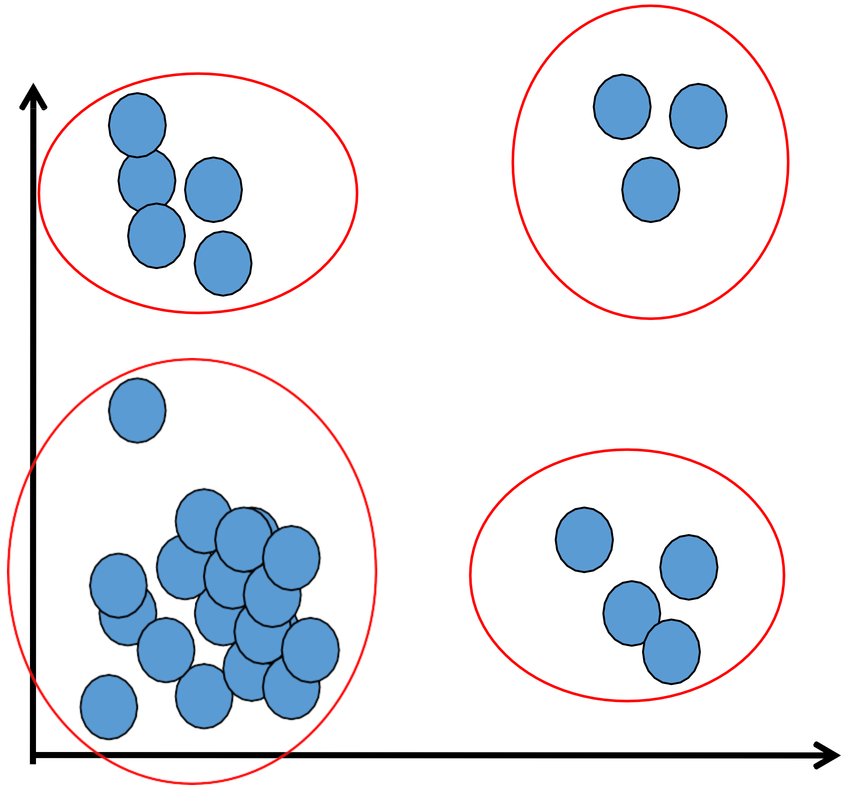
Unsupervised Learning

Presenter: Samuel OCEN

Content

- Motivation
- Introduction
- Applications
- Types of clustering
- Clustering criterion functions
- Distance functions
- Normalization
- Which clustering algorithm to use?
- Cluster evaluation
- Summary

Motivation



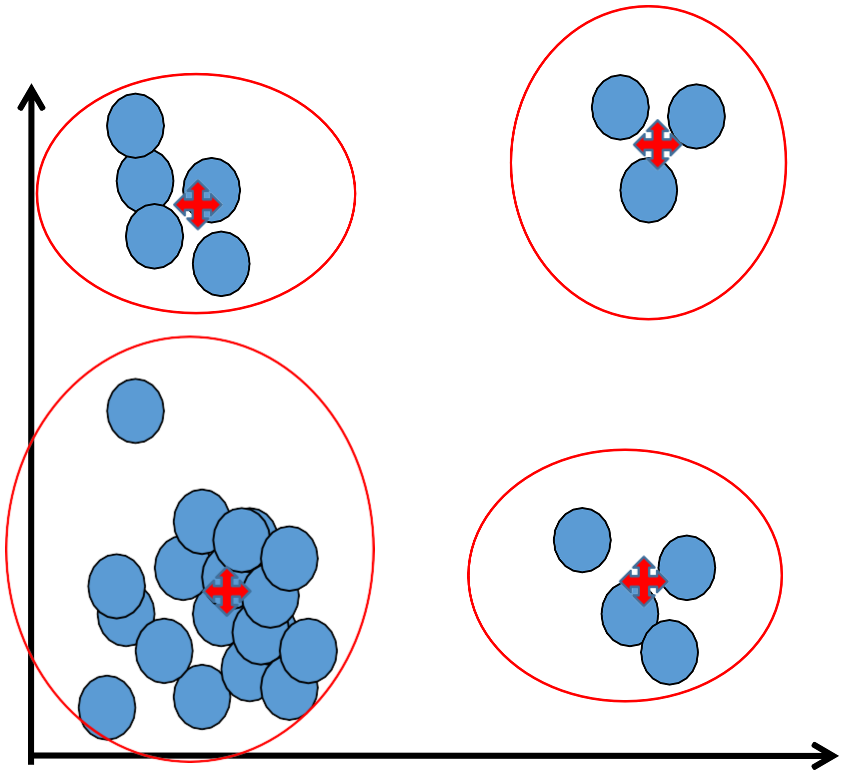
- The goal of clustering is to
  - group data points that are close (or **similar**) to each other
  - identify such groupings (or clusters) in an **unsupervised** manner

- How to define similarity ?
- How many iterations for checking cluster quality ?

Introduction

- Supervised learning: discover patterns in the data with known target (class) or label.
  - These patterns are then utilized to predict the values of the target attribute in future data instances.
  - Examples ?
- Unsupervised learning: The data have no target attribute.
  - We want to explore the data to find some intrinsic structures in them.
  - Can we perform regression here ?
  - Examples ?

Cluster



- A cluster is represented by a single point, known as centroid (or cluster center) of the cluster.
- Centroid is computed as the mean of all data points in a cluster

$$C_j = \sum x_i$$

- Cluster boundary is decided by the farthest data point in the cluster.

Applications

- **Example 1**: groups people of similar sizes together to make “small”, “medium” and “large” T-Shirts.
  - Tailor-made for each person: too expensive
  - One-size-fits-all: does not fit all.
- **Example 2**: In marketing, segment customers according to their similarities
  - To do targeted marketing.
- **Example 3**: Given a collection of text documents, we want to organize them according to their content similarities,
  - To produce a topic hierarchy

Content

- Motivation
- Introduction
- Applications
- **Types of clustering**
- Clustering criterion functions
- Distance functions
- Normalization
- Which clustering algorithm to use?
- Cluster evaluation
- Summary

Types of clustering

- Clustering: Task of grouping a set of data points such that data points in the same group are more similar to each other than data points in another group (group is known as cluster)
  - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.

Types:

1. Exclusive Clustering: K-means
2. Overlapping Clustering: Fuzzy C-means
3. Hierarchical Clustering: Agglomerative clustering, divisive clustering
4. Probabilistic Clustering: Mixture of Gaussian models

1. Exclusive clustering: K-means

- Basic idea: randomly initialize the *k* cluster centers, and iterate between the two steps we just saw.
  1. Randomly initialize the cluster centers, $c_1, \ldots, c_K$
  2. Given cluster centers, determine points in each cluster
     - For each point p, find the closest c_i. Put p into cluster i
  3. Given points in each cluster, solve for c_i
     - Set c_i to be the mean of points in cluster i
  4. If c_i have changed, repeat Step 2

Properties

- Will always converge to *some* solution
- Can be a "local minimum"
  - does not always find the global minimum of objective function:

$$\sum_{\text{clusters } i} \; \sum_{\text{points p in cluster } i} ||p - c_i||^2$$

K-means contd..

- Algorithm

Begin

initialize n, c, μ_1, μ_2, …, μ_c (randomly selected)

do classify n samples according to nearest μ_i

recompute μ_i

until no change in μ_i

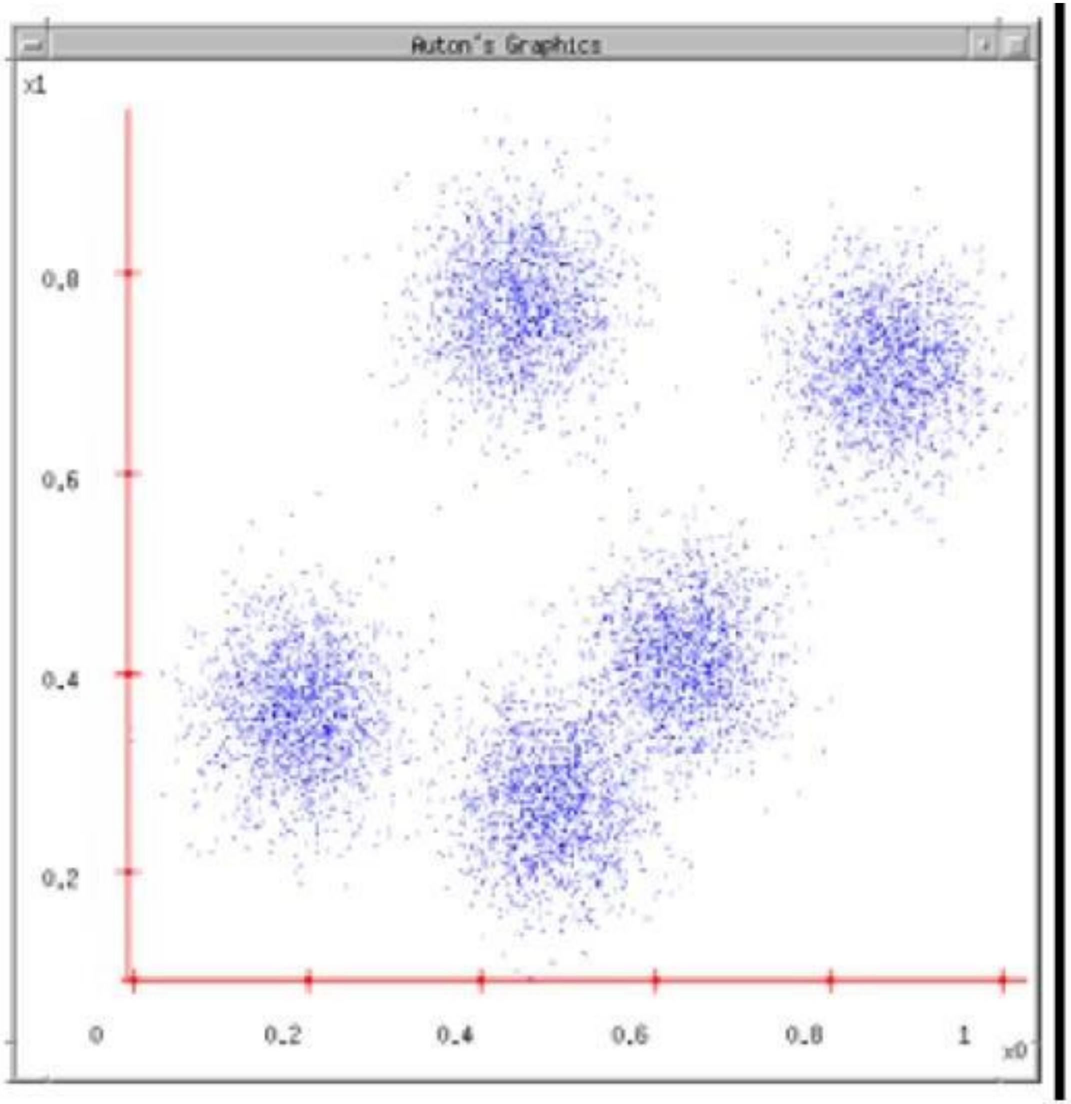
return μ_1, μ_2, …, μ_c

End

K-means example

K-means

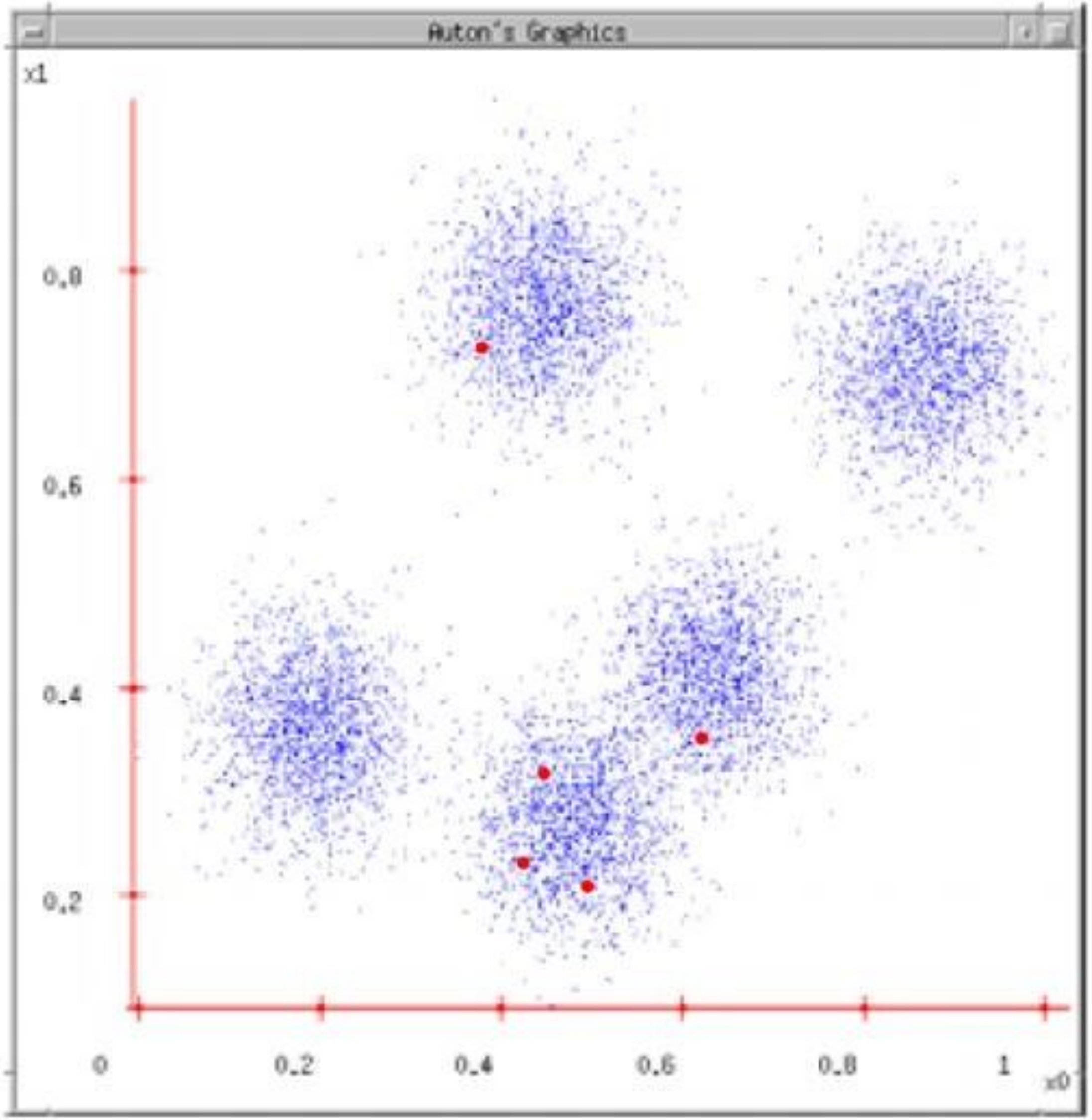
1. Ask user how many clusters they'd like. *(e.g. k=5)*



Example contd..

K-means

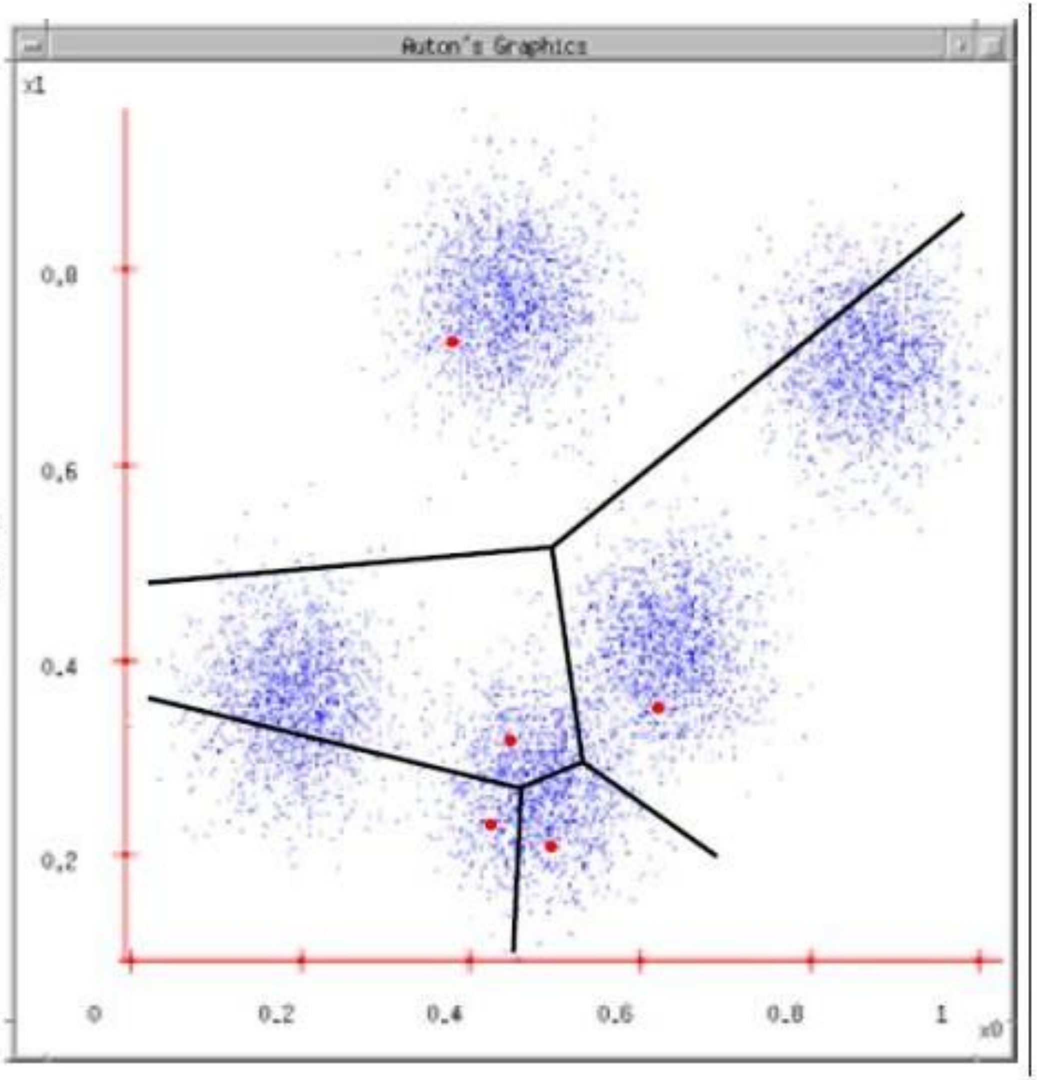
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



Example contd..

K-means

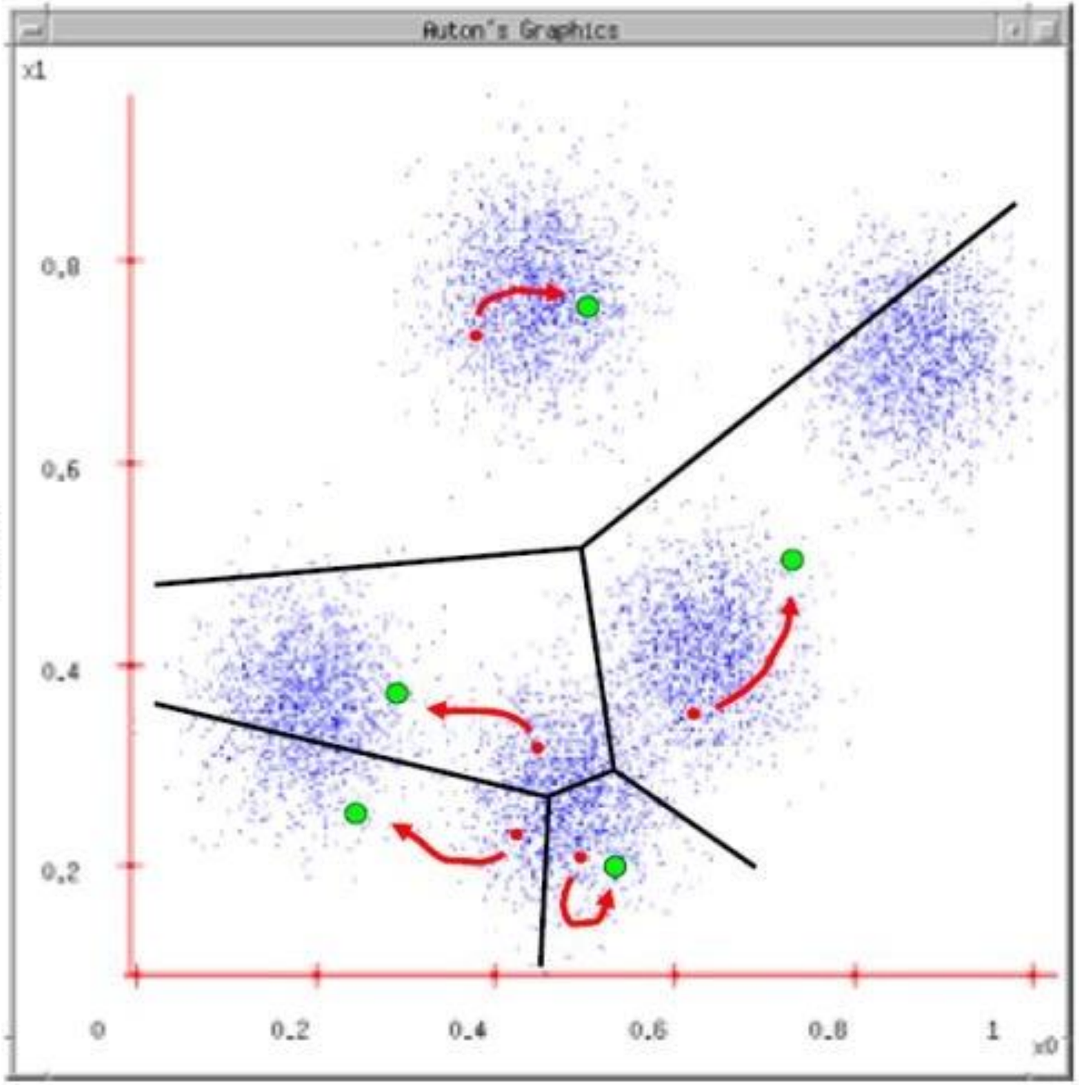
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



Example contd..

K-means

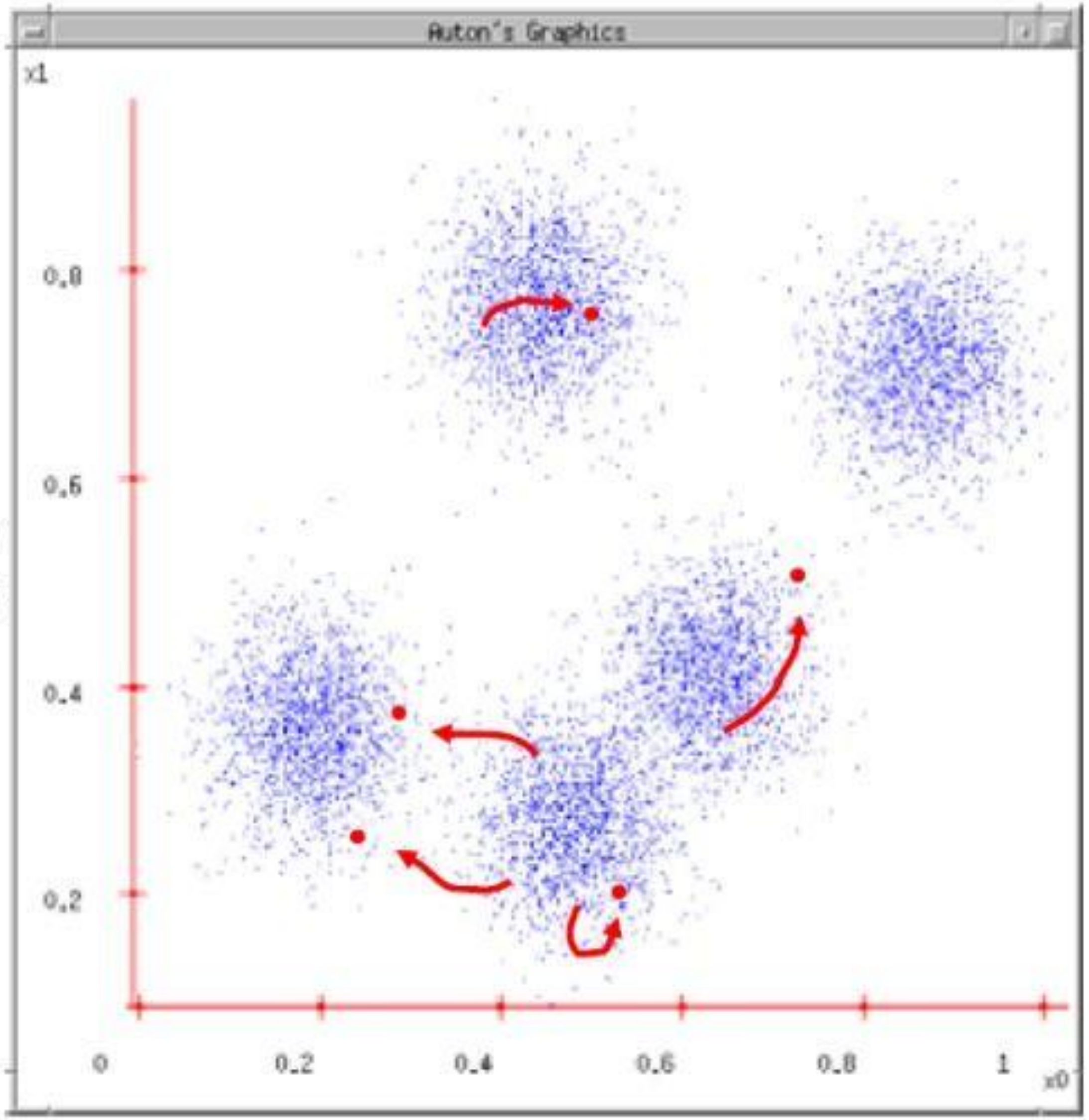
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



Example contd..

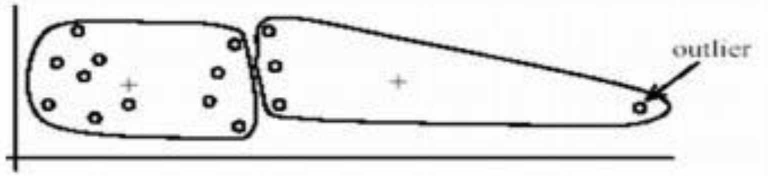
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!

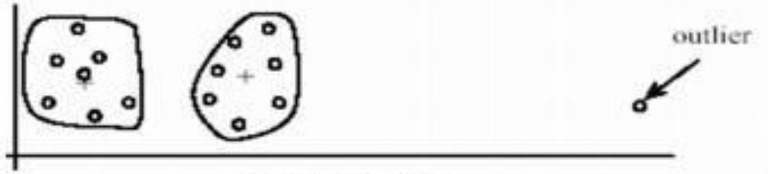


Contd..

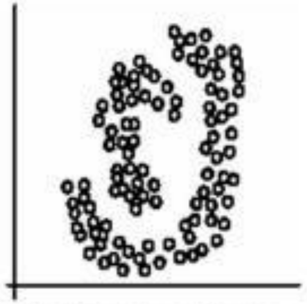
- Pros
  - Simple, fast to compute
  - Converges to local minimum of within-cluster squared error
- Cons
  - Setting k?
  - Sensitive to initial centers
  - Sensitive to outliers
  - Detects spherical clusters
  - Assuming means can be computed



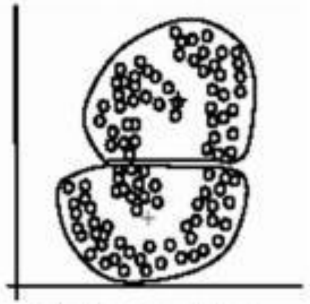
(A): Undesirable clusters



(B): Ideal clusters



(A): Two natural clusters



(B): k-means clusters

2. Fuzzy C-Means Clustering

- One data point may belong to two or more cluster with different memberships.
- Objective function:

$$J = \sum_{j=1}^{K} \sum_{i=1}^{n} u_{i,j}^{m} ||x_i^j - c_j||^2$$

where $1 \leq m < \infty$

- An extension of k-means

Fuzzy c-means algorithm

- Let x_i be a vector of values for data point g_i.

1. Initialize membership $U^{(0)} = [\ u_{ij}\ ]$ for data point g_i of cluster cl_j by random
2. At the k-th step, compute the fuzzy centroid $C^{(k)} = [\ c_j\ ]$ for $j = 1, .., nc$, where nc is the number of clusters, using

$$c_j = \frac{\sum_{i=1}^{n} (u_{ij})^m x_i}{\sum_{i=1}^{n} (u_{ij})^m}$$

where m is the fuzzy parameter and n is the number of data points.

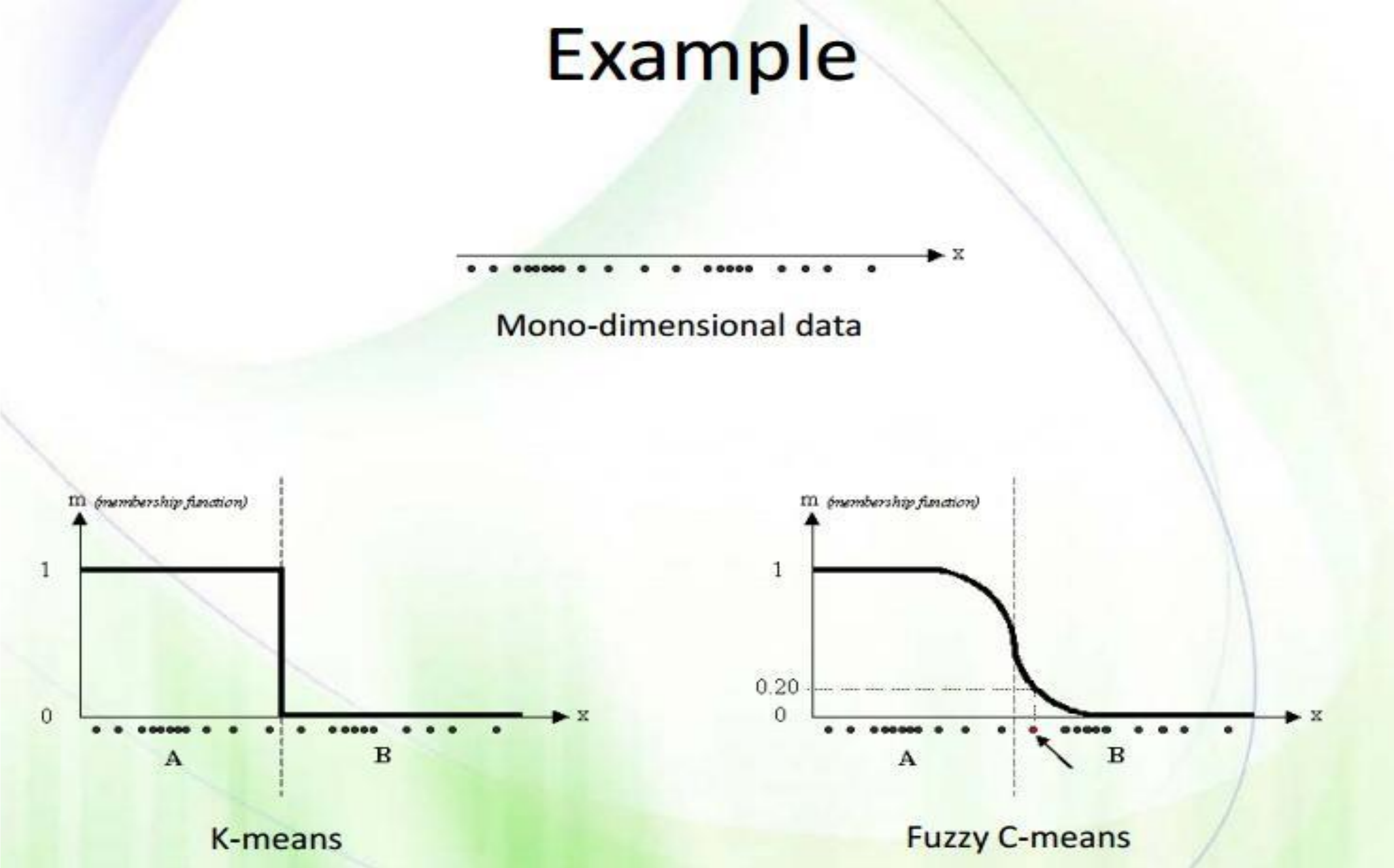
Fuzzy c-means algorithm

3. Update the fuzzy membership $U^{(k)} = [\ u_{ij}\ ]$, using

$$u_{ij} = \frac{\left(\dfrac{1}{\left\| x_i - c_j \right\|}\right)^{\frac{1}{(m-1)}}}{\sum_{j=1}^{n_c}\left(\dfrac{1}{\left\| x_i - c_j \right\|}\right)^{\frac{1}{(m-1)}}}$$

4. If $\|U^{(k)} - U^{(k-1)}\| < \varepsilon$, then STOP, else return to step 2.
5. Determine membership cutoff
   - For each data point g_i, assign g_i to cluster cl_j if u_{ij} of $U^{(k)} > \alpha$

Example

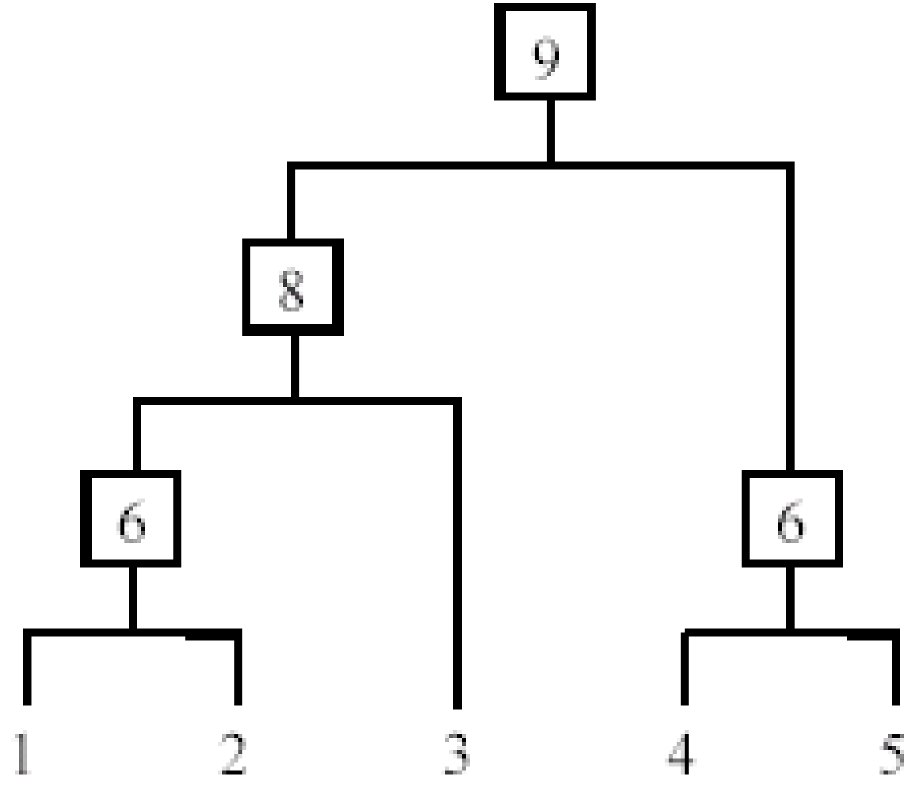

Example
x
Mono-dimensional data
m (membership function)
1
0
x
A
B
K-means
m (membership function)
1
0.20
0
x
A
B
Fuzzy C-means

Fuzzy c-means

- Pros:
  - Allows a data point to be in multiple clusters
  - A more natural representation of the behavior of genes
    - genes usually are involved in multiple functions
- Cons:
  - Need to define c (k in K-means), the number of clusters
  - Need to determine membership cutoff value
  - Clusters are sensitive to initial assignment of centroids
    - Fuzzy c-means is not a deterministic algorithm

3. Hierarchical Clustering

- Produce a nested sequence of clusters, a tree, also called Dendrogram.



Types of hierarchical clustering

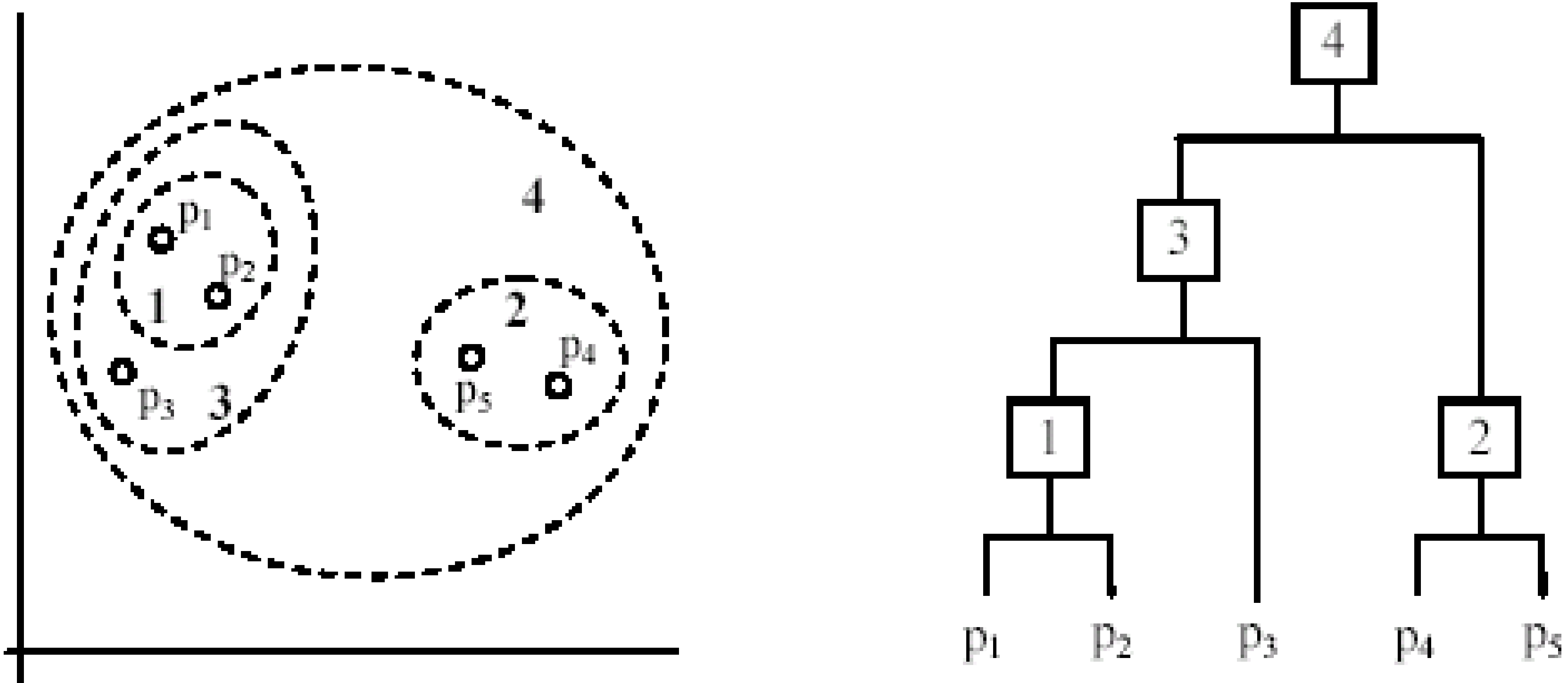
- Agglomerative (bottom up) clustering: It builds the dendrogram (tree) from the bottom level, and
  - merges the most similar (or nearest) pair of clusters
  - stops when all the data points are merged into a single cluster (i.e., the root cluster).
- Divisive (top down) clustering: It starts with all data points in one cluster, the root.
  - Splits the root into a set of child clusters. Each child cluster is recursively divided further
  - stops when only singleton clusters of individual data points remain, i.e., each cluster with only a single point

Agglomerative clustering

It is more popular then divisive methods.

- At the beginning, each data point forms a cluster (also called a node).
- Merge nodes/clusters that have the least distance.
- Go on merging
- Eventually all nodes belong to one cluster

An example: working of the algorithm



(A). Nested clusters

(B) Dendrogram

Hierarchical clustering

- Pros
  - Dendograms are great for visualization
  - Provides hierarchical relations between clusters
  - Shown to be able to capture concentric clusters
- Cons
  - Not easy to define levels for clusters
  - Experiments showed that other clustering techniques outperform hierarchical clustering

4. Probabilistic clustering

- Gaussian mixture models

Content

- Motivation
- Introduction
- Applications
- Types of clustering
- **Clustering criterion functions**
- Distance functions
- Data standardization
- Which clustering algorithm to use?
- Cluster evaluation
- Summary

Clustering criterion ..

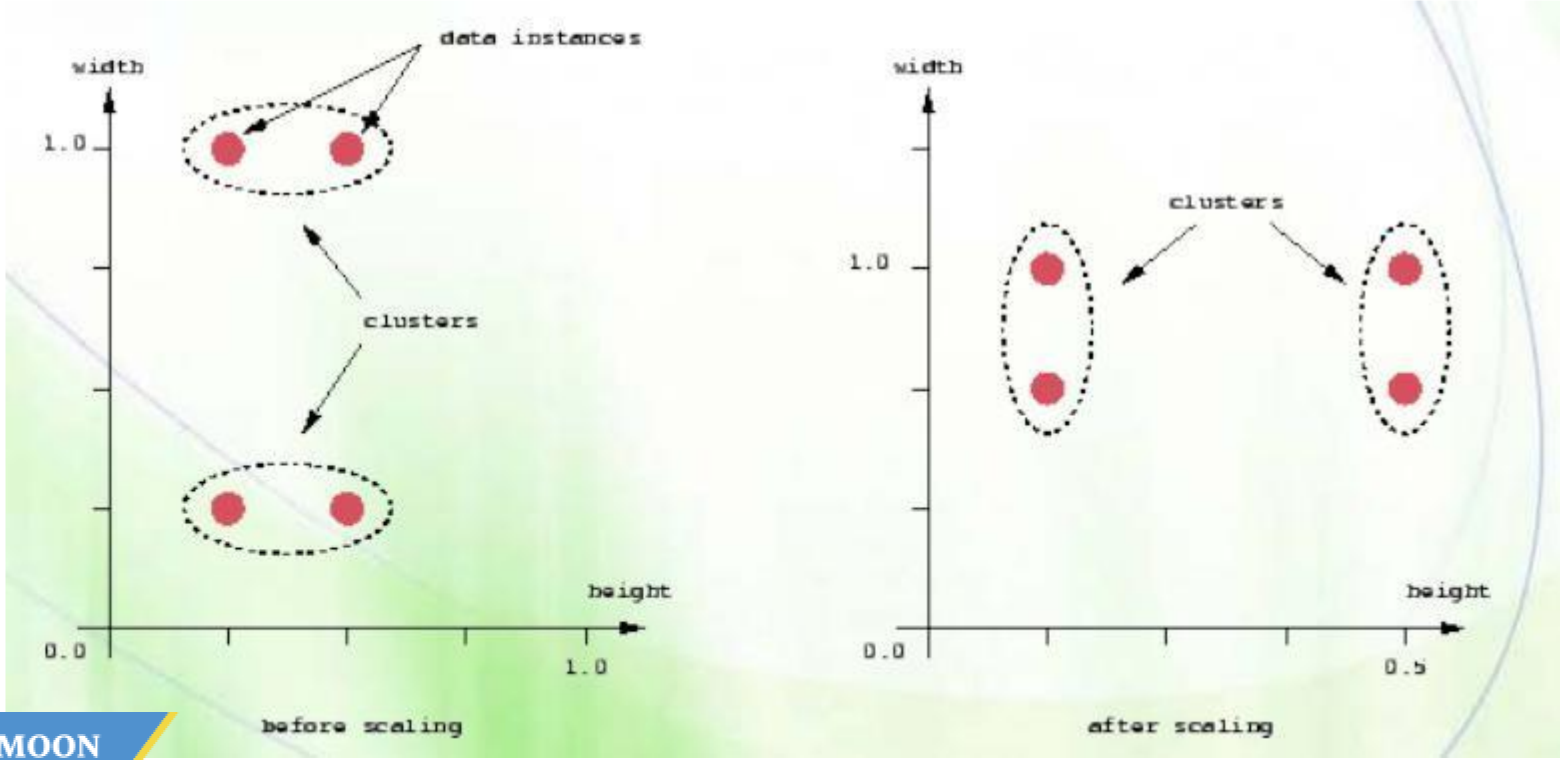
1. Similarity function

2. Stopping criterion

3. Cluster Quality

1. Similarity function / Distance measure

- How to find distance b/w data points
- Euclidean distance:
  - Problems with Euclidean distance



Euclidean distance and Manhattan distance

- Euclidean distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2}$$

- Manhattan distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = | x_{i1} - x_{j1} | + | x_{i2} - x_{j2} | + \ldots + | x_{ir} - x_{jr} |$$

- Weighted Euclidean distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \ldots + w_r(x_{ir} - x_{jr})^2}$$

Squared distance and Chebychev distance

- Squared Euclidean distance: to place progressively greater weight on data points that are further apart.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2$$

- Chebychev distance: one wants to define two data points as "different" if they are different on any one of the attributes.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \max(|x_{i1} - x_{j1}|, |x_{i2} - x_{j2}|, \ldots, |x_{ir} - x_{jr}|)$$

Distance functions for binary and nominal attributes

- Binary attribute: has two values or states but no ordering relationships, e.g.,
  - Gender: male and female.
- We use a confusion matrix to introduce the distance functions/measures.
- Let the *i*th and *j*th data points be $\mathbf{x}_i$ and $\mathbf{x}_j$ (vectors)

Confusion matrix

		Data point j		
		1	0	
Data point i	1	a	b	$a+b$
	0	c	d	$c+d$
		$a+c$	$b+d$	$a+b+c+d$

(10)

a: the number of attributes with the value of 1 for both data points.

b: the number of attributes for which $x_{if} = 1$ and $x_{jf} = 0$, where x_{if} (x_{jf}) is the value of the fth attribute of the data point $\mathbf{x}_i$ ($\mathbf{x}_j$).

c: the number of attributes for which $x_{if} = 0$ and $x_{jf} = 1$.

d: the number of attributes with the value of 0 for both data points.

Contd..

- Cosine similarity

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

- Euclidean distance

$$d(x, y) = \sqrt{\sum (x_i - y_i)^2}$$

- Minkowski Metric

$$d_p(x_i, y_j) = \left(\sum_{K=1}^{d} |x_{i,k} - x_{i,k}|^p\right)^{\frac{1}{p}}$$

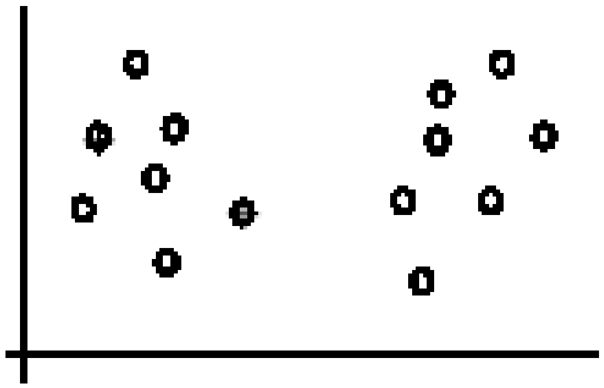
2. Stopping criteria

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

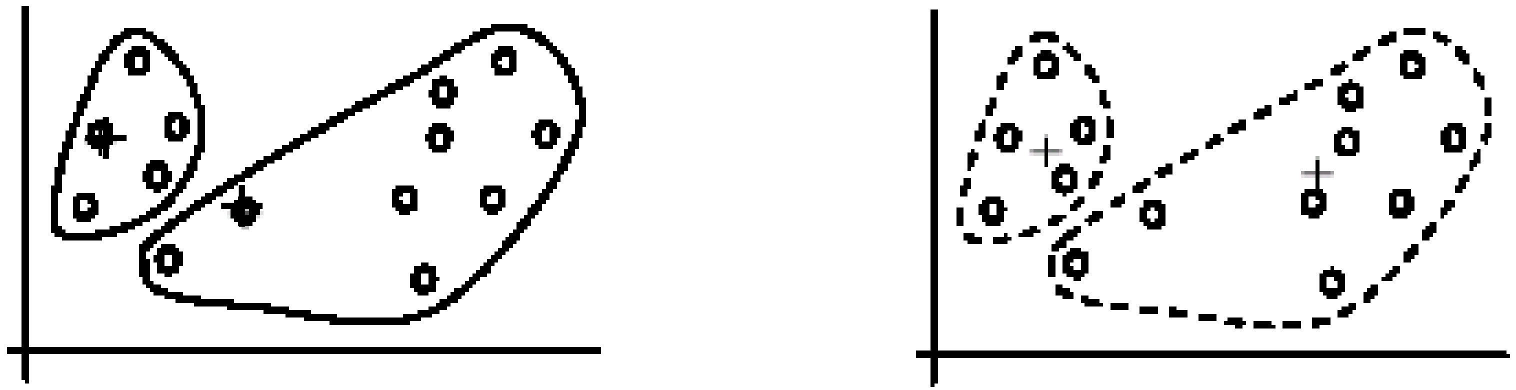
$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2$$

- C_i 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), and $dist(\mathbf{x}, \mathbf{m}_j)$ is the distance between data point $\mathbf{x}$ and centroid $\mathbf{m}_j$.

An example



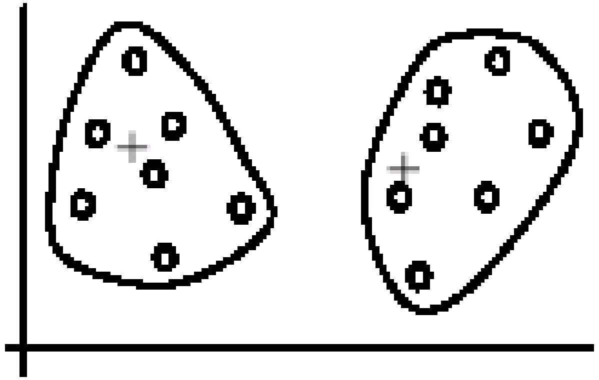
(A). Random selection of *k* centers



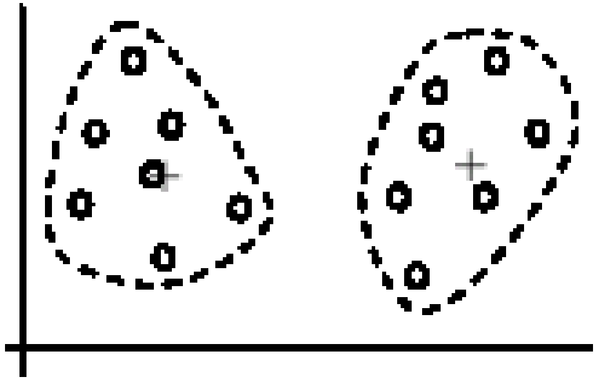
Iteration 1: (B). Cluster assignment

(C). Re-compute centroids

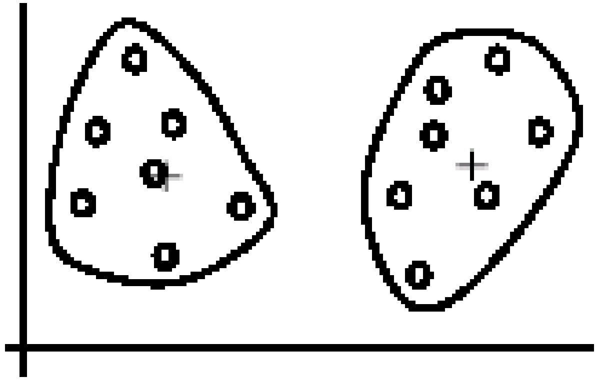
An example (cont …)



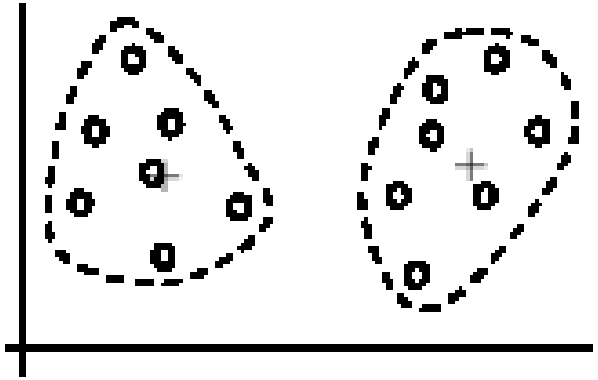
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



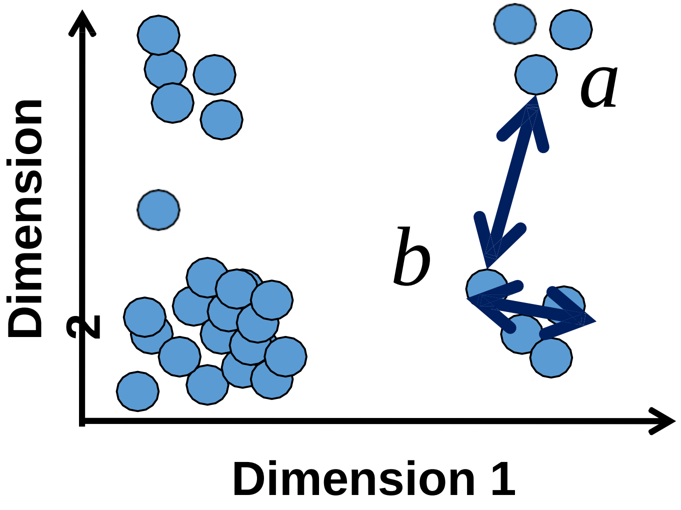
Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

3. Cluster quality

- **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.



Content

- Motivation
- Introduction
- Applications
- Types of clustering
- Clustering criterion functions
- Distance functions
- **Normalization**
- Which clustering algorithm to use?
- Cluster evaluation
- Summary

Normalization

- Technique to force the attributes to have a common value range
- What is the need ?
  - Consider the following pair of data points

    $\mathbf{x}_i$: (0.1, 20) and $\mathbf{x}_j$: (0.9, 720).

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(0.9 - 0.1)^2 + (720 - 20)^2} = 700.000457,$$

- Two main approaches to standardize interval scaled attributes, **range** and **z-score**. *f* is an attribute

$$range(x_{if}) = \frac{x_{if} - \min(f)}{\max(f) - \min(f)},$$

Contd..

- **Z-score**: transforms the attribute values so that they have a mean of zero and a **mean absolute deviation** of 1. The mean absolute deviation of attribute *f*, denoted by s_f, is computed as follows

$$m_f = \frac{1}{n}\left(x_{1f} + x_{2f} + \ldots + x_{nf}\right),$$

$$s_f = \frac{1}{n}\left(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|\right),$$

Z-score: $$z(x_{if}) = \frac{x_{if} - m_f}{s_f}.$$

Content

- Motivation
- Introduction
- Applications
- Types of clustering
- Clustering criterion functions
- Distance functions
- Normalization
- **Which clustering algorithm to use?**
- Cluster evaluation
- Summary

How to choose a clustering algorithm

- A vast collection of algorithms are available. Which one to choose for our problem ?
- Choosing the "best" algorithm is a challenge.
  - Every algorithm has limitations and works well with certain data distributions.
  - It is very hard, if not impossible, to know what distribution the application data follow. The data may not fully follow any "ideal" structure or distribution required by the algorithms.
  - One also needs to decide how to standardize the data, to choose a suitable distance function and to select other parameter values.

Contd..

- Due to these complexities, the common practice is to
  - run several algorithms using different distance functions and parameter settings, and
  - then carefully analyze and compare the results.
- The interpretation of the results must be based on insight into the meaning of the original data together with knowledge of the algorithms used.
- Clustering is highly application dependent and to certain extent subjective (personal preferences).

Content

- Motivation
- Introduction
- Applications
- Types of clustering
- Clustering criterion functions
- Distance functions
- Normalization
- Which clustering algorithm to use?
- **Cluster evaluation**
- Summary

Cluster Evaluation: hard problem

- The quality of a clustering is very hard to evaluate because
  - We do not know the correct clusters
- Some methods are used:
  - User inspection
    - Study centroids, and spreads
    - Rules from a decision tree.
    - For text documents, one can read some documents in clusters.

Cluster evaluation: ground truth

- We use some labeled data (for classification)
- Assumption: Each class is a cluster.
- After clustering, a confusion matrix is constructed. From the matrix, we compute various measurements, entropy, purity, precision, recall and F-score.
  - Let the classes in the data D be $C = (c_1, c_2, \ldots, c_k)$. The clustering method produces k clusters, which divides D into k disjoint subsets, $D_1, D_2, \ldots, D_k$.

Evaluation measures: Entropy

Entropy: For each cluster, we can measure its entropy as follows:

$$entropy(D_i) = -\sum_{j=1}^{k} \Pr_i(c_j)\log_2 \Pr_i(c_j), \tag{29}$$

where $\Pr_i(c_j)$ is the proportion of class c_j data points in cluster i or D_i. The total entropy of the whole clustering (which considers all clusters) is

$$entropy_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times entropy(D_i) \tag{30}$$

Evaluation measures: purity

Purity: This again measures the extent that a cluster contains only one class of data. The purity of each cluster is computed with

$$purity(D_i) = \max_j(\Pr_i(c_j)) \qquad (31)$$

The total purity of the whole clustering (considering all clusters) is

$$purity_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times purity(D_i) \qquad (32)$$

Indirect evaluation

- In some applications, clustering is not the primary task, but used to help perform another task.
- We can use the performance on the primary task to compare clustering methods.
- For instance, in an application, the primary task is to provide recommendations on book purchasing to online shoppers.
  - If we can cluster books according to their features, we might be able to provide better recommendations.
  - We can evaluate different clustering algorithms based on how well they help with the recommendation task.
  - Here, we assume that the recommendation can be reliably evaluated.

Content

- Motivation
- Introduction
- Applications
- Types of clustering
- Clustering criterion functions
- Distance functions
- Normalization
- Which clustering algorithm to use?
- Cluster evaluation
- **Summary**

Summary

- Studied need for unsupervised learning
- Types of clustering:
  - K-means, Fuzzy C, hierarchical
- Similarity functions:
  - Euclidean distance, Manhattan distance
- Stopping criteria:
  - SSD
- Which algorithm to choose ?
- Cluster evaluation

Thank you

Contact:

Samuel OCEN
samuel.ocen@mmu.ac.ug
Class Schedule: Thursday