CSC345/M45:
Big Data & Machine Learning
(clustering: part one)

Dr. Xianghua Xie

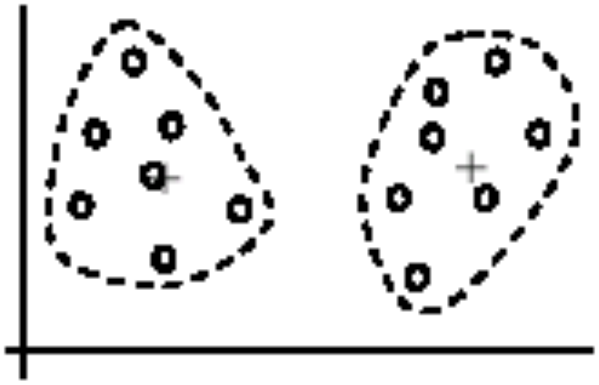
x.xie@swansea.ac.uk
http://csvision.swan.ac.uk
224 Computational Foundry, Bay Campus

Clustering

- To understand its structure, given a cloud of data points.
  - Example: digits in a certain attribute/feature space

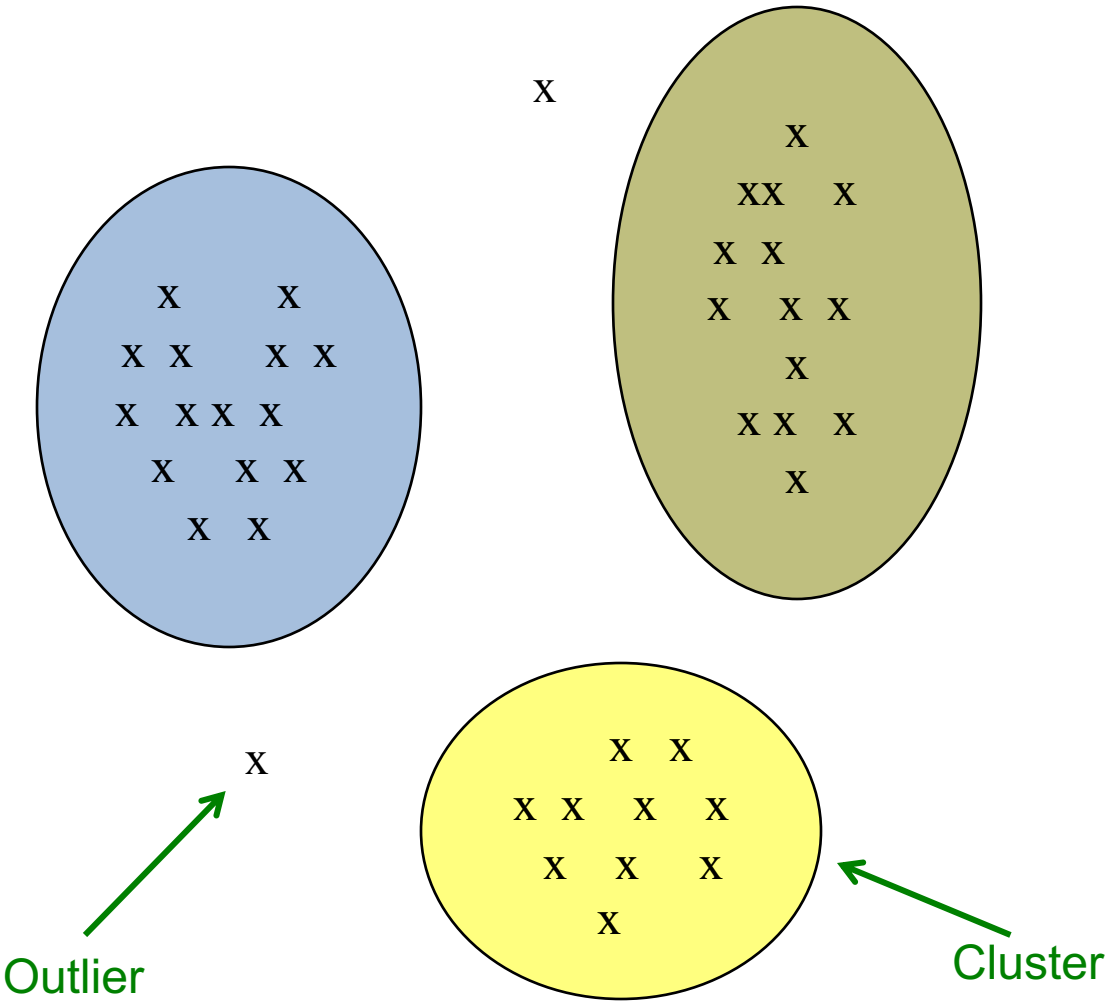
Clustering

- Clustering is a process to find **similarity groups** in data, called **clusters**
  - Group data instances that are similar or near to each other in one cluster
  - Data instances that are (very) different or far away from each other should be in different clusters
  - Clusters are unlabelled and **no *a priori* grouping** of the data instances are given
  - Thus, often known as **unsupervised learning**
- Approaches
  - **K-means** (this lecture)
  - Fuzzy C-means
  - **Gaussian Mixture Modelling** (later)
  - … many more



Clustering

- Clusters & Outliers



Clustering

- Clustering in low dimension and with small amounts of data is often relatively straight forward.

- More often than not, clustering needs to be performed in more than a 2-dimensional space, e.g. 10s or even 10,000s.

- Curse of Dimensionality
  - Data samples required grows exponentially with the increase of dimensionality
  - For random generated points, almost all pair of points are equally far away from one another (average: 1/3 in a unit cube).

Clustering Example Problem 1

- A catalog of 2 billion “sky objects” represents objects by their radiation in 7 frequency bands
  - 7 dimensional space
- Problem: cluster them into similar objects, e.g. galaxies, nearby stars, quasars, and so on
- Sloan Digital Sky Survey

Clustering Example Problem 2

- Intuition: music divides into categories, and customers prefer a few categories
  - But how should the categories be defined?
    - Based on genre? (yes, if you ask a database person)
    - How to carry out more effective recommendation to customers?

- Data driven approach
  - Represent a song by a set of customers
  - Likely, similar songs have similar sets of customers and vice-versa
  - Put all available songs into a space defined by customers
  - Clusters in this space is arguably a better representation of music category than pre-defined genre

Clustering Example Problem 2

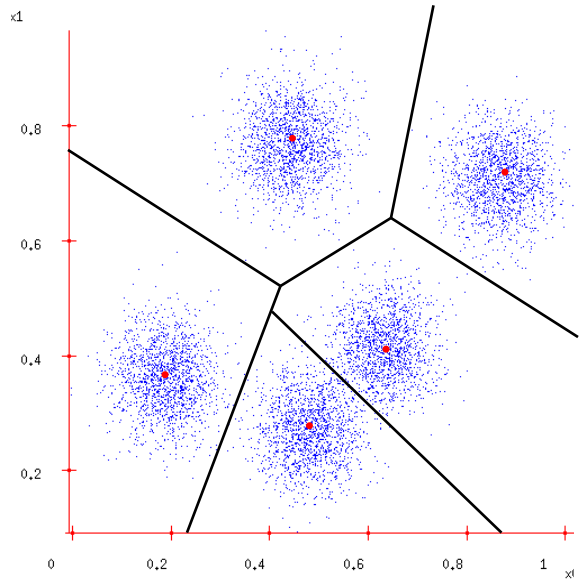
- Space of “all” songs
  - A space with one dimension for each customer
  - Values in a dimension may be 0 or 1 only, i.e. binary
  - A song thus is a point in this space (x_1, x_2,…, x_k), where x_i = 1 iff the i^{th} customer bought the song
    - “iff” read as “if and only if”

- Task: find clusters of similar songs

- For Amazon and Apple, the dimension is tens of millions

K-means Clustering

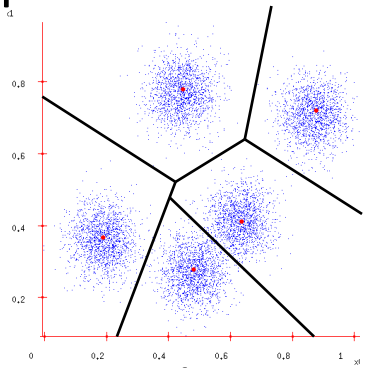
- K-means is a partitional clustering algorithm
- The number of clusters is often pre-set (value for K)
- Each cluster is represented by the centre of the cluster
- Each data point is assigned to the nearest cluster centre or centroid
- It is an iterative process, start with a random initialisation of the centroids



K-means Clustering

- How to determine it is a good clustering (according to K-means)?
  - Minimise the **Sum of Squared Error** (SSE) from data points to their corresponding centroids

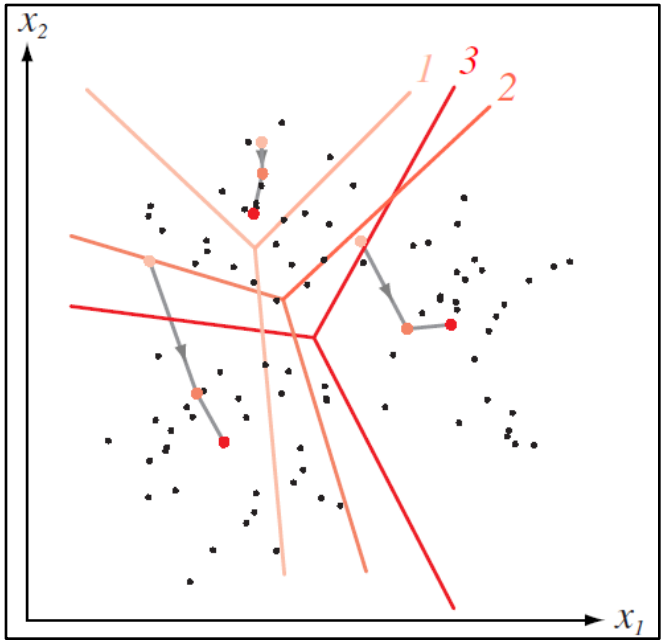
$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2$$



  - C_j denotes the jth cluster, $\mathbf{m}_j$ is the centroid of cluster C_j, and $dist(\mathbf{x}, \mathbf{m}_j)$ denotes the distance between data point **x** and its centroid.
  - Hence, the stopping criteria for the iterative estimation of the centroids is often based on the change in SSE
    - Very small changes in SSE indicates convergence
    - Sometimes, fixed number of iterations is used

K-means Clustering

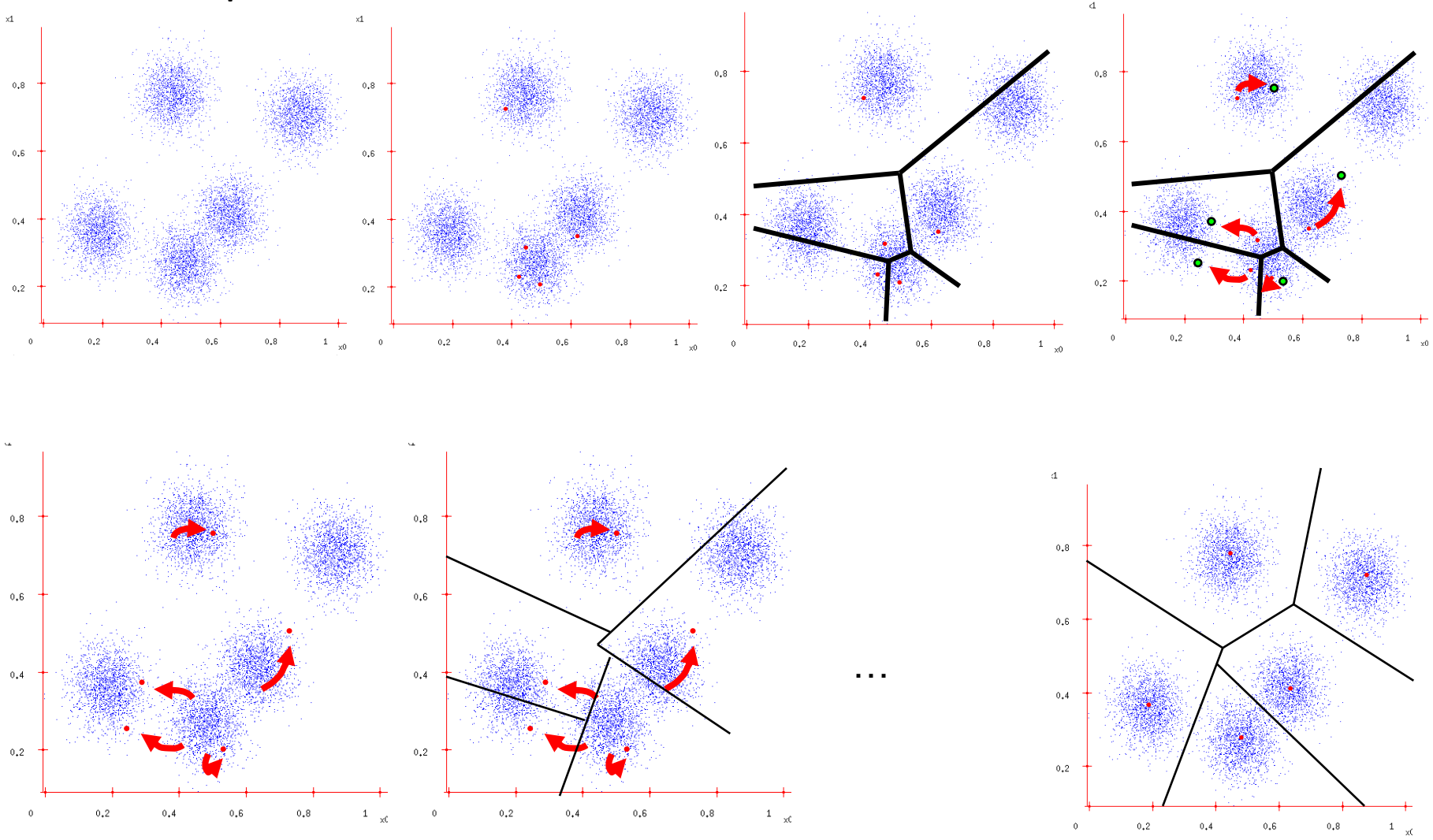
- Algorithmic steps
  - Determine the value for K (number of clusters)
  - Randomly choose initial K centroids
  - Repeat:
    - Assign each data point to the nearest centroid
    - Update the centroids based on data partitioning
  - Until the stopping criterion is met



An online example:
http://www.math.le.ac.uk/people/ag153/homepage/KmeansKmedoids/Kmeans_Kmedoids.html

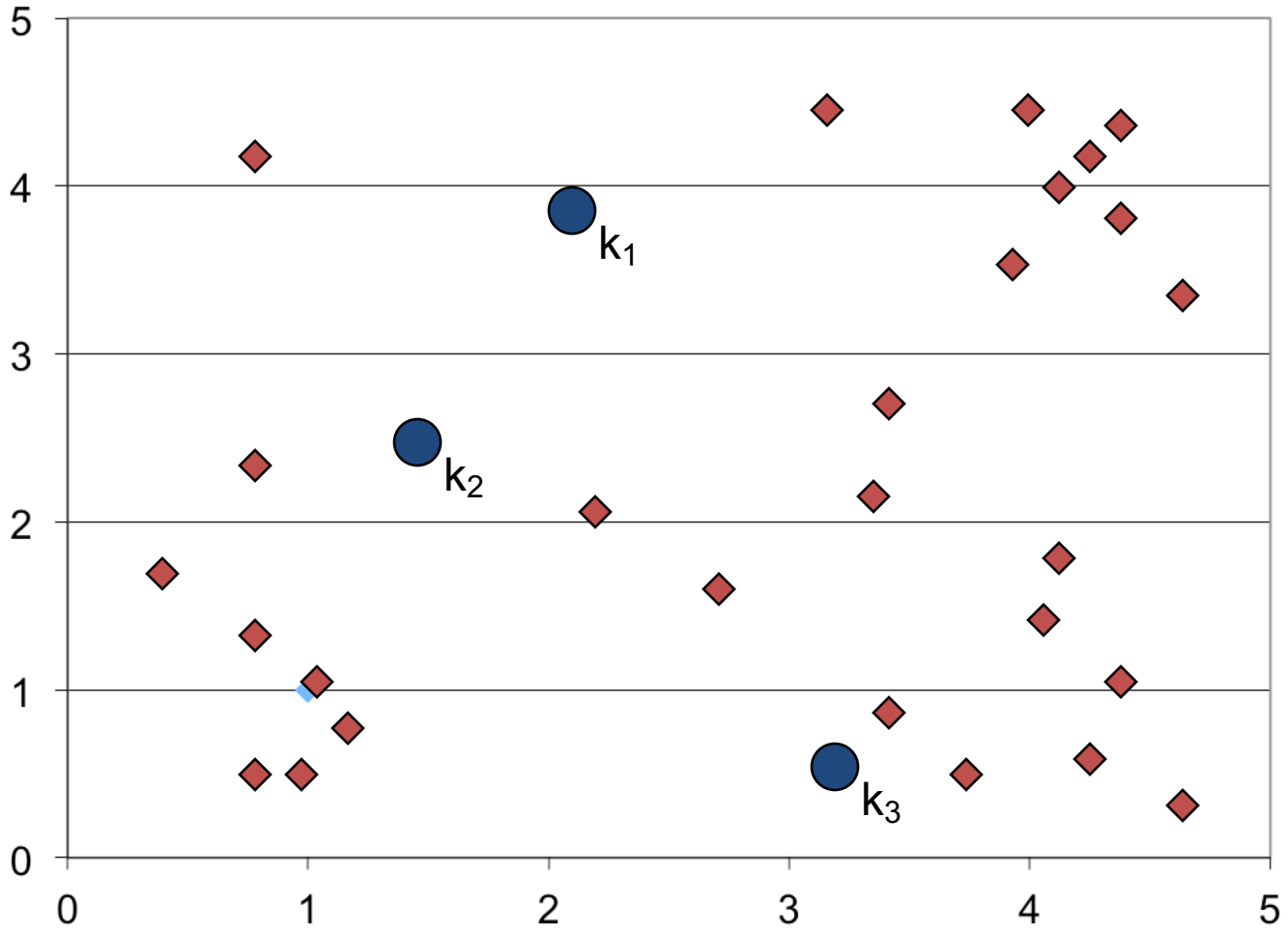
K-means Clustering

- Example



K-means Clustering

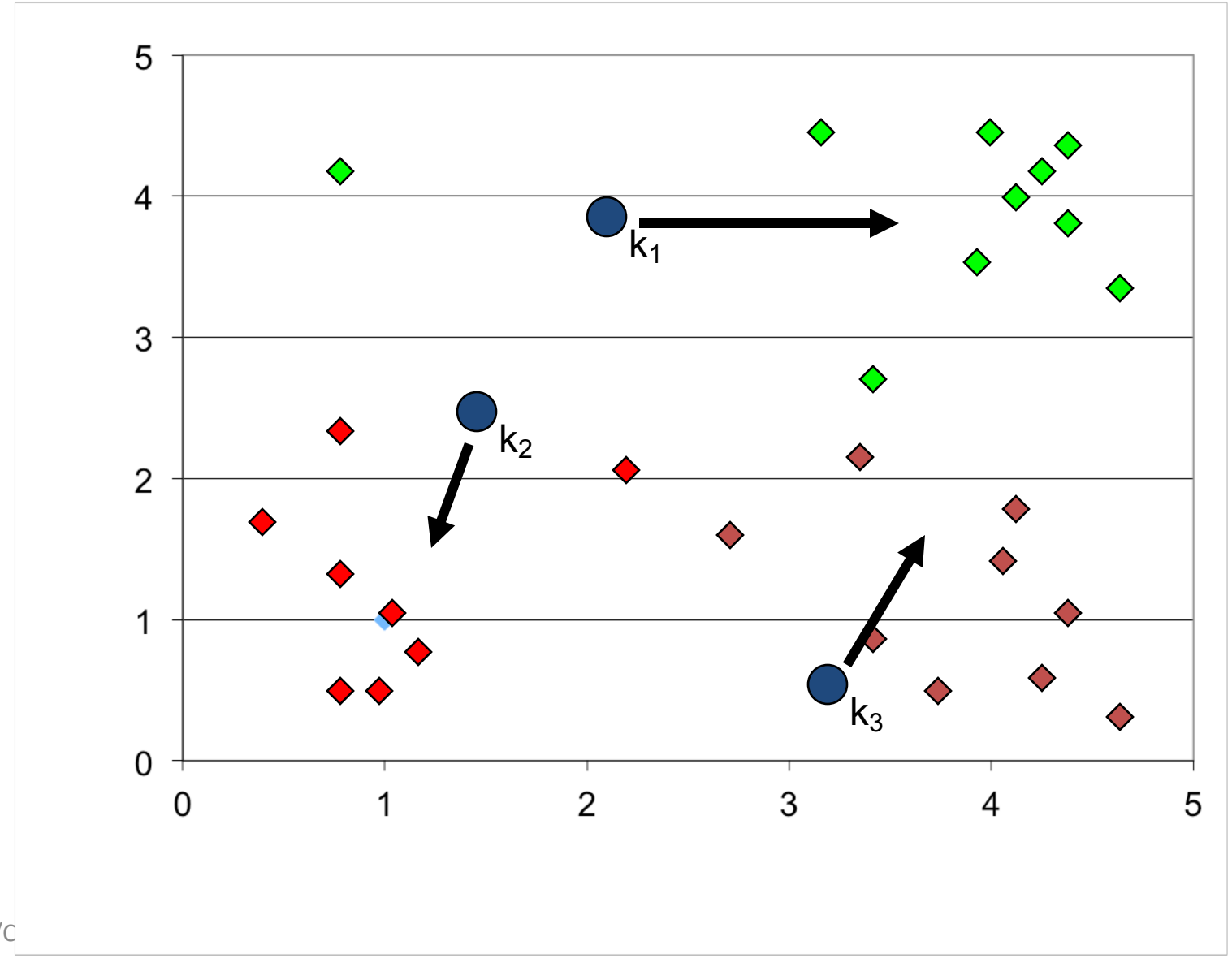
- Example 2: step 1, random initialisation of centroids



Credit: WashUni BIO5488

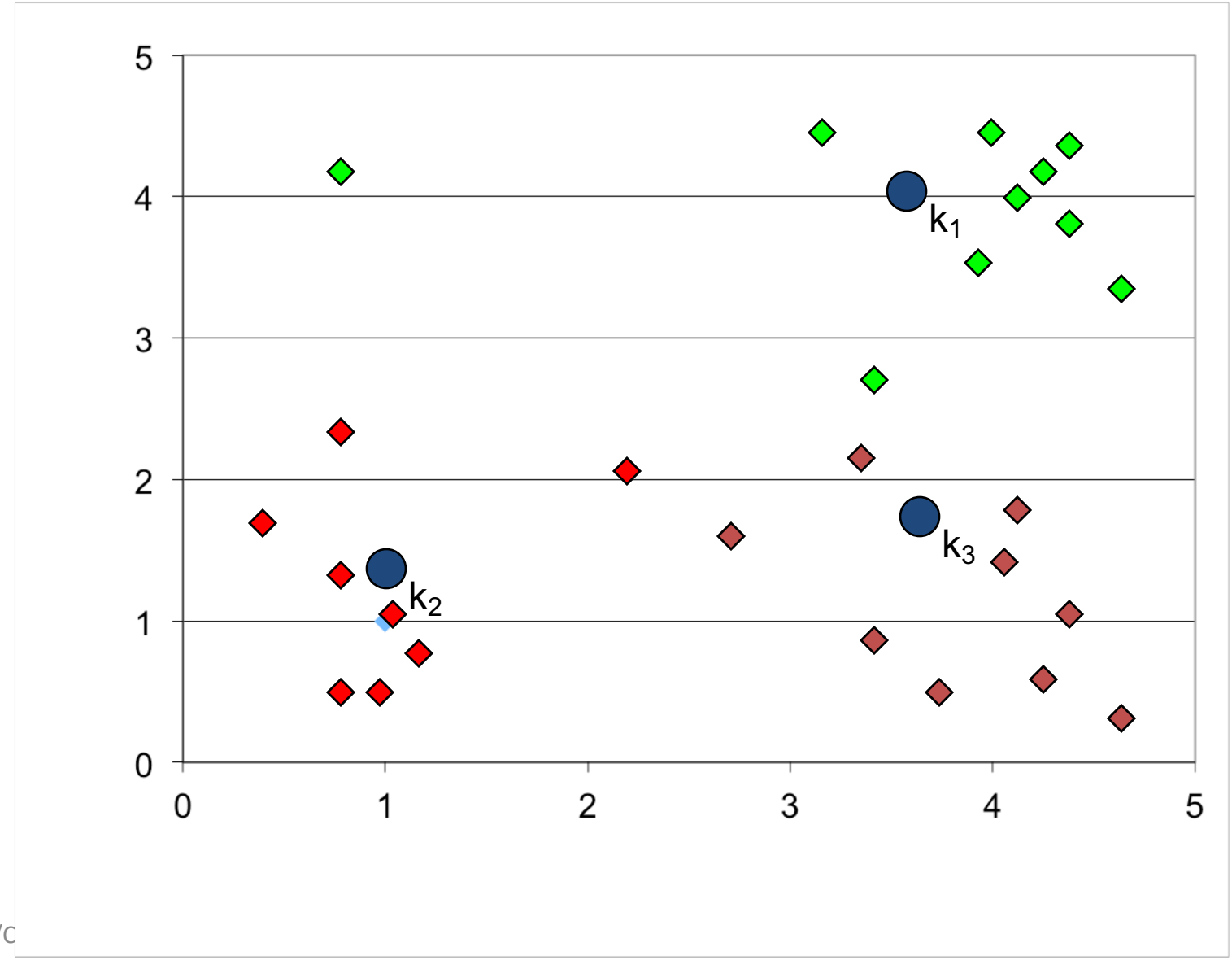
K-means Clustering

- Step 2, assign each data to nearest centroid



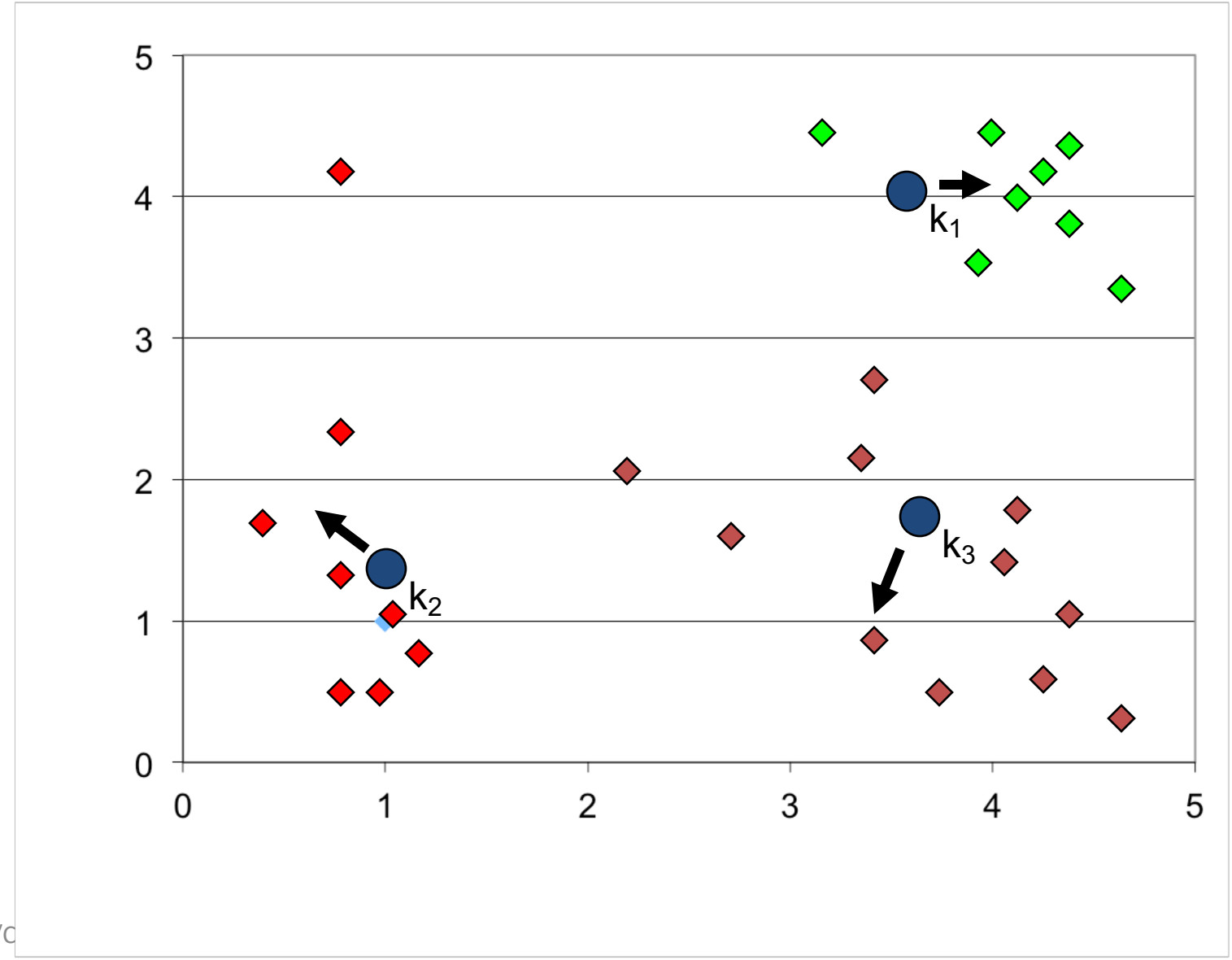
K-means Clustering

- Step 3, recalculate centroids



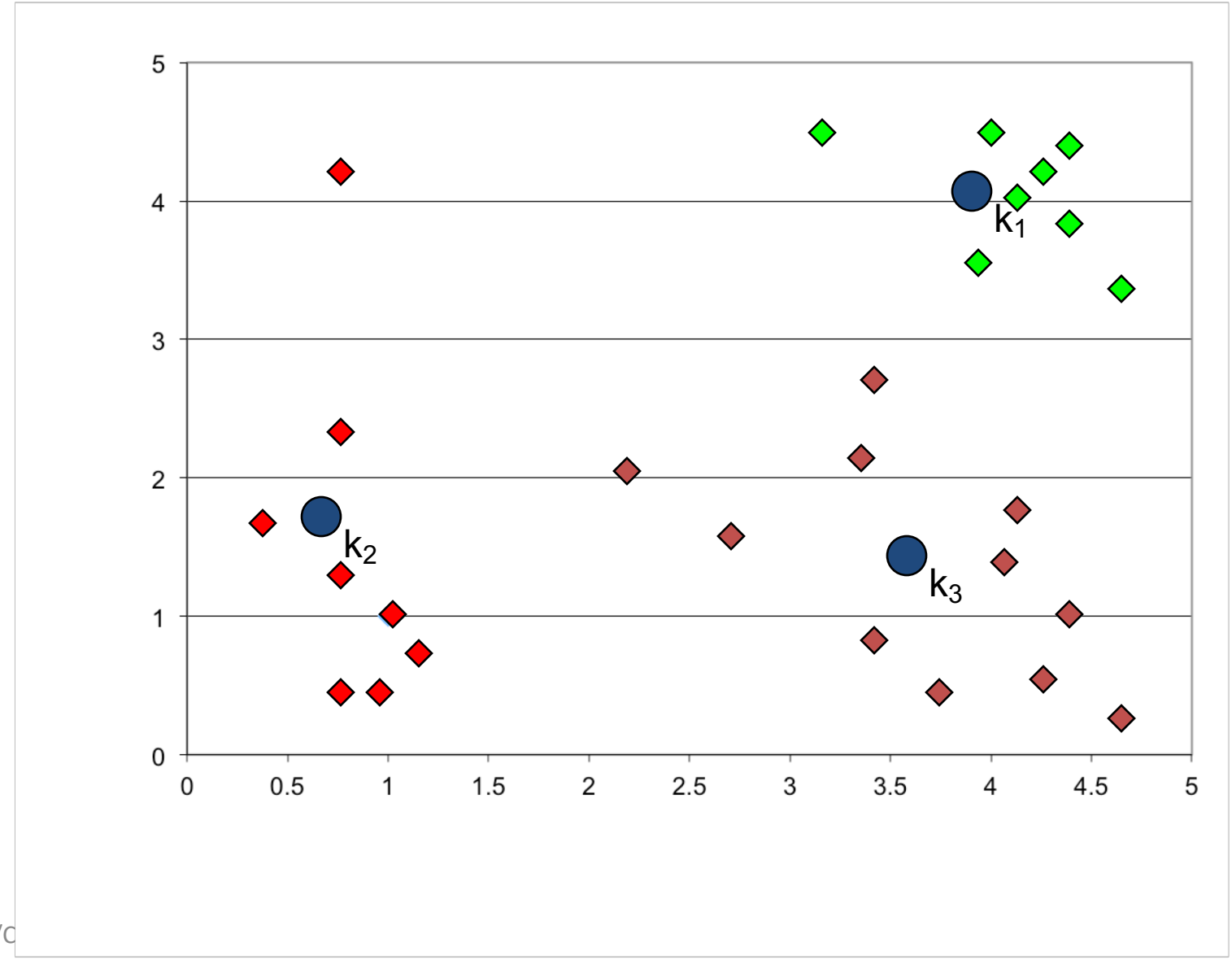
K-means Clustering

- Repeat steps 2 and 3



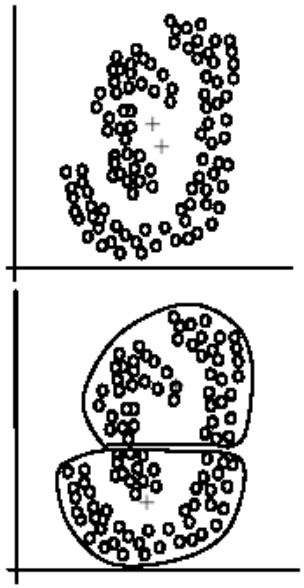
K-means Clustering

- Until converges



K-means Clustering

- Generally fast (although an iterative process)
- Still one of the most popular clustering algorithms
  - Fuzzified version often is more robust
- Have to know the number of clusters to start with
  - Not always an easy task
- Provides a local solution
  - Results depends on initialisation
- Sensitive to outliers



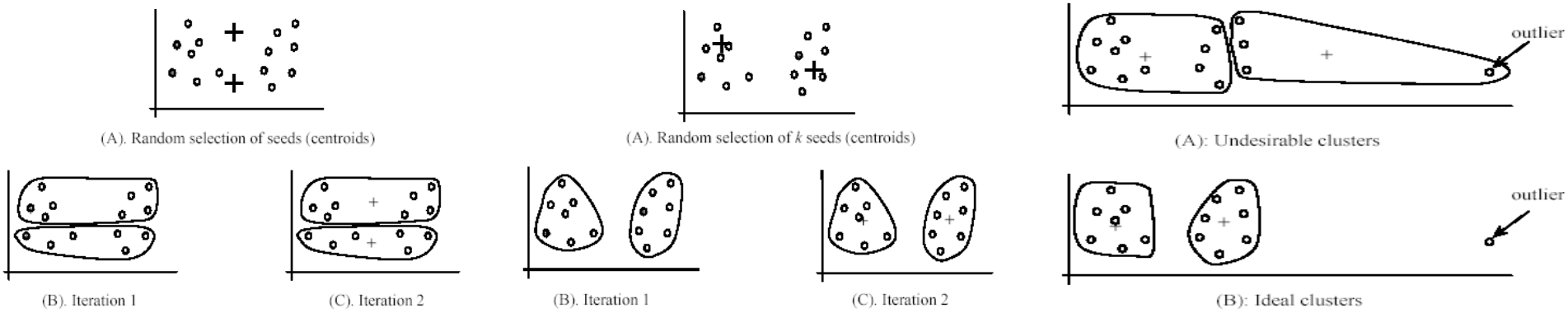


Image credit: Bing Liu, UIC