text{Recall} = \frac{A}{A+C}$$

*instances that are in class C and that the model identified as class C*

*All instances that are in class C*

# Example

| | *Target* | *system 1* | *system 2* | *system 3* |
|---|---|---|---|---|
| | X1 √ | X1 × | X1 √ | X1 √ |
| | X2 √ | X2 × | X2 × | X2 √ |
| | X3 √ | X3 × | X3 √ | X3 √ |
| | X4 √ | X4 × | X4 √ | X4 √ |
| | X5 √ | X5 × | X5 × | X5 √ |
| | X6 × | X6 × | X6 × | X6 √ |
| | X7 × | X7 × | X7 × | X7 √ |
| | … × | … | … × | … × |
| | … × | … | … × | … × |
| | X500 × | X500 × | X500 × | X500 × |
| *Accuracy* | | 495/500 = 99% ! | 498/500 = 99.6% | 498/500 = 99.6% |
| *Precision* | | 0/0 | 3/3 = 100% | 5/7 = 71% |
| *Recall* | | 0/5 = 0% | 3/5 = 60% | 5/5 = 100% |

***Which system is better?***

# Evaluation: A Single Value Measure

- cannot take mean of Precision & Recall
  - if R = 50%  P = 50%  M = 50%
  - if R = 100%  P = 10%  M = 55% (not fair)
- take harmonic mean (HM)

$$HM = \frac{2}{\frac{1}{R} + \frac{1}{P}}$$

HM is high only when both P&R are high

if R = 50% and P = 50%  HM = 50%

if R = 100% and P = 10%  HM = 18.2%

# Evaluation: A Single Value Measure

- take weighted harmonic mean (WHM)

  $w_r$: weight of R  $w_p$: weight of P  $a = 1/w_r$  $b = 1/w_p$

$$WHM = \frac{a+b}{\left(\frac{a}{R}+\frac{b}{P}\right)} = \frac{\frac{(a+b)}{b}}{\left(\frac{a}{Rb}+\frac{b}{Pb}\right)} = \frac{\frac{a}{b}+1}{\left(\frac{a}{bR}+\frac{1}{P}\right)}$$

  - let $\beta^2 = a/b$

$$WHM = \frac{\beta^2+1}{\left(\frac{\beta^2}{R}+\frac{1}{P}\right)} = \frac{(\beta^2+1)PR}{\beta^2 P+R}$$

… which is called the ***F-measure***

# Evaluation: the F-measure

- A weighted combination of precision and recall

$$F = \frac{(\beta^2 + 1)PR}{(\beta^2 P + R)}$$

- $\beta$ represents the relative importance of precision and recall
  - when $\beta = 1$, precision & recall have same importance
  - when $\beta > 1$, precision is favored
  - when $\beta < 1$, recall is favored

# Example

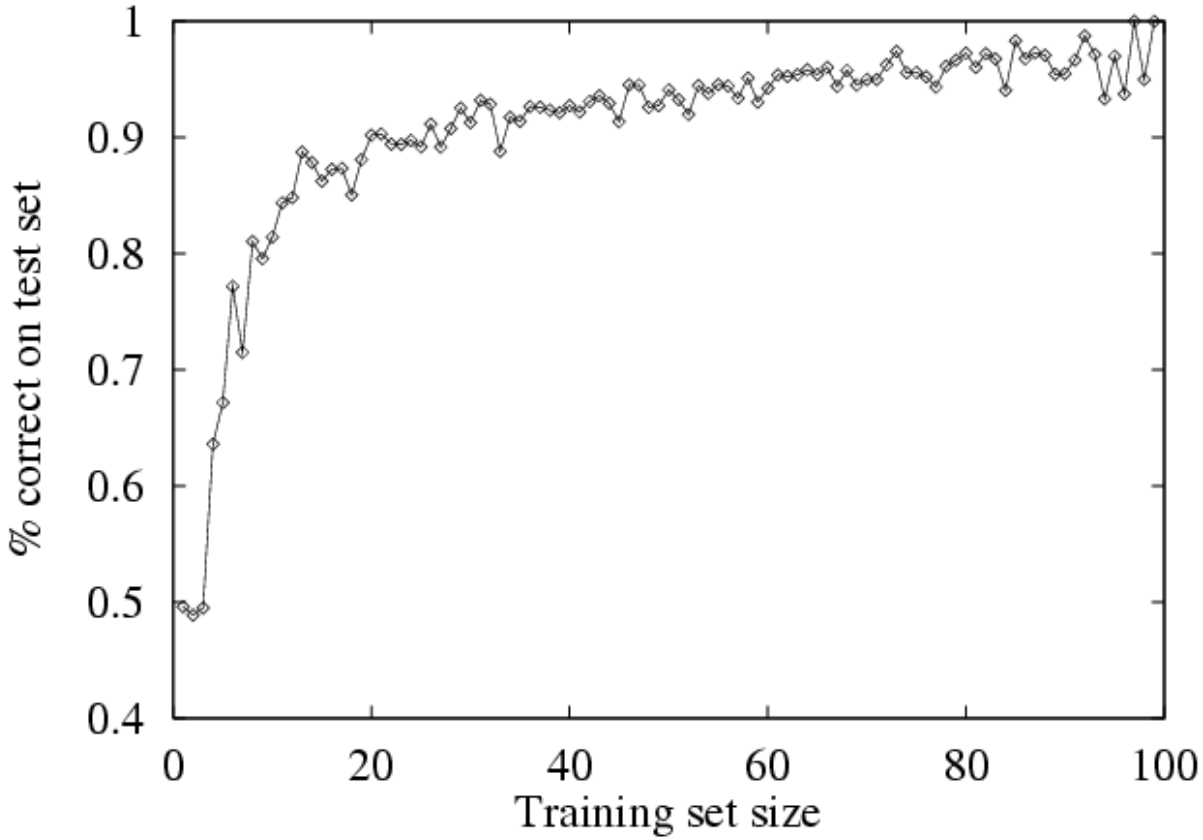
| | *Target* | *system 1* | *system 2* | *system 3* |
|---|---|---|---|---|
| | X1 √ | X1 × | X1 √ | X1 √ |
| | X2 √ | X2 × | X2 × | X2 √ |
| | X3 √ | X3 × | X3 √ | X3 √ |
| | X4 √ | X4 × | X4 √ | X4 √ |
| | X5 √ | X5 × | X5 × | X5 √ |
| | X6 × | X6 × | X6 × | X6 √ |
| | X7 × | X7 × | X7 × | X7 √ |
| | … × | … | … × | … × |
| | … × | … | … × | … × |
| | X500 × | X500 × | X500 × | X500 × |
| *Accuracy* | | 495/500 = 99% ! | 498/500 = 99.6% | 498/500 = 99.6% |
| *Precision* | | 0/0 | 3/3 = 100% | 5/7 = 71% |
| *Recall* | | 0/5 = 0% | 3/5 = 60% | 5/5 = 100% |
| *F1 -measure (B=1)* | | Undef | 2PR/P+R = 75% | 2PR/P+R = 83% |

# Error Analysis

- Where did the learner go wrong ?
- Use a confusion matrix / contingency table

| correct class (that should have been assigned) | classes assigned by the learner | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **C1** | **C2** | **C3** | **C4** | **C5** | **C6** | … | **Total** |
| **C1** | 94 | 3 | 0 | 0 | 3 | 0 | | 100 |
| **C2** | 0 | 93 | 3 | 4 | 0 | 0 | | 100 |
| **C3** | 0 | 1 | 94 | 2 | 1 | 2 | | 100 |
| **C4** | 0 | 1 | 3 | 94 | 2 | 0 | | 100 |
| **C5** | 0 | 0 | 3 | 2 | 92 | 3 | | 100 |
| **C6** | 0 | 0 | 5 | 0 | 10 | 85 | | 100 |
| … | | | | | | | | |

# A Learning Curve




- Size of training set
  - the more, the better
  - but after a while, not much improvement…

# Some words on Training

- In all types of learning… watch out for:
  - Noisy input
  - Overfitting/underfitting the training data

# Noisy Input

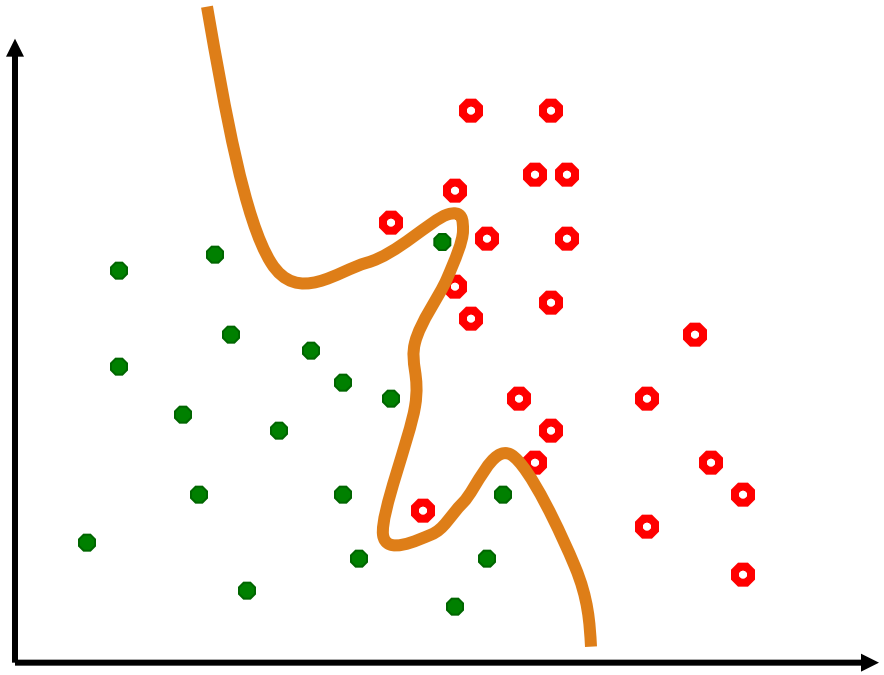
- Noisy Input:
  - Two examples have the same feature-value pairs, but different outputs

| Size | Color | Shape | Output |
|---|---|---|---|
| Big | Red | Circle | + |
| Big | Red | Circle | - |

  - Some values of features are incorrect or missing (ex. errors in the data acquisition)
  - Some relevant attributes are not taken into account in the data set

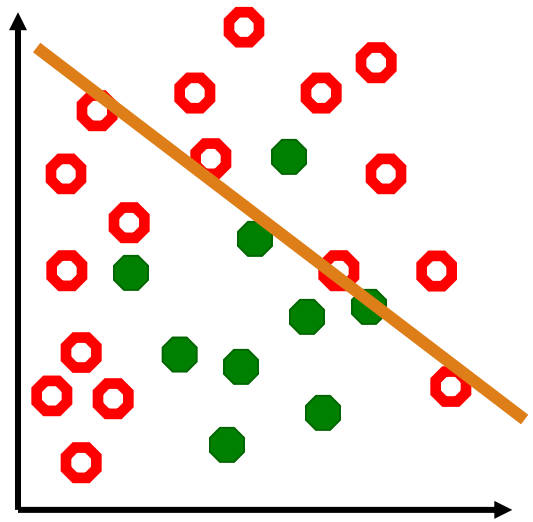
# Overfitting

- If a large number of irrelevant features are there, we may find meaningless regularities in the data that are particular to the training data but irrelevant to the problem.
- Complicated boundaries overfit the data
- they are too tuned to the particular training data at hand
- They do not generalize well to the new data
- Extreme case: "rote learning"
- Training error is low
- Testing error is high



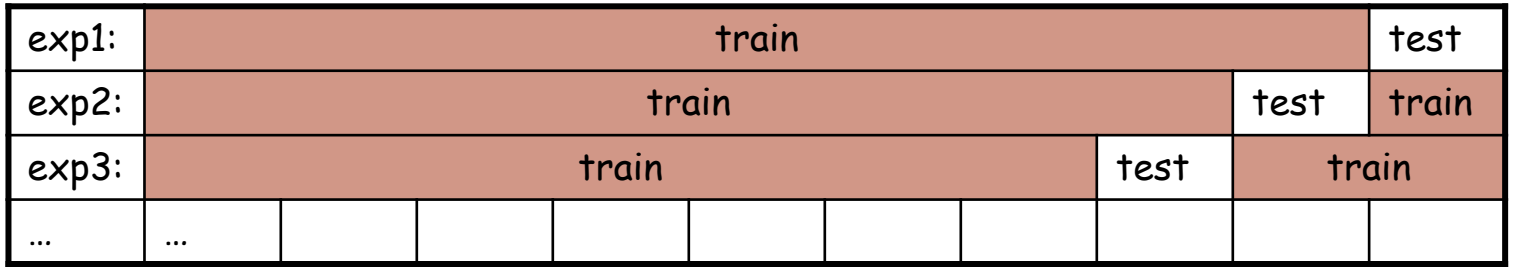
# Underfitting

- We can also underfit data, i.e. use too simple decision boundary
- Model is not expressive enough (not enough features)
- There is no way to fit a linear decision boundary so that the training examples are well separated
- Training error is high
- Testing error is high



# Cross - validation

- K-fold cross-validation
  - run k experiments, each time you test on 1/k of the data, and train on the rest
  - get the average of results
- ex: 10-fold cross validation
  1. Collect a large set of examples (all with correct classifications)
  2. Divide collection into two disjoint sets: training (90%) and test (10% = 1/k)
  3. Apply learning algorithm to training set
  4. Measure performance with the test set
  5. Repeat steps 2-4, with the 10 different portions
  6. Average the results of the 10 experiments



# Topics

- ***Evaluation of Learning Approach***
- Unsupervised Learning

# Unsupervised Learning

- Learn without labeled examples
  - i.e. X is given, but not f(X)

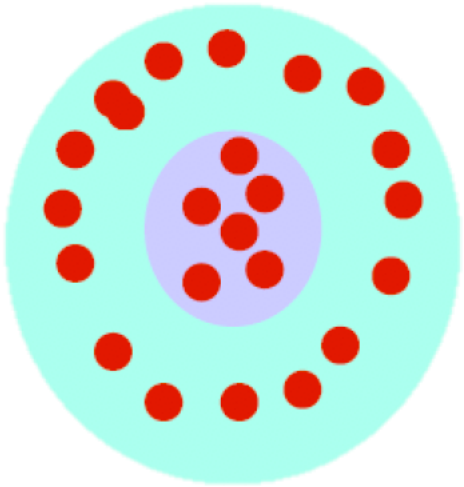
| | | | | |
|---|---|---|---|---|
| **big nose** | **big teeth** | **big eyes** | **no moustache** | ***not given*** |
| **small nose** | **small teeth** | **small eyes** | **no moustache** | ***not given*** |

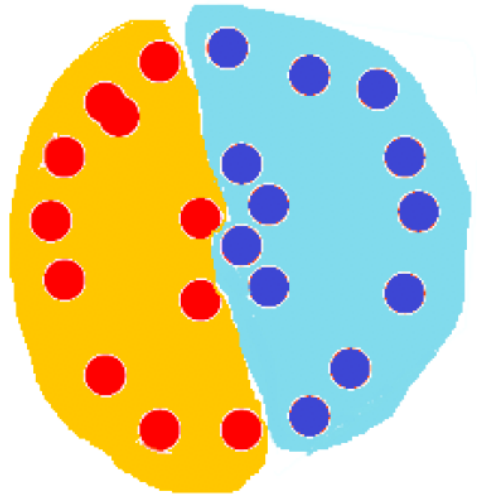
| | | | | |
|---|---|---|---|---|
| **small nose** | **big teeth** | **small eyes** | **moustache** | **f(X) = ?** |

- Without a f(X), you can't really identify/label a test instance
- But you can:
  - Cluster/group the features of the test data into a number of groups
  - Discriminate between these groups without actually labeling them

# What is Clustering

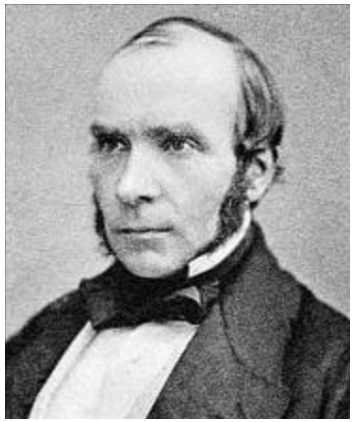
- The organization of unlabeled data into similarity groups called clusters.
- A cluster is a collection of data items which are “similar” between them, and “dissimilar” to data items in other clusters.

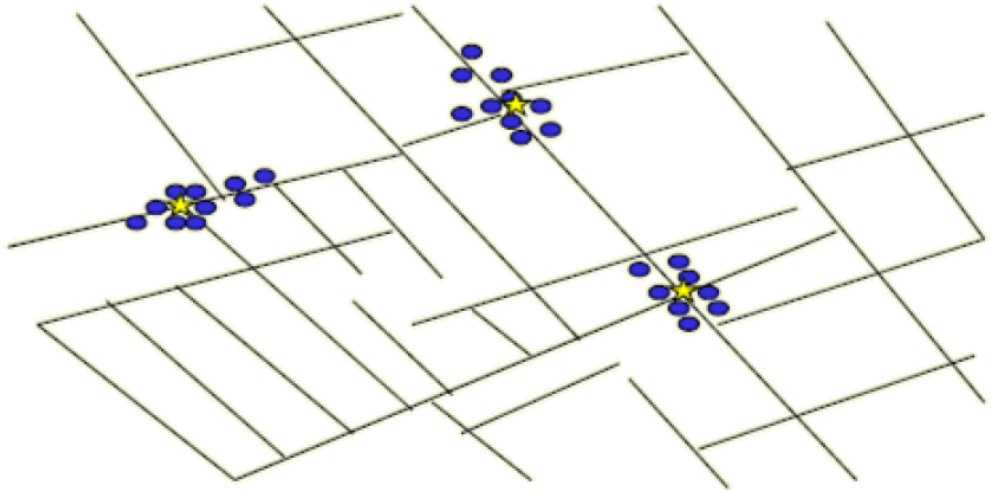




# Historic application of Clustering

- John Snow, a London physician plotted the location of cholera on a map during an outbreak in the 1850s.
- The locations indicated that cases were clustered arounds certain intersections where there were polluted wells – thus exposing both the problem and the solution.

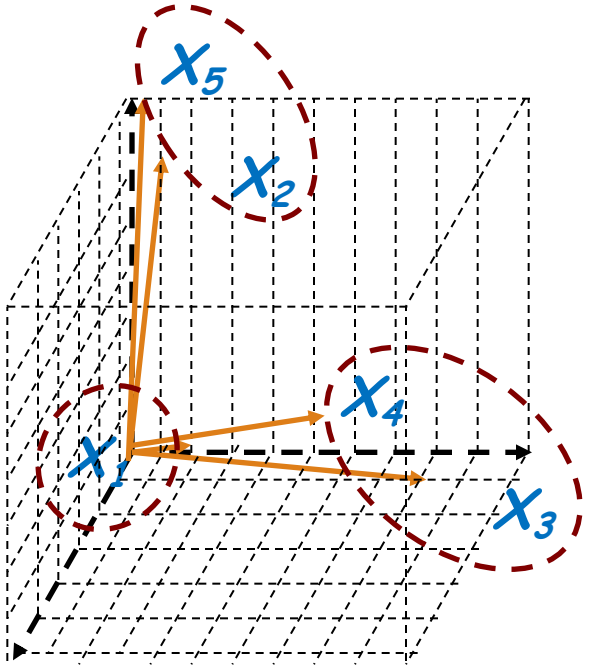




*FROM: Nina Mishra HP Labs*

# Clustering

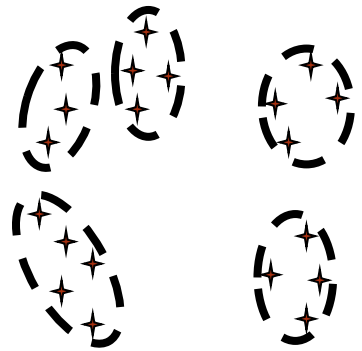
- Represent each instance as a vector <$a_1$, $a_2$, $a_3$,…, $a_n$>
- Each vector can be visually represented in a n dimensional space



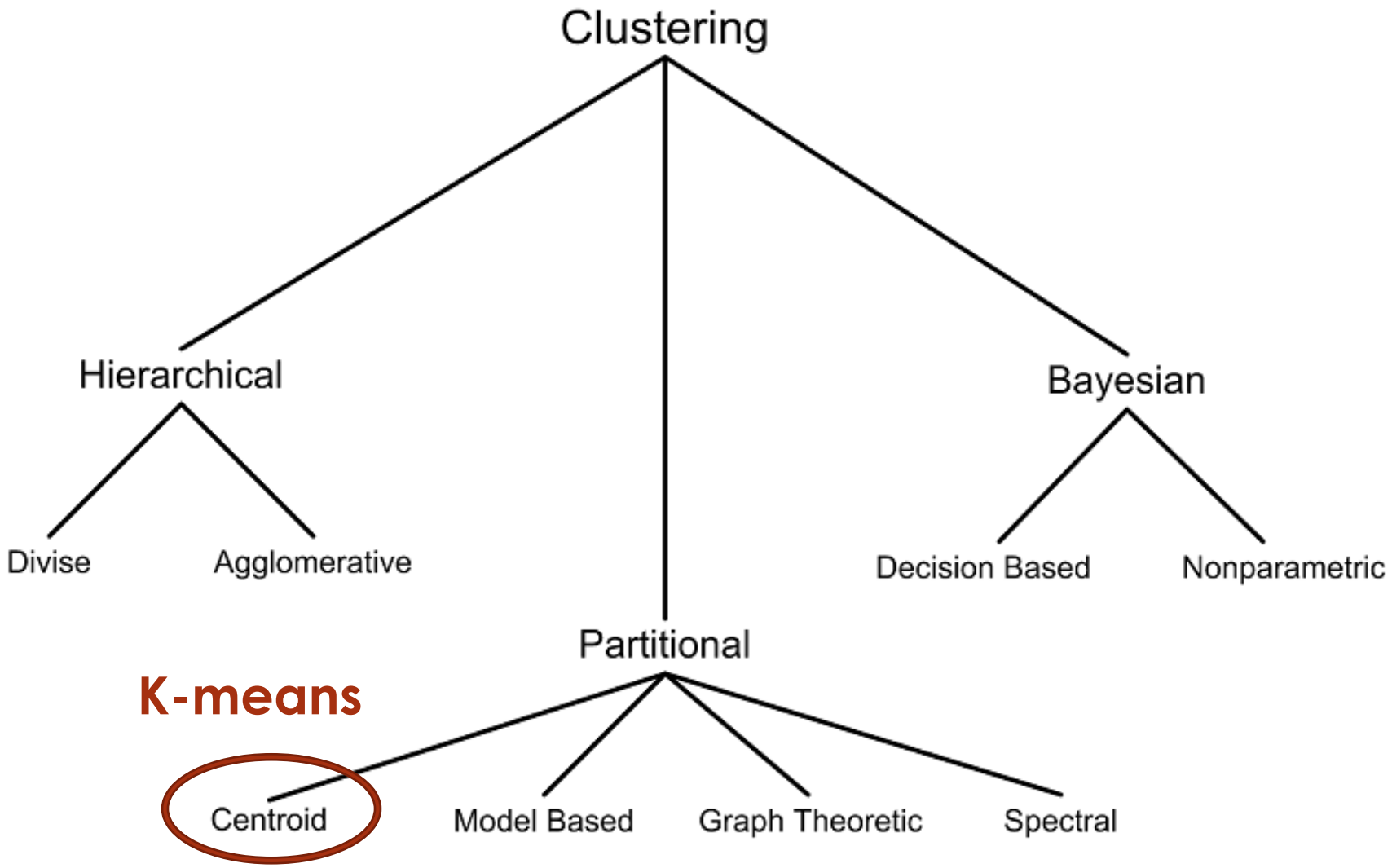

| | $a_1$ | $a_2$ | $a_3$ | Output |
|---|---|---|---|---|
| $X_1$ | 1 | 0 | 0 | ? |
| $X_2$ | 1 | 6 | 0 | ? |
| $X_3$ | 8 | 0 | 1 | ? |
| $X_4$ | 6 | 1 | 0 | ? |
| $X_5$ | 1 | 7 | 1 | ? |

# Clustering

- Clustering Algorithm
  - Represent test instances on a n dimensional space
  - Partition them into regions of high density
    - How? … many algorithms (ex. k-means)
  - Compute the centroïd of each region as the average of data points in the cluster



# Clustering Techniques



# k-means Clustering

- K-means (MacQueen, 1967) is a partitional clustering algorithm
- Let the set of data points D be $\{X_1, X_2, X_3, \ldots, X_n\}$

where $X_i = (X_{i1}, X_{i2}, X_{i3,\ldots,} X_{ir})$ is a vector in $X \subseteq R^r$, and r is the number of dimensions.

- K-means algorithm partitions the given data into k clusters:

-- Each cluster has cluster center, called centroid.

-- k is specified by the user

# k-means Algorithm

- User selects how many clusters they want… (the value of k)

1. Place k points into the space (ex. at random). These points represent initial group centroïds.
2. Assign each data point $x_n$ to the nearest centroïd.
3. When all data points have been assigned, recalculate the positions of the K centroïds as the average of the cluster
4. Repeat Steps 2 and 3 until none of the data instances change group.

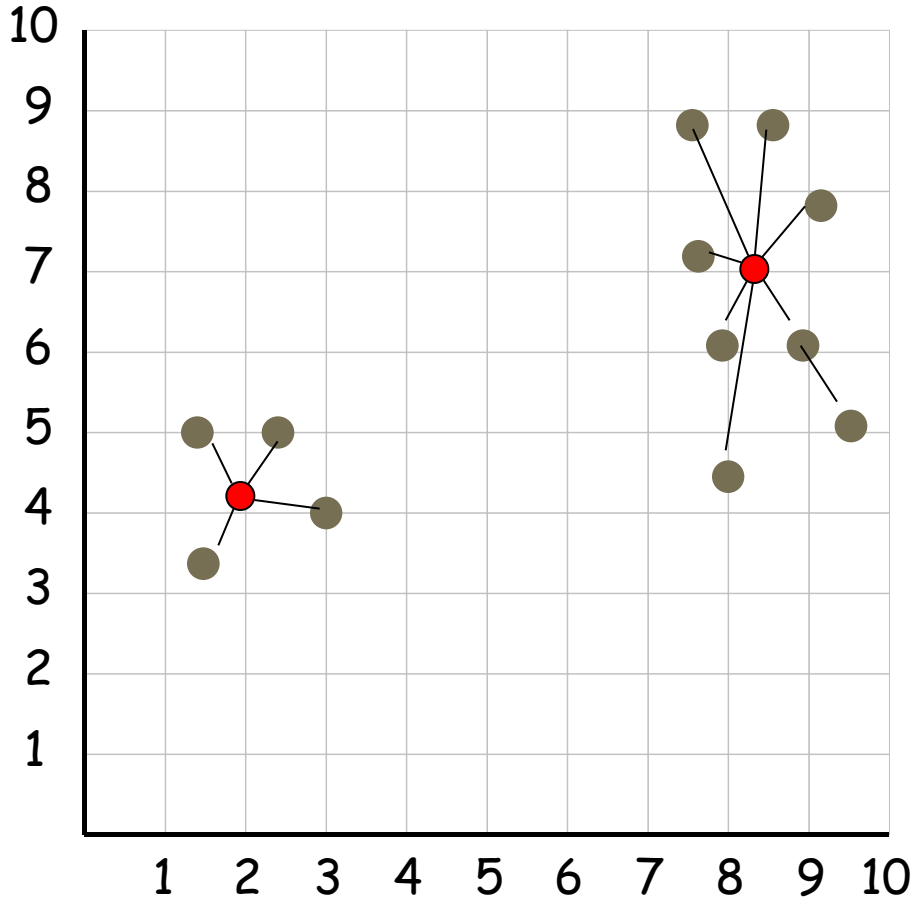
# Euclidean Distance

- To find the nearest centroïd…
- a possible metric is the Euclidean distance
- distance between 2 pts

  $p = (p_1, p_2, ...., p_n)$
  $q = (q_1, q_2, ...., q_n)$

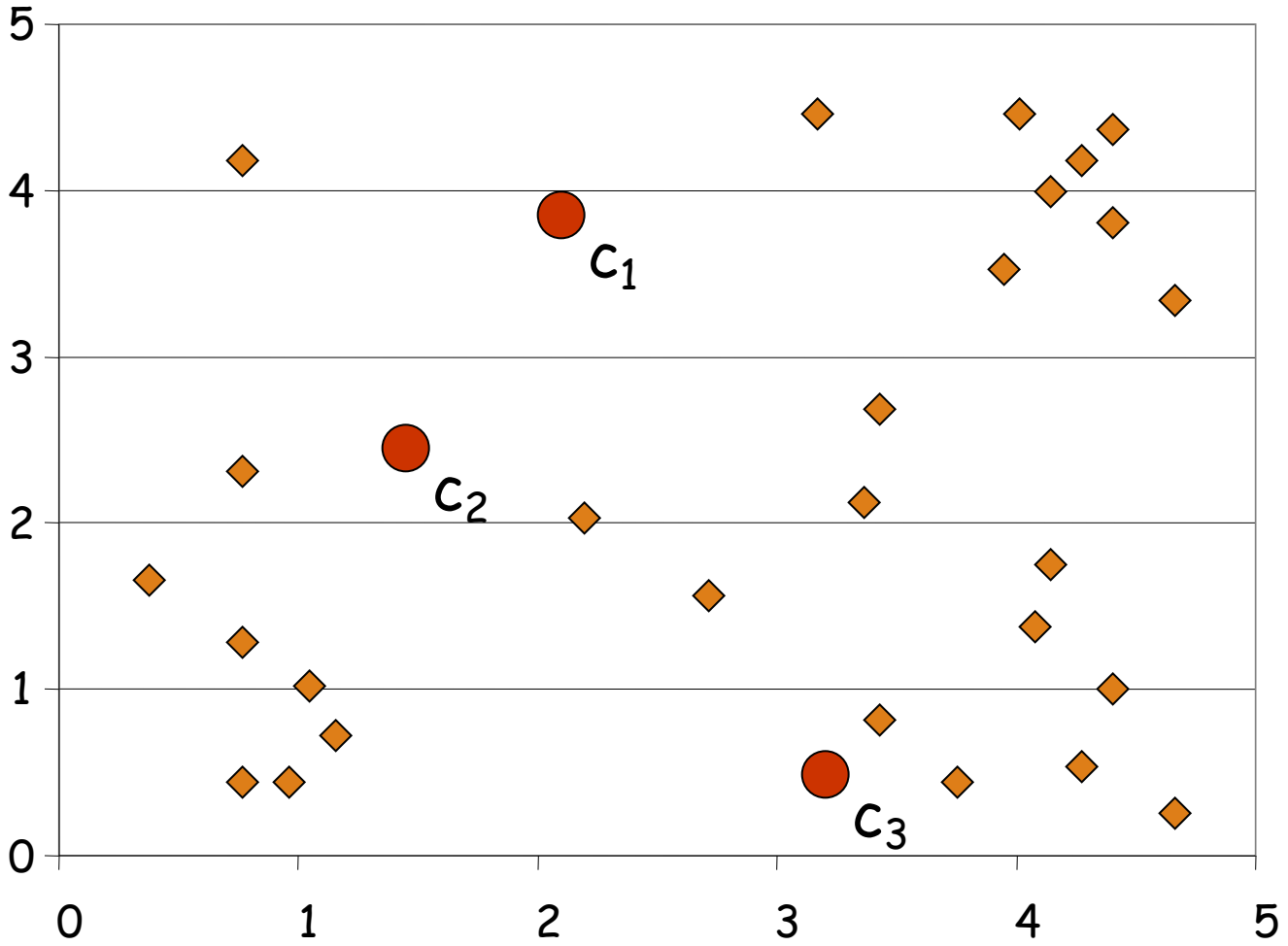
$$d = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

- where to assign a data point x?
- For all k clusters, chose the one where x has the smallest distance.

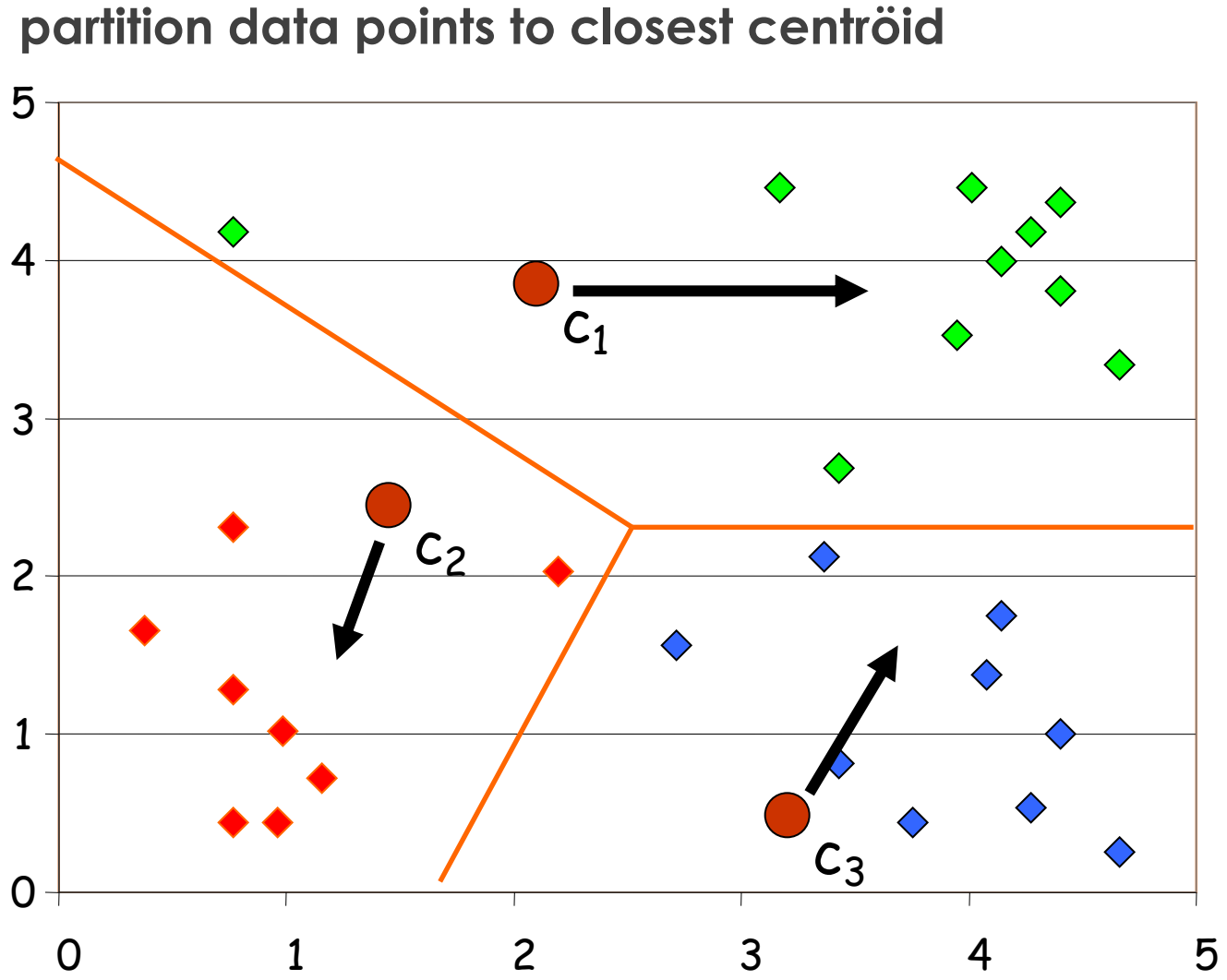


# Example

**initial 3 centroïds (ex. at random) in 2-D i.e. 2 features**

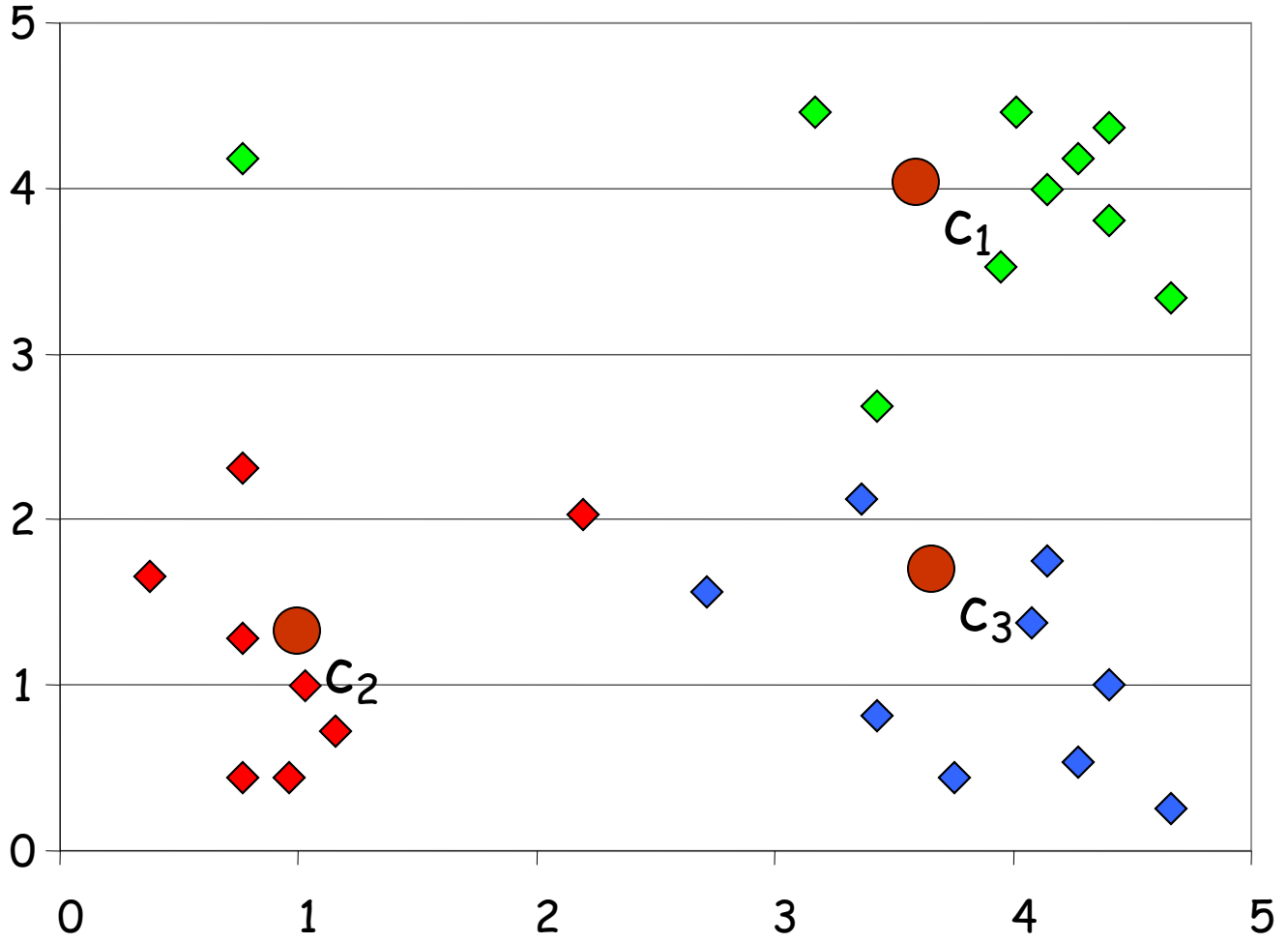


# Example


partition data points to closest centröid
5
4
3
2
1
0
0
1
2
3
4
5
$c_1$
$c_2$
$c_3$

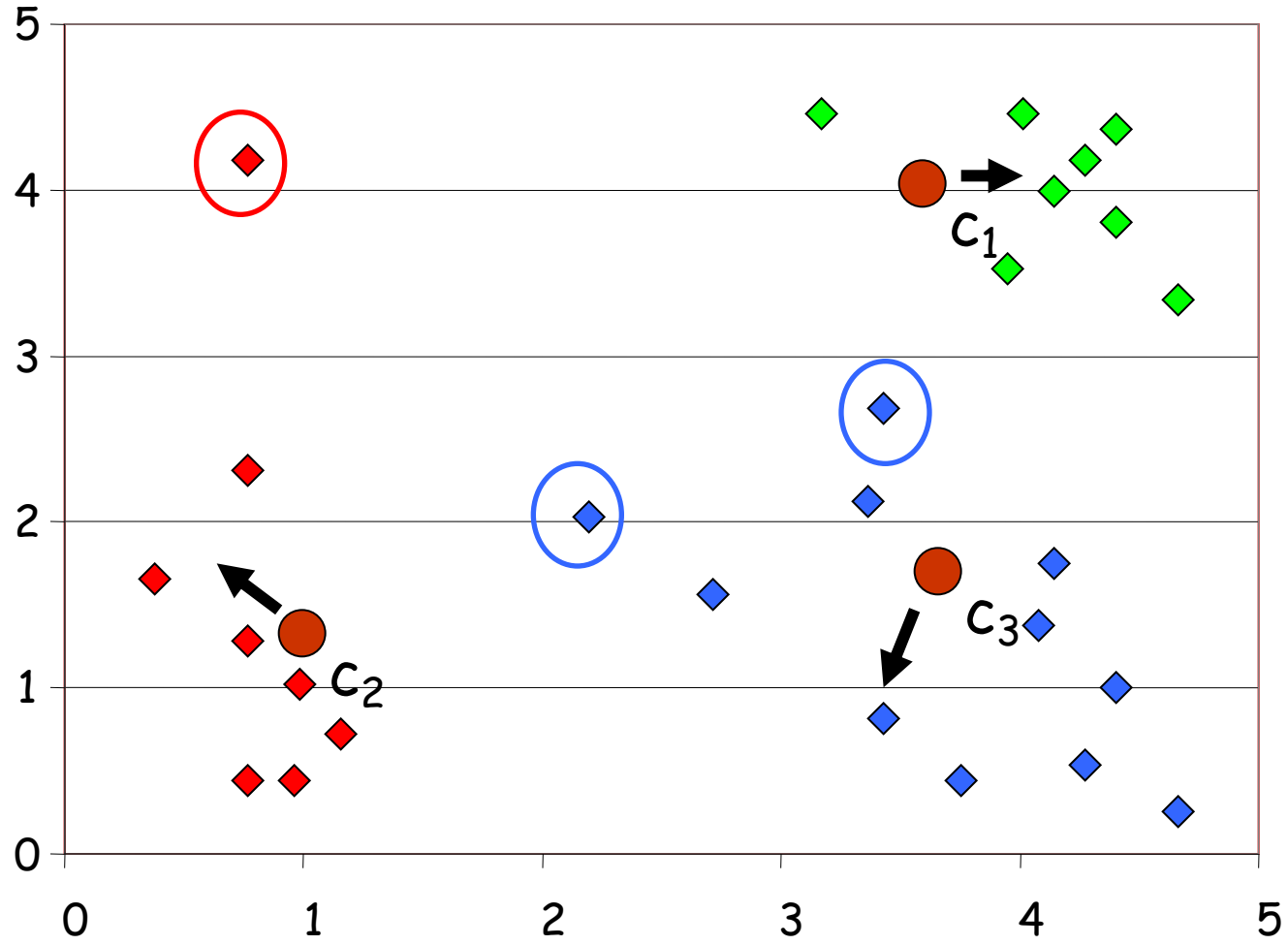
# Example

**re-compute new centroïds**

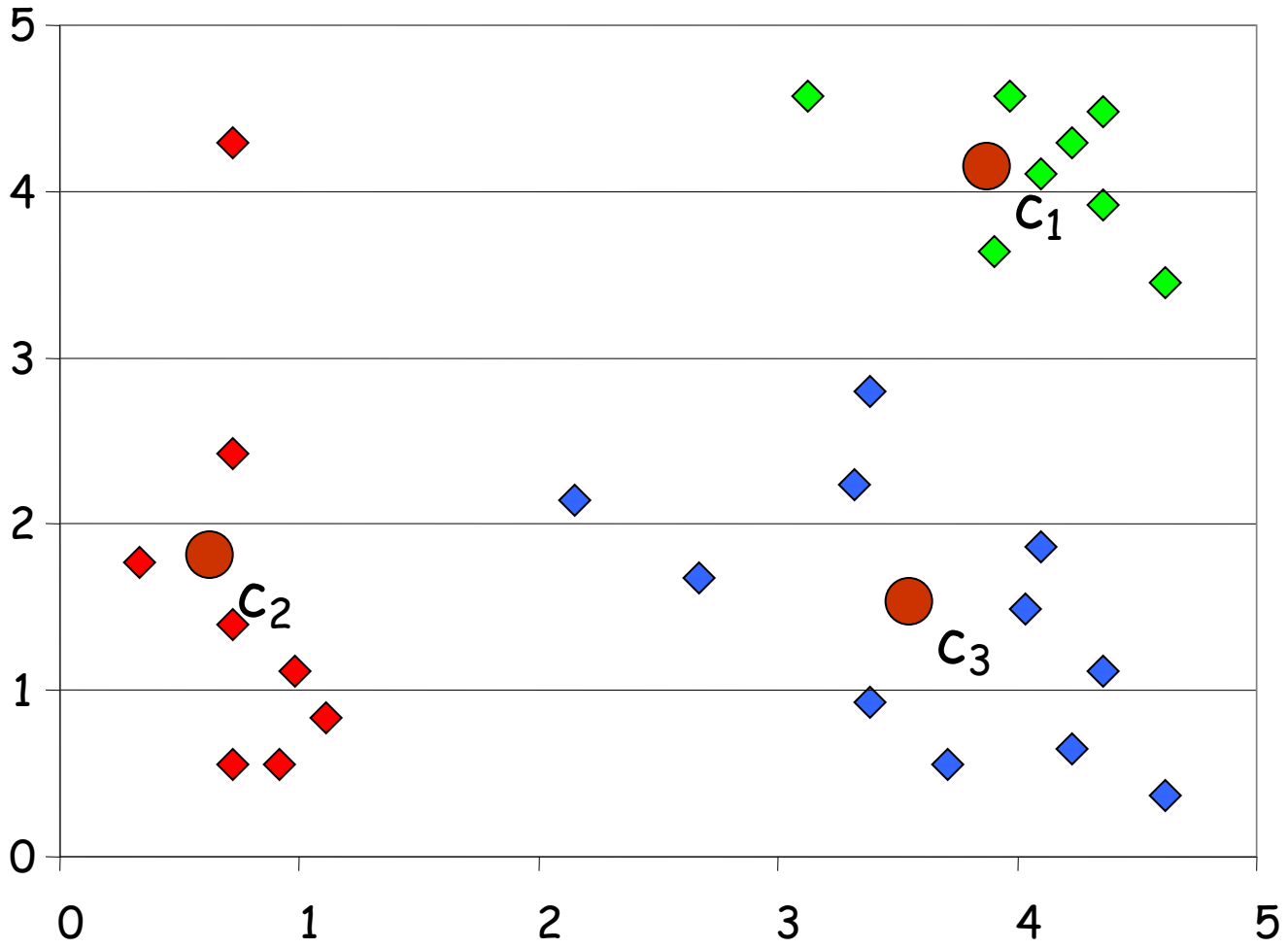


# Example

**re-assign data points to new closest centroïds**



# Example



# 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. The global optimum is hard to find due to complexity.

# Weakness of K-means

- The algorithm is only applicable if the mean is defined.

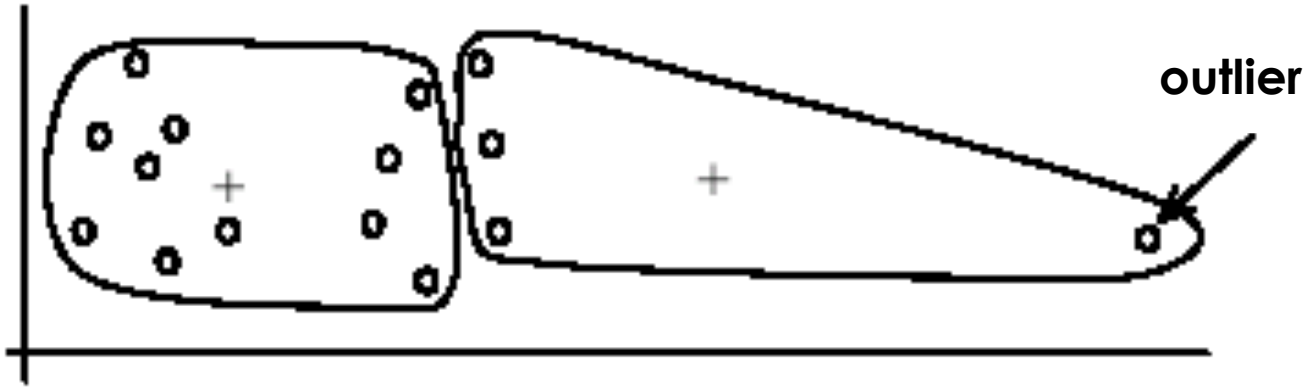
– For categorical data, *k*-mode - the centroid is represented by most frequent values.

- The user needs to specify k.
- The algorithm is sensitive to **outliers**

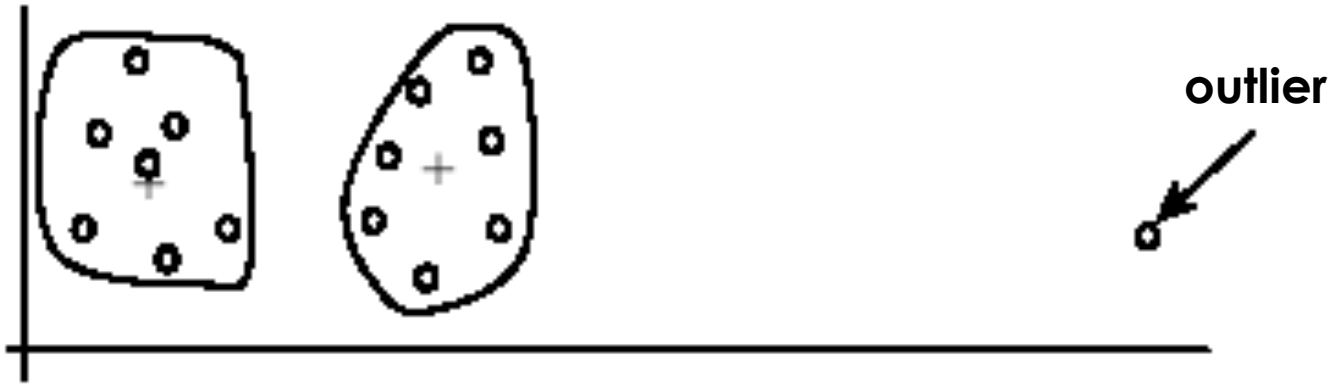
– Outliers are data points that are very far away from other data points.

– 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

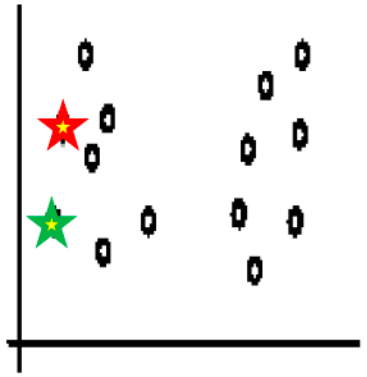
- Remove some data points that are much further away from the centroids than other data points

– To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.

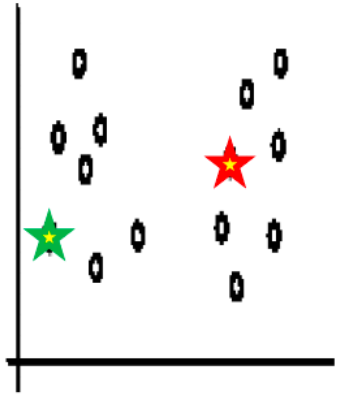
- Perform random sampling: by choosing a small subset of the data points, the chance of selecting an outlier is much smaller

– Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

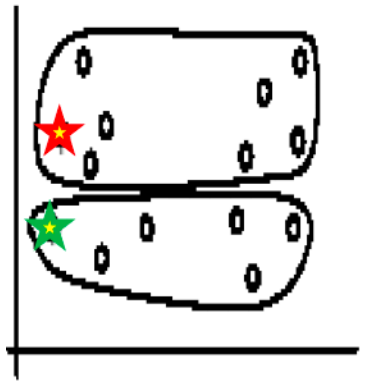
# Sensitivity to initial seeds



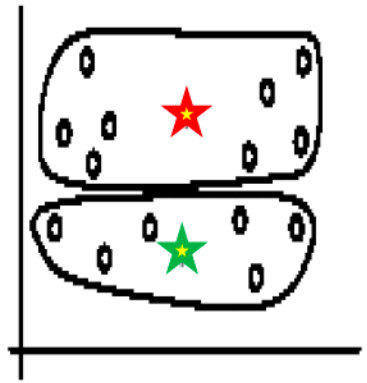
Random selection of seeds (centroids)



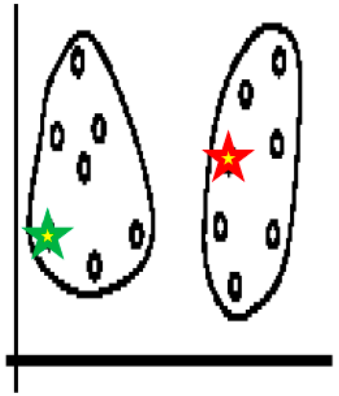
Random selection of seeds (centroids)



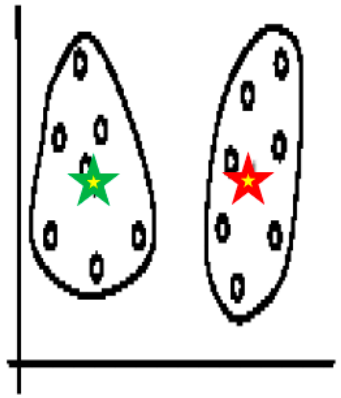
Iteration 1



Iteration 2



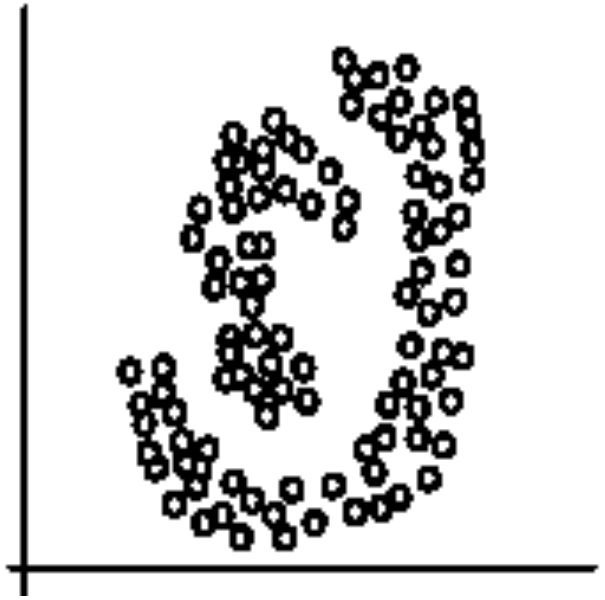
Iteration 1



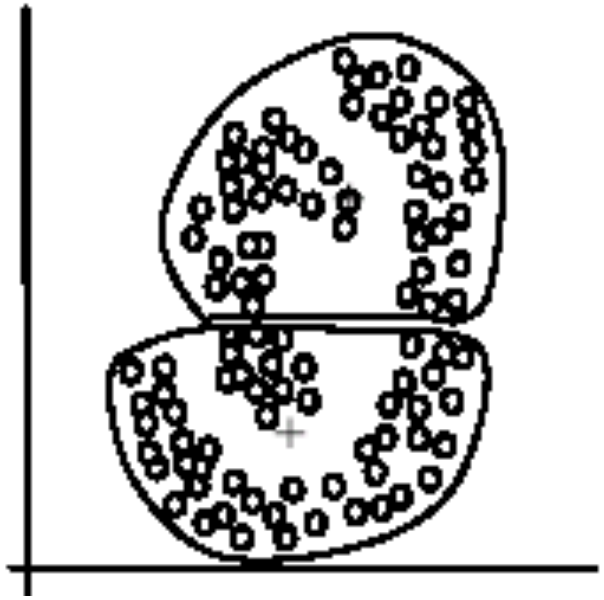
Iteration 2

# 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!

# The End