Data Clustering

Danushka Bollegala

Outline

- Why cluster data?
- Clustering as unsupervised learning
- Clustering algorithms
  - k-mean, k-medoids
  - agglomerative clustering
  - Brown's clustering
  - Spectral clustering
- Cluster evaluation measures
  - Purity, Inverse-Purity
  - Rand Index
  - B-CUBED
  - macro vs. micro averaged precision, recall, F-score
- Supervised clustering

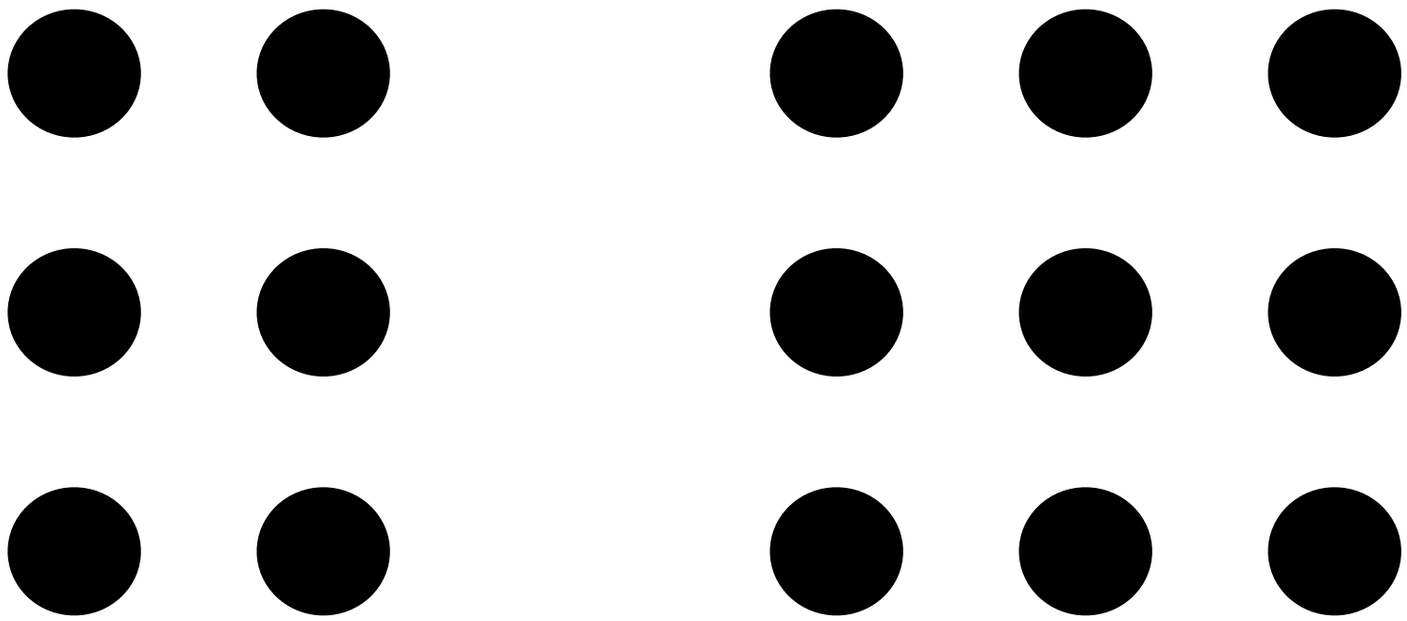
Why cluster data?

- Data Mining has two main objectives
  - Prediction: classification, regression etc.
  - Description: pattern mining, rule extraction, visualisation *clustering*
- Clustering is:
  - Unsupervised learning
    - no label data is required (consider classification algorithms we discussed so far in the lecture which are supervised algorithms)
- Generalisation / Abstraction of concepts
- Topic detection
- Visualisation
- Outlier detection

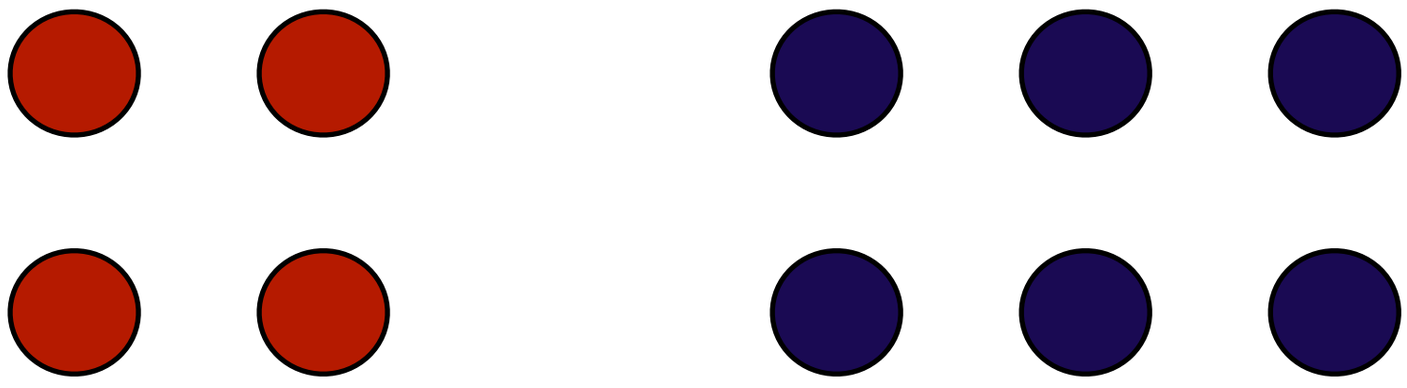
Unsupervised Learning

- Supervised learning
  - labels for training instances are provided
- Unsupervised learning
  - No labels for training instances are provide
- Semi-supervised learning
  - Both labeled and unlabeled training instances are provided
- What can we learn about training data if we do not have any labels?
  - The similarity and distribution of the features can still be learnt and this can be used to create rich feature spaces for supervised learning (if required)

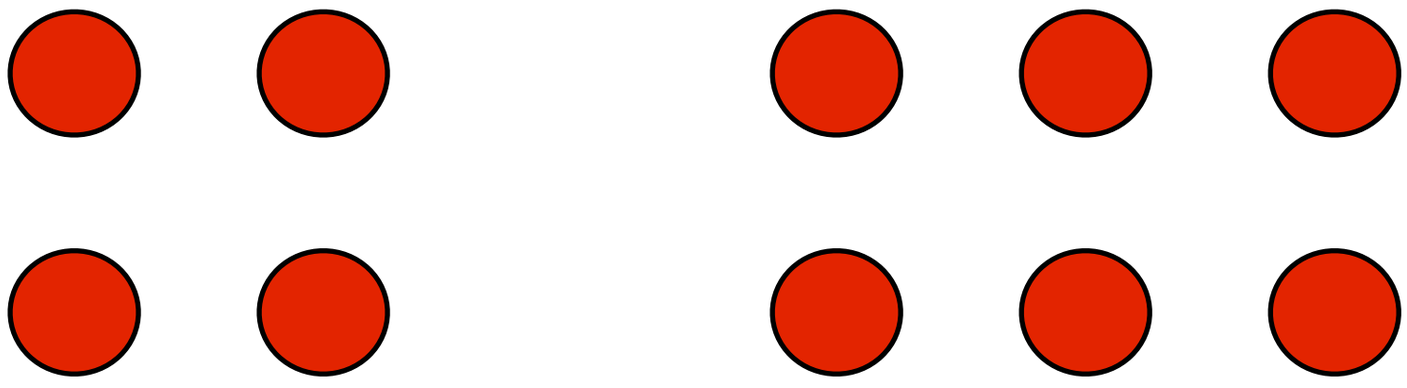
Quiz: Cluster the Following Data



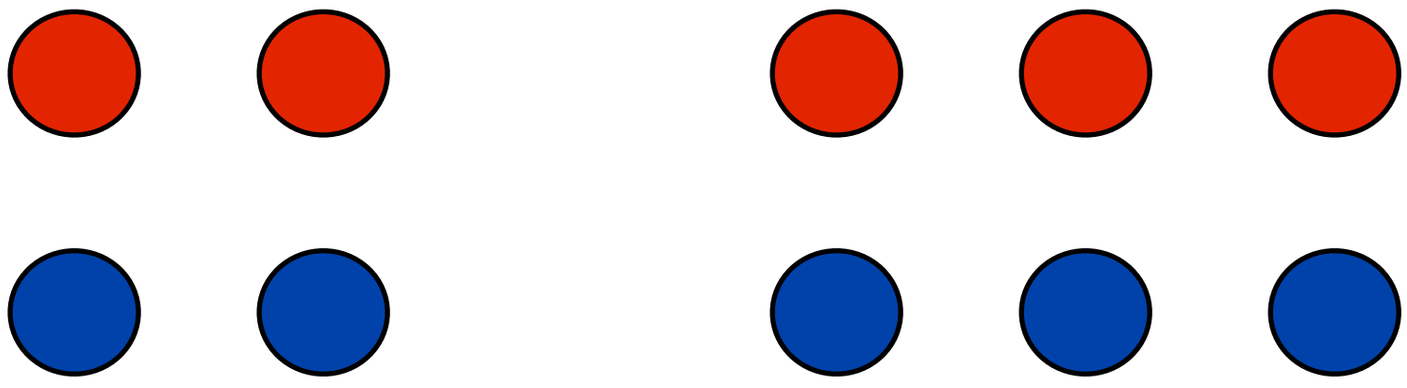
Quiz: Cluster the Following Data



Quiz: Cluster the Following Data



Quiz: Cluster the Following Data



How many clusters?

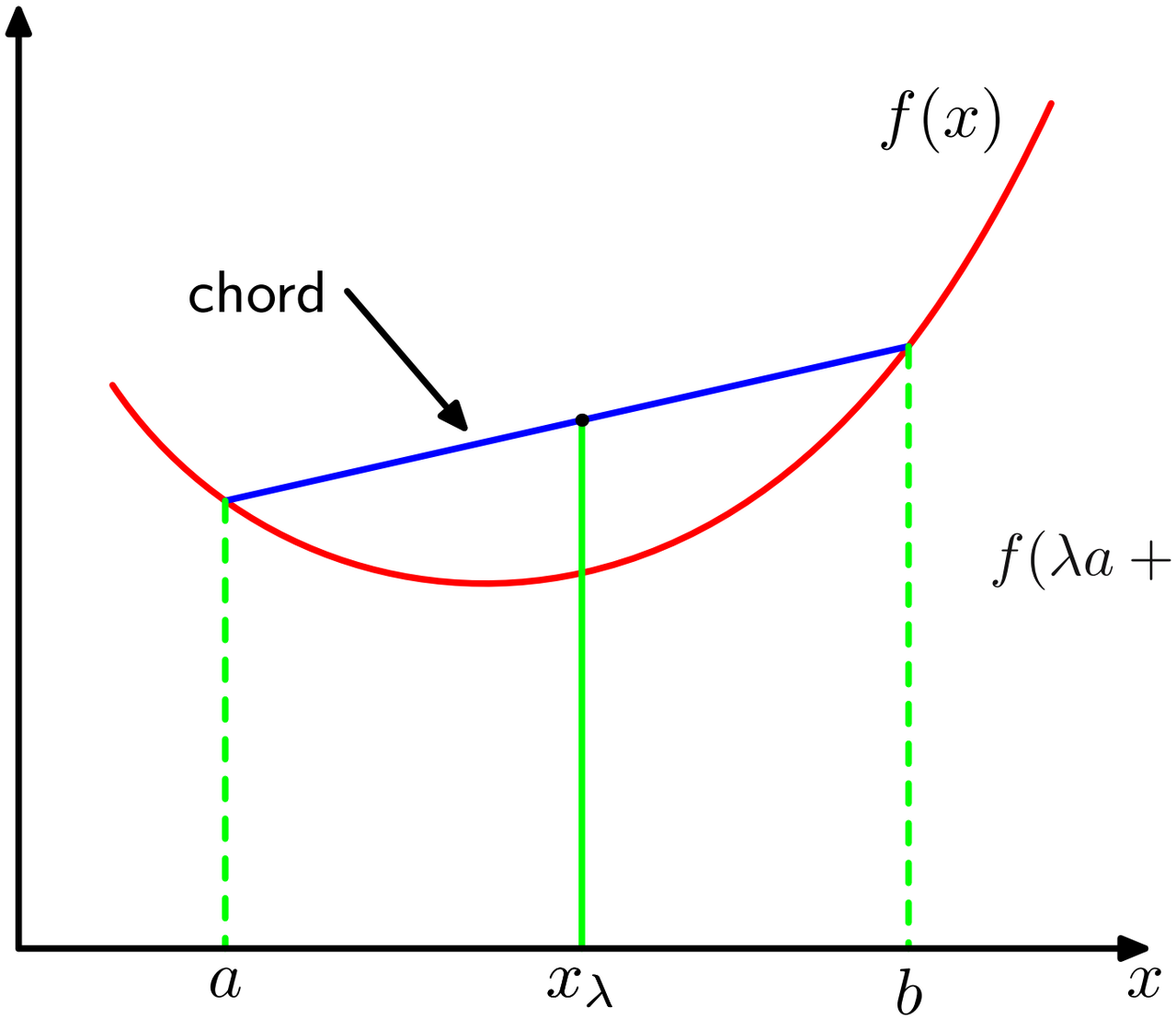
General Remarks

- A single dataset can be clustered into several ways
- There is no single right or wrong clustering
  - Simply different views on the same data
- If then how can we measure the quality of a clustering algorithm?
  - Two ways
    - Compare the clusters produced by a clustering algorithm against some reference (gold standard) set of clusters (direct evaluation)
    - Use the clusters as features for some other (eg. supervised learning) task and measure the difference in the performance of the second task (indirect evaluation)

Clustering as Optimisation

- Given a dataset $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ of N instances represented as *d* dimensional real vectors ($\mathbf{x}_i \in R^d$), partition these N instances into k clusters $S_1,...,S_k$ such that some objective function $f(S_1,...,S_k)$ is minimised
- Observations
  - k and *f* are given
  - *f* can be the similarity between the clusters (good to create dissimilar clusters as much as possible), information gain, correlation and various other such *goodness* measures (heuristics)
  - Often clustering is an NP hard and a non-convex problem
    - http://rangevoting.org/VattaniKmeansNPC.pdf
    - approximations, relaxations are required in practice

Convex Functions



$$f(\lambda a + (1-\lambda)b) \leqslant \lambda f(a) + (1-\lambda) f(b).$$

Clustering Algorithms

- Partitioning
  - Construct k partitions and iteratively update the partitions
    - k-Means, k-Medoids
- Hierarchical
  - Create a hierarchy of clusters (dendrogram)
    - Agglomerative clustering (bottom-up)
    - Conglomerative clustering (top-down)
- Graph-based clustering
  - Graph-cut algorithms (Spectral Clustering)
- Model-based clustering
  - Mixture of Gaussians
- Other types: Non-parametric Bayesian (Latent Dirichlet Allocation), Expectation Maximisation (EM) algorithm, and many more ...

k-Means Derivation

$$\arg \min_{S_1,\ldots,S_k} \sum_{i=1}^{k} \sum_{\boldsymbol{x}_j \in S_i} ||\boldsymbol{x}_j - \boldsymbol{\mu}_i||^2$$

We want to minimise the distance between data instances (x_j) and some cluster centres (μ_i)

$$f(S_1, \ldots, S_k) = \sum_{i=1}^{k} \sum_{\boldsymbol{x}_j \in S_i} ||\boldsymbol{x}_j - \boldsymbol{\mu}_i||^2$$

This objective function is called the *within cluster sum of squares* (WCSS) objective

$$\frac{\partial f(S_1, \ldots, S_k)}{\partial \mu_i} = 0$$

$$\frac{\partial f(S_1, \ldots, S_k)}{\partial \mu_i} = \sum_{\boldsymbol{x}_j \in S_i} 2(\boldsymbol{x}_j - \boldsymbol{\mu}_i)$$

$$\mu_i = \frac{1}{|S_i|} \sum_{\boldsymbol{x}_j \in S_i} \boldsymbol{x}_j$$

Just compute the centroid (mean) of each cluster and that will give you the cluster centres

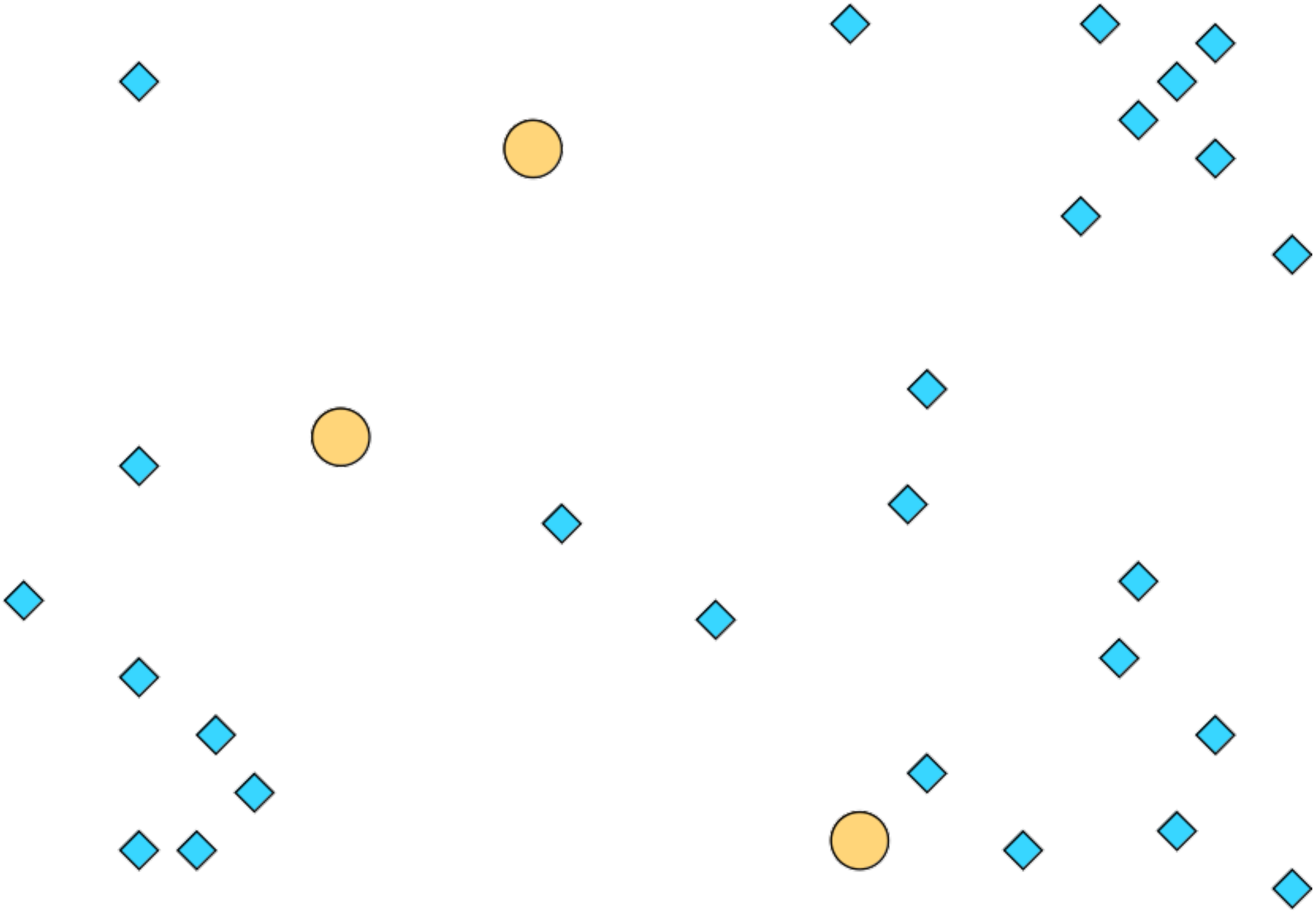
k-Means Clustering

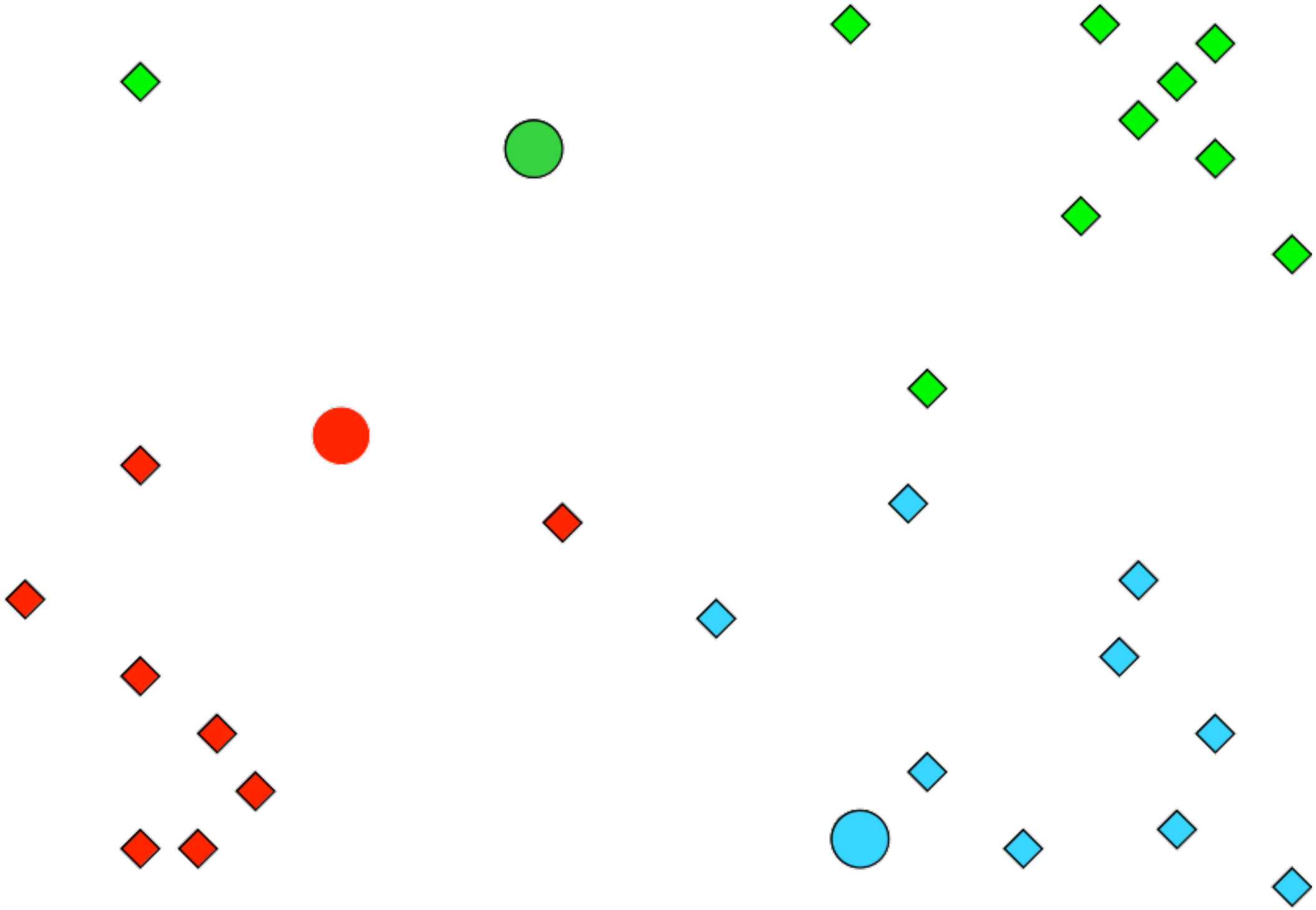
- INPUT
  - The number of clusters *k*
  - Dataset $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ of N instances represented as *d* dimensional real vectors ($\mathbf{x}_i \in R^d$)

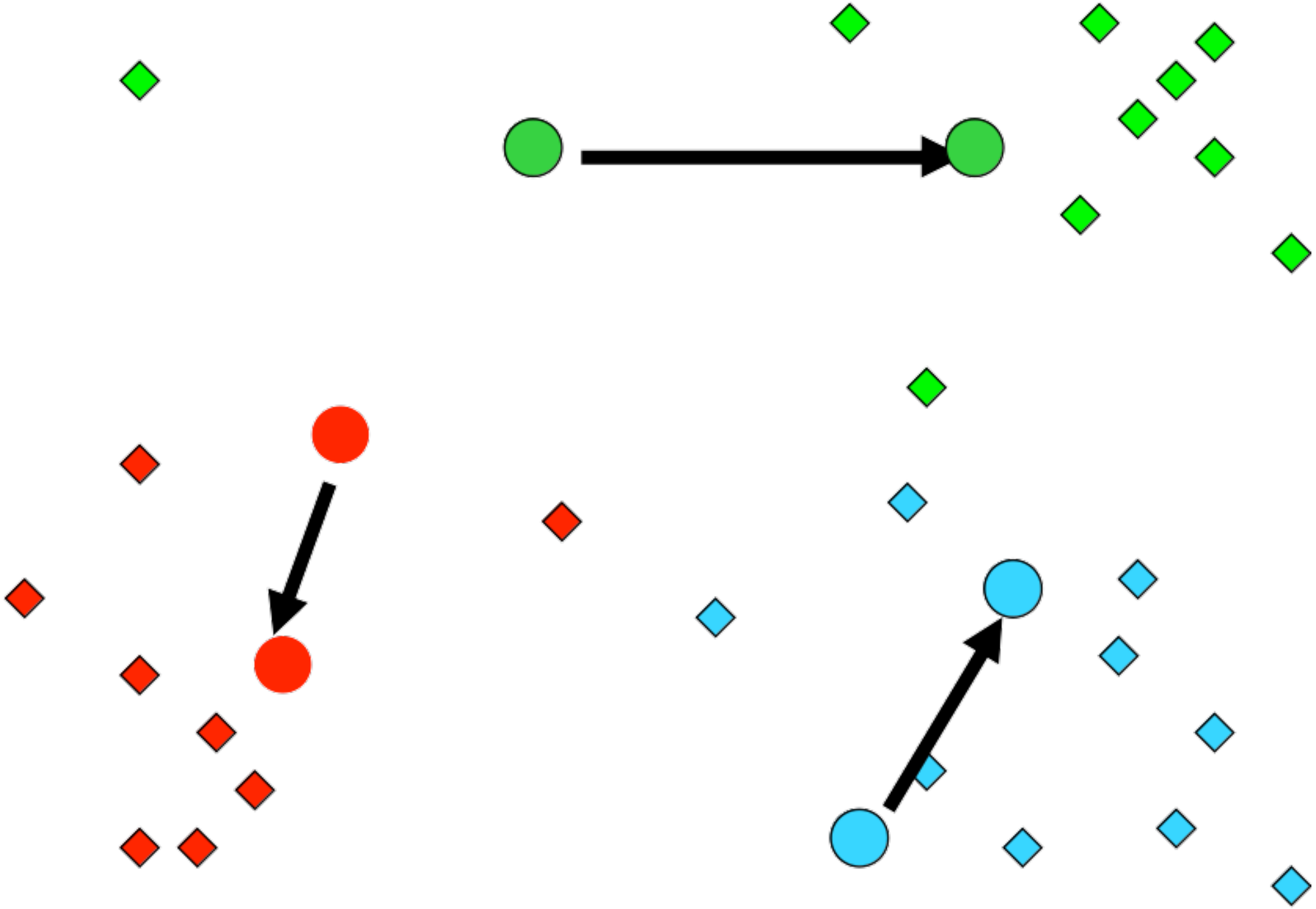
1. Set k instances from the dataset randomly. (initial cluster means/ centres)
2. Assign all other instances to the closest cluster centre.
3. Compute the mean of each cluster
4. until convergence repeat between steps 2 and 3

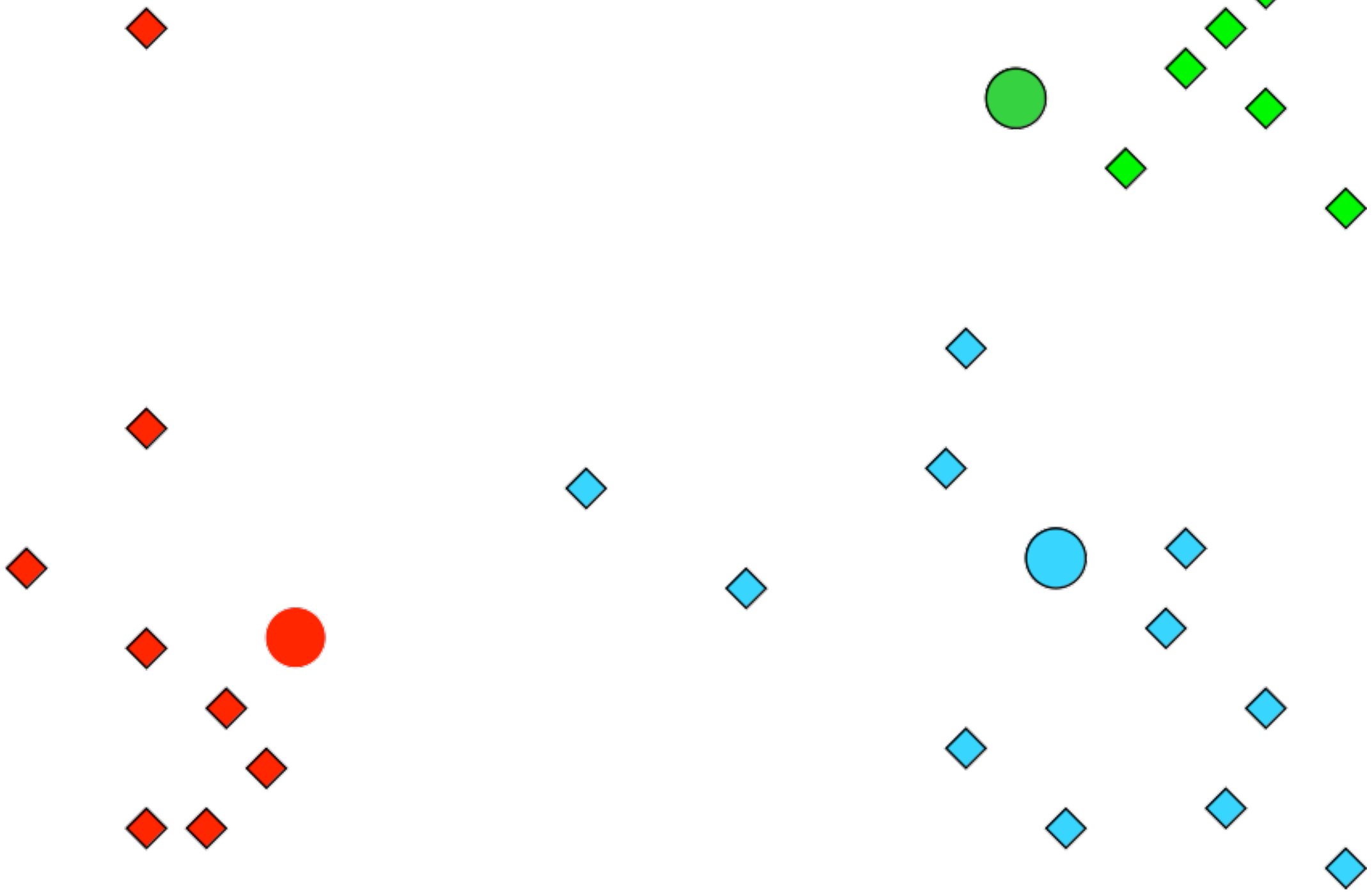
convergence = no instances have moved among clusters

(often after a fixed number of iterations specified by the user)







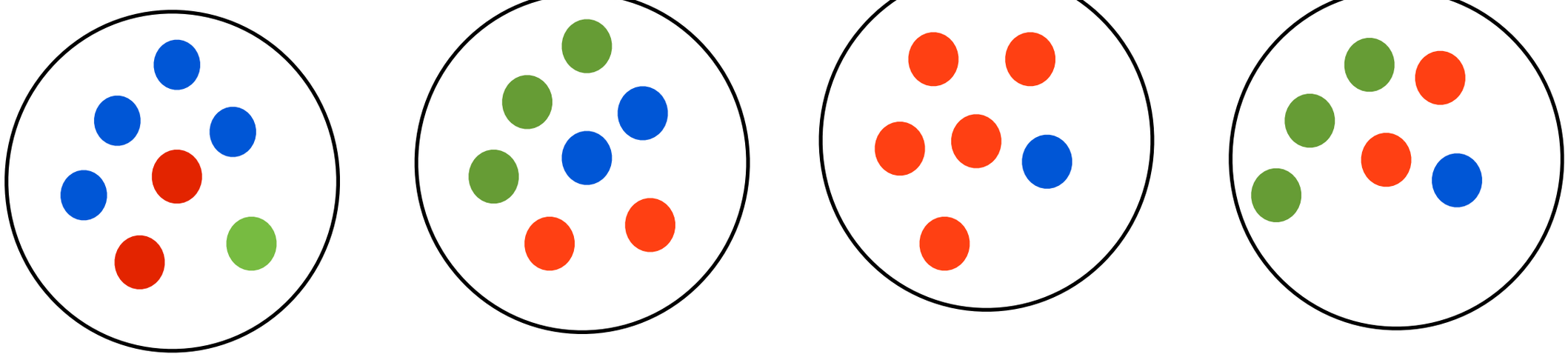


Issues with k-Means

- Results can vary depending on initial random choices
- Can get trapped in a local minimum that isn't the global optimal solution
  - Repeat the clustering procedure multiple times with different initialisations and select the *best* final clustering
    - *best*? according to what? many heuristics exist.
      - smallest number of iterations before convergence
      - largest total distance between the final cluster means
- Outliers have a larger effect on the mean value, hence cluster centre and the cluster
- cluster centres (means) are not actual instances in the cluster

Evaluating Clustering

- Assign each cluster the label that appears most in that cluster
- Merge clusters with the same label
- Measure Precision, Recall, and F-measure for each label type
- Compute the macro-averages
  - Compute the total of Precision and divide by the total number of label types (classes) to compute macro-averaged Precision
  - Compute macro-averaged Recall and macro-averaged F-score similarly



Quiz: Compute macro-averaged Precision, Recall, and F-score for the three clusters shown above.

B-CUBED Measure

- Proposed in (Bagga B. Baldwin = B^3)
  - A. Bagga and B. Baldwin. Entity-based cross document coreference resolution using the vector space model, In Proc. of 36th COLING-ACL, pages 79--85, 1998.
- We would like to evaluate clustering without labelling any clusters.

$$\text{precision}(x) = \frac{\text{No. of items in C(x) with A(x)}}{\text{No. of items in C(x)}}$$

$$\text{recall}(x) = \frac{\text{No. of items in C(x) with A(x)}}{\text{Total no. of items with A(x)}}$$

C(x): The ID of the cluster that x belongs to

A(x): label of x

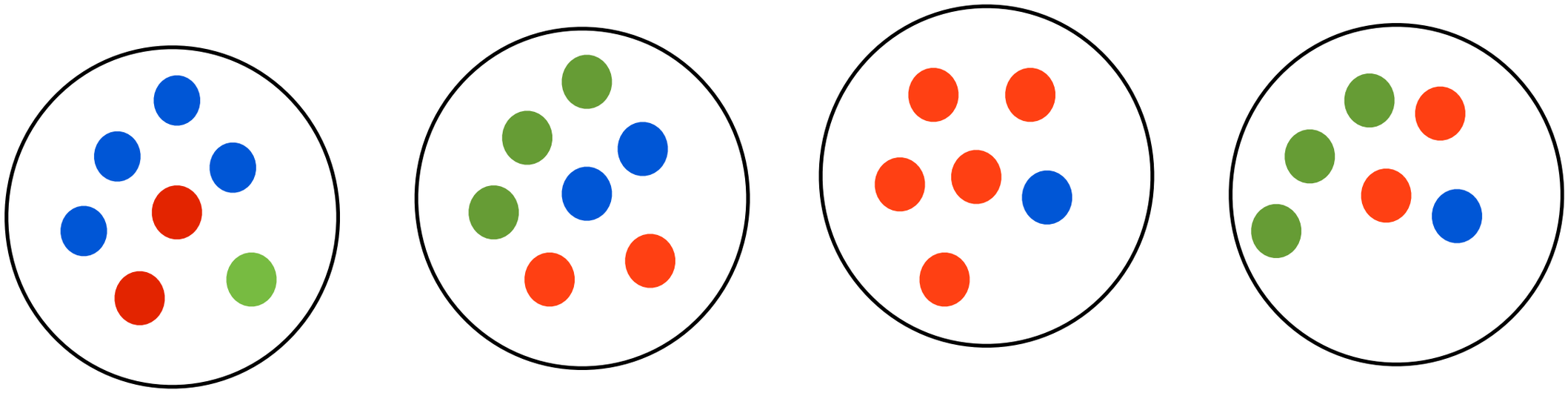
B-CUBED Measure

- Compute the average over all the items (instances) that appear in all clusters (N)

$$\text{Precision} = \frac{1}{N} \sum_{p \in DataSet} \text{Precision}(p)$$

$$\text{Recall} = \frac{1}{N} \sum_{p \in DataSet} \text{Recall}(p)$$

$$F\text{-Score} = \frac{1}{N} \sum_{p \in DataSet} \text{F}(p)$$



Quiz: Compute B-CUBED Precision, Recall, and F-score for the three clusters shown above.

Hierarchical Clustering

- Sometimes we might want to organise the data into a hierarchy of subsuming concepts for visualisation (abstraction) purposes
- Two methods exists
  - Conglomerative clustering
    - Start from one big cluster with all data instances and repeatedly partition it
    - Top-down approach
- Agglomerative clustering
  - Start singletons (clusters with exactly one instance) and iteratively merge the most *similar* two clusters
    - Bottom-up approach
    - computationally more efficient (O(log*n*) merges required)

Merging two clusters

- Single linkage
  - Distance between two clusters A and B is the smallest distance between any instance a ∈ A and b ∈ B

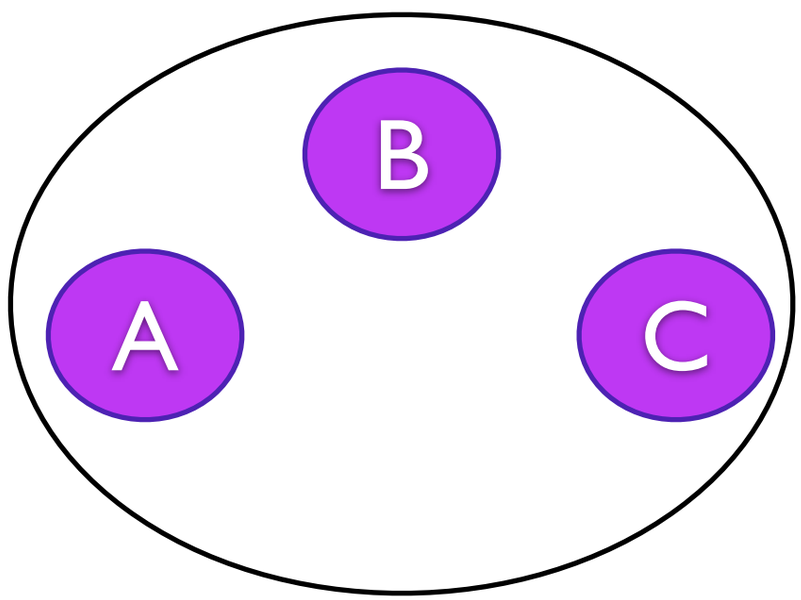
$$D(\mathcal{A}, \mathcal{B}) = \min_{a \in \mathcal{A}, b \in \mathcal{B}} dist(a, b)$$

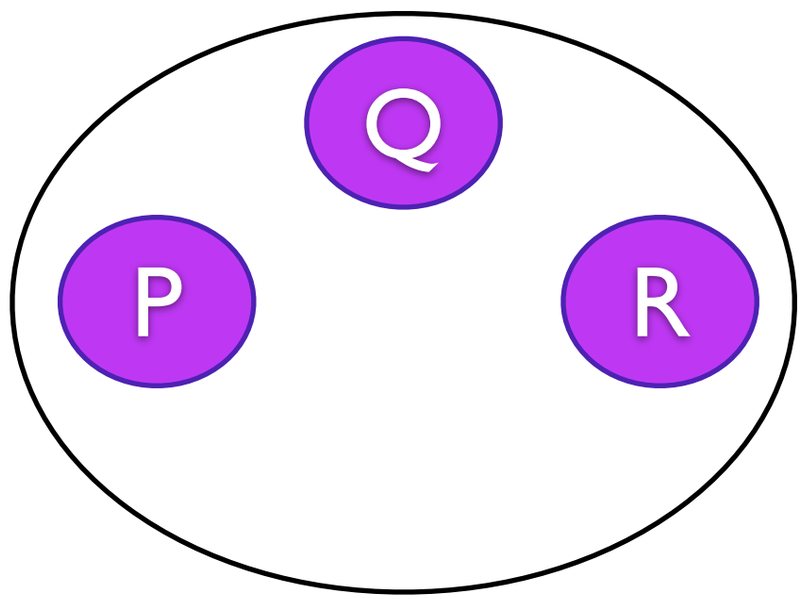
- Complete linkage
  - Distance between two clusters A and B is the largest distance between any instance a ∈ A and b ∈ B

$$D(\mathcal{A}, \mathcal{B}) = \max_{a \in \mathcal{A}, b \in \mathcal{B}} dist(a, b)$$

- Average linkage (Group-Average)
  - Average of all the pairs selected from each cluster

$$D(\mathcal{A}, \mathcal{B}) = \frac{1}{|\mathcal{A}||\mathcal{B}|} \sum_{a \in \mathcal{A}, b \in \mathcal{B}} dist(a, b)$$





Quiz: Let us assume that in the 2D space there are two clusters {A,B,C} and {P,Q,R}.

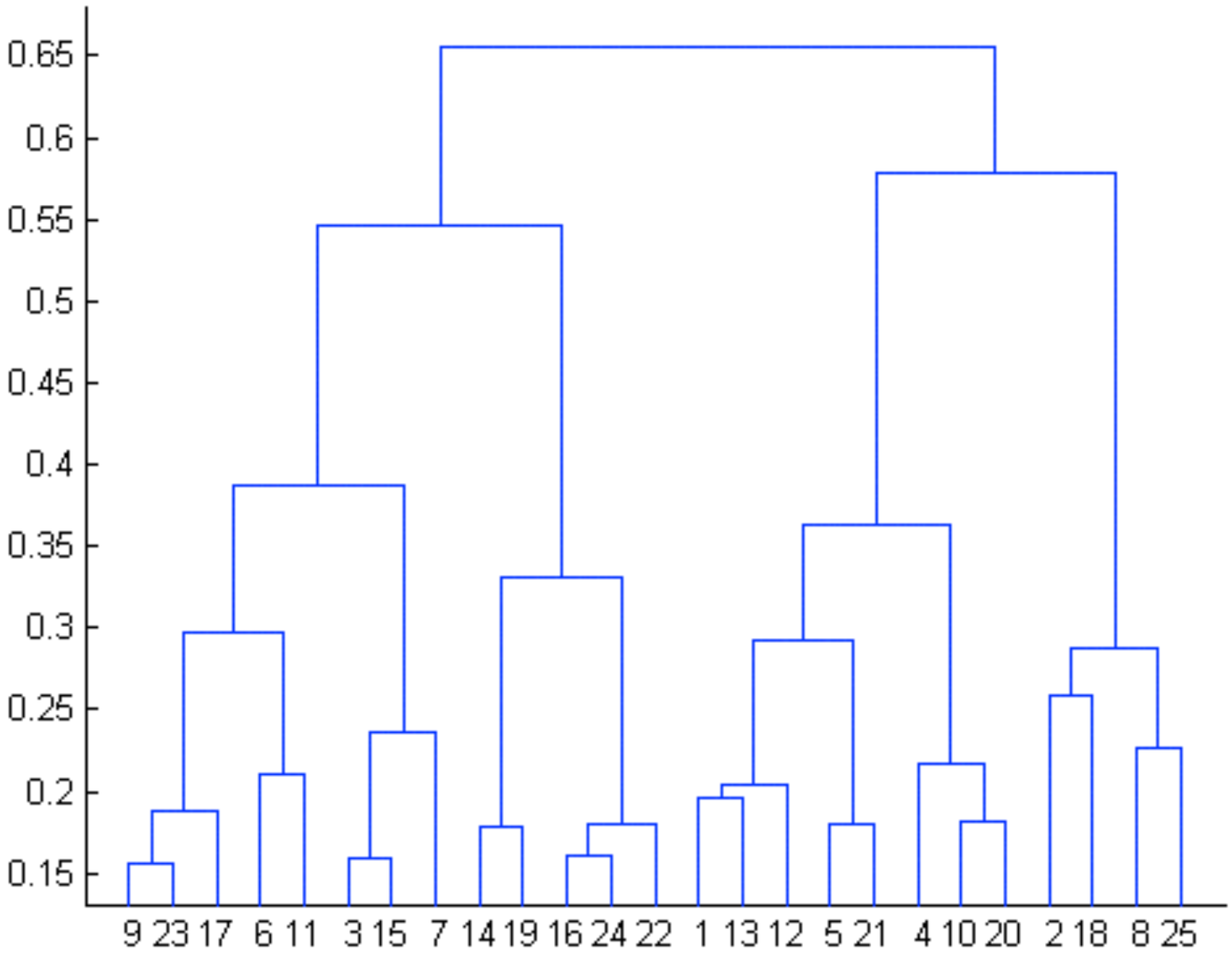
Which of the distances correspond to the single link and complete link distances between the shown clusters?

Group-Average Agglomerative Clustering

- INPUT:
  - A set of N data instances $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$, Number of clusters *k*
- Initialise
  - Create singletons $S_i = \{\mathbf{x}_i\}$ for i = 1, ..., N
- Repeat until only we are left with one cluster
  - Merge the two clusters S_i and S_j with the minimum distance (cf. maximum similarity)

$$D(\mathcal{S}_i, \mathcal{S}_j) = \frac{1}{|\mathcal{S}_i||\mathcal{S}_j|} \sum_{a \in \mathcal{S}_i, b \in \mathcal{S}_j} dist(a, b)$$

Dendrogram


0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
9 23 17 6 11 3 15 7 14 19 16 24 22 1 13 12 5 21 4 10 20 2 18 8 25

Clusters as Features

- We can use clustering to find *similar* features in instances without requiring any supervision (no label data is required for clustering)
  - Distributional similarity of features over instances
- Once we have clustered the features, we can use the cluster IDs as features
- Benefits
  - Reduces the dimensionality of the feature space
    - *dogs* and *cats* are mapped to *pets*
  - Reduces feature sparseness
    - If at least one of the features in a cluster appears in an instance, then we assume that the entire cluster appeared as a feature for that instance

Clustering the word-document Matrix

	d_1	d_2	d_3	d_4	d_5	d_6
dog	2	3	0	0	1	5
cat	1	1	0	0	3	2
apple	0	0	1	2	3	0
banana	0	0	2	5	0	0

Clustering the 2D matrix

- Each row vector can be seen as the feature vector for each word (row elements)
  - We can measure similarity between row vectors and cluster the words
- Each column vector can be seen as the feature vector for each document (column elements)
  - We can measure similarity between column vectors and cluster the documents

Co-clustering

- Cluster both rows and columns simultaneously!
  - You will get both row and column clusters
- Information Theoretic Co-Clustering (ITCC)
  - Inderjit Dhillion and Subramanyam Mallela and Dharmendra Modha, pp. 89--98, International Conference on Knowledge Discovery and Data Mining (KDD), 2003.

Clustering as Graph Partitioning

- Given a set of instances $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$, we can represent these instances using a weighted undirected graph G, where the weight of the edge that connects two vertices in the graph corresponds to the similarity between the corresponding vertices.

- Then, the clustering problem becomes a graph partitioning problem where we must delete *k* edges from this graph to create *k* number of clusters

  - Spectral clustering algorithms (discussed later at the lecture for Graph Mining)