Clustering 2

Manuel R. Ciosici

Agenda

- Determining the number of clusters
- Evaluation
- Hierarchical clustering
- Global vs Local

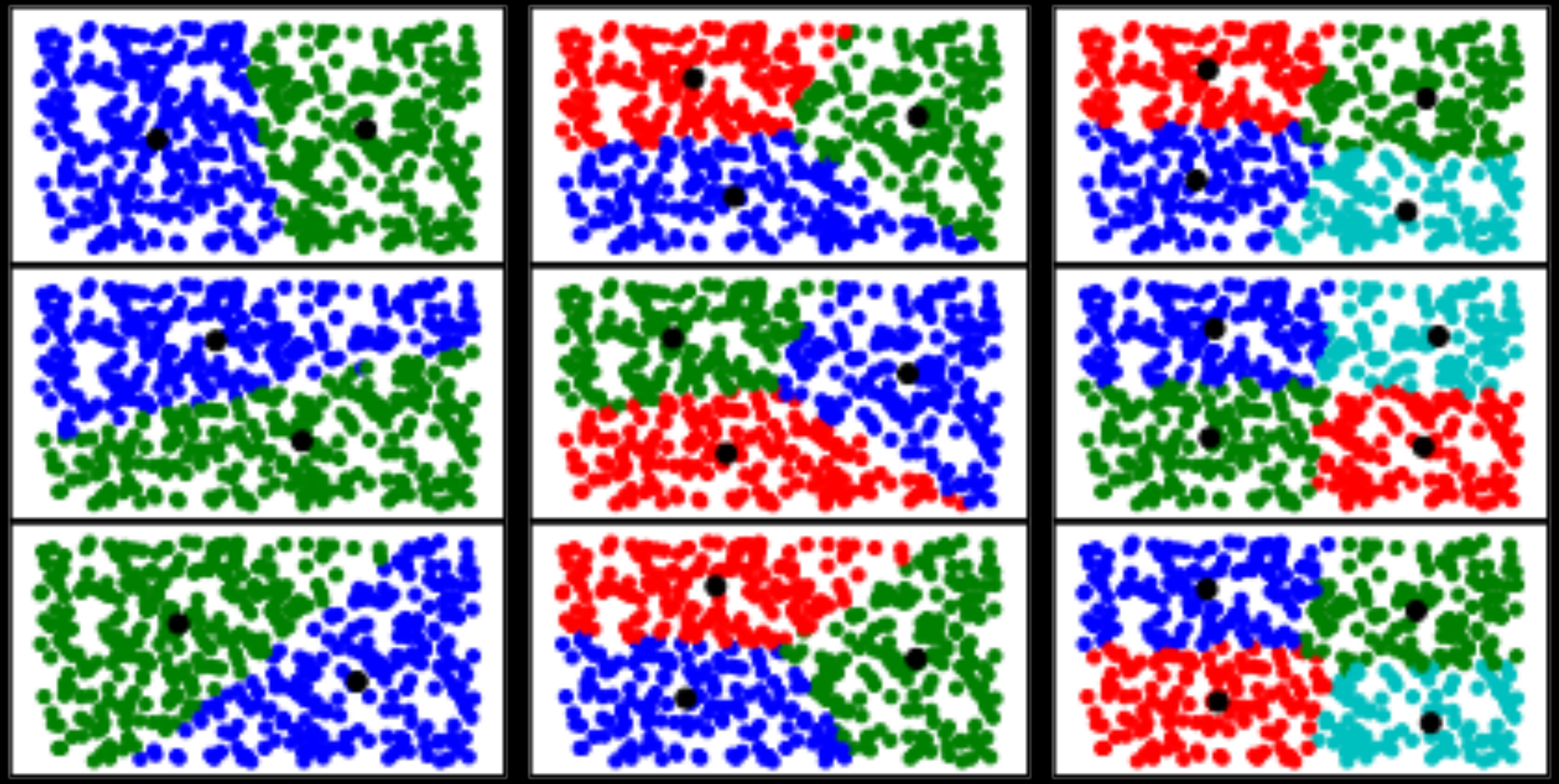
Evaluation of clustering

3 tasks fall under evaluation of clusters:

- Assessing clustering tendency (Are there any clusters in the data?)
- Determining the number of clusters in a data set (How many clusters are there?)
- Measuring clustering quality (What method computes the best clusters?)

Assessing clustering tendency

- We try to determine whether a given data set has a non-random structure.
- Clustering makes no sense if there is no structure in the data



K-means on uniform, randomly distributed data

The Hopkins Statistic

1. Sample n points (p_i) from the data set D uniformly and compute the distance to their nearest neighbor in D;

2. Generate n points (q_i) uniformly distributed in the space of D and compute their distance to their nearest neighbor in D;

3. Compute the Hopkins test:

$$H = \frac{\sum_1^n d(q_i)}{\sum_1^n d(q_i) + \sum_1^n d(p_i)}$$

The Hopkins Statistic

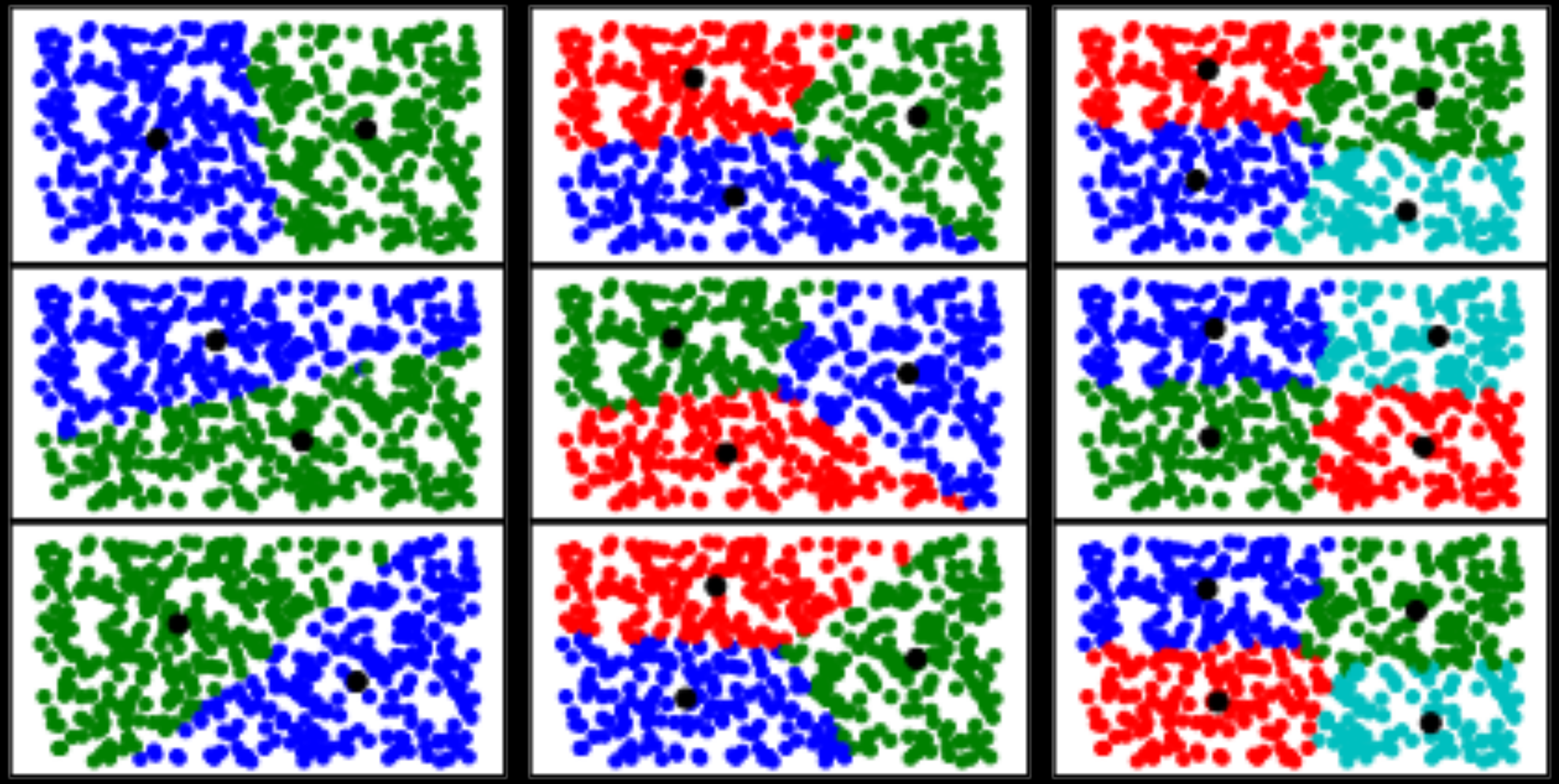
A measure of whether the data is uniformly distributed

If the data is uniformly distributed, then $\sum_{1}^{n} d(q_i)$ and $\sum_{1}^{n} d(p_i)$ should be about the same, so H takes a value of about 0.5

If the data is highly clustered, then $\sum_{1}^{n} d(q_i)$ will on average be larger than $\sum_{1}^{n} d(p_i)$ since the nearest neighbor of each p_i will be within the cluster and thus small, therefore H will take values closer to 1.

The Hopkins Statistic

- The data in this example has a Hopkins Statistic score around 0.6.



K-means on uniform, randomly distributed data

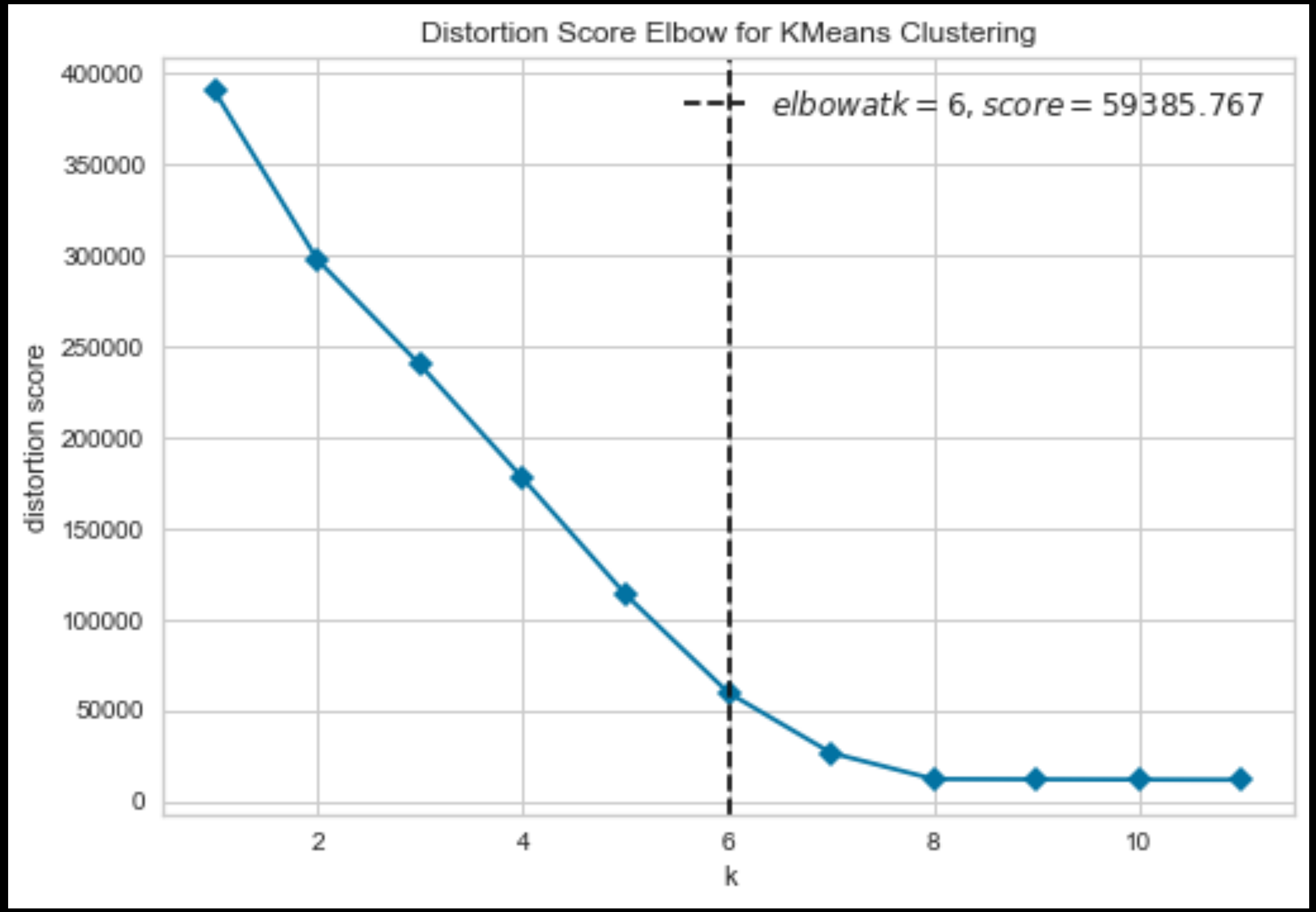
Determining the number of clusters in a data set

- Most clustering algorithms require the number of clusters as input
- How do we select the number of clusters?

Determining the number of clusters in a data set

The elbow method:

- As we increase the number of clusters, we reduce the sum of within-cluster variance of each cluster.
- As the number of clusters increases, the sum decreases less and less → an elbow shaped graph
- Can be used with any intrinsic cluster quality measure, e.g. Silhouette Coefficient (see slide later on)



Elbow graph for k-means. Ideal k for the data is 6 or 7.

Evaluating clustering

- Different cluster algorithms will result in clusters of different quality.
- We want to get high quality clusterings

Quality: What is good clustering?

- A good clustering method will produce high quality clusters with
  - High intra-class similarity
  - Low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

Evaluating clustering

- Different cluster algorithms will result in clusters of different quality.

- We want to understand how different clustering methods compare with each other:

    1. Intrinsic methods – We do not have the ground truth data. We try to determine the quality of the clustering based exclusively on the data

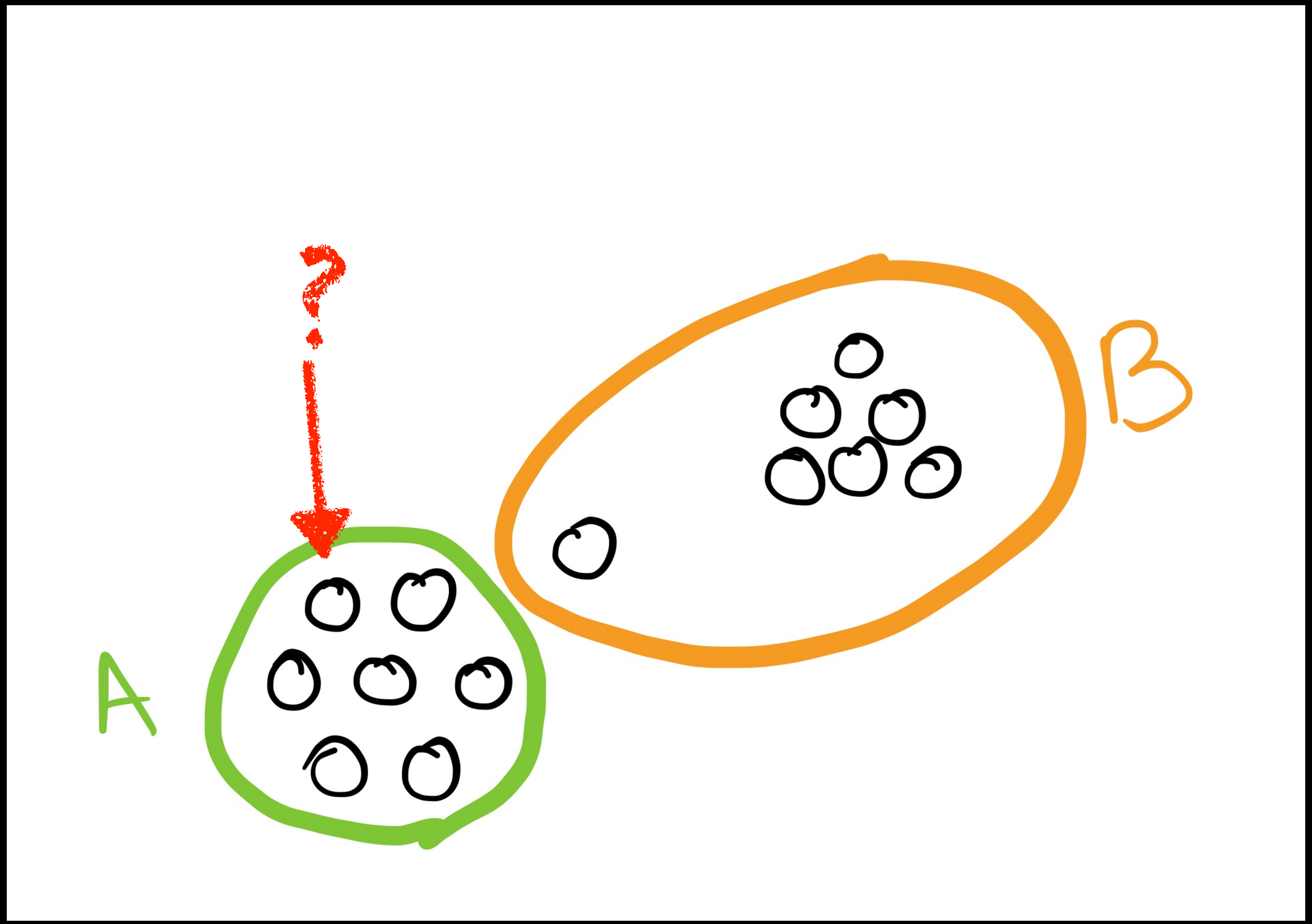
    2. Extrinsic evaluation – We have the ground truth, we compare the clustering with the ground truth

Intrinsic methods

- We do not have access to the ground truth data (the usual case in data mining).
- Intrinsic measures utilize notions of intracluster similarity or compactness, contrasted with notions of intercluster separation.
- Many intrinsic methods take advantage of similarity measures between objects in the data set.

Is this object allocated to the proper cluster?

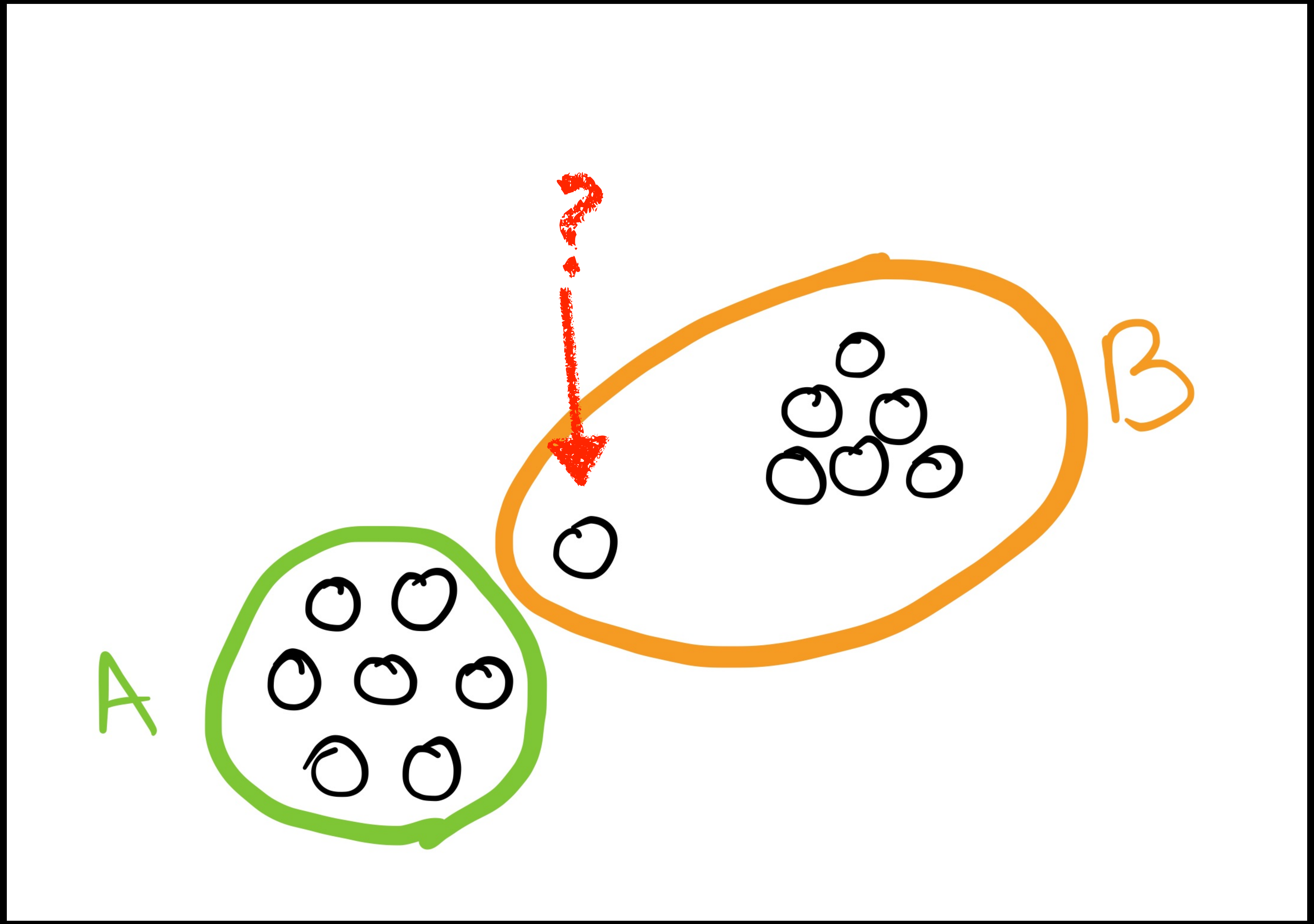
- [✓ Yes]
- No



How about this one?

- Yes
- [✓ No]

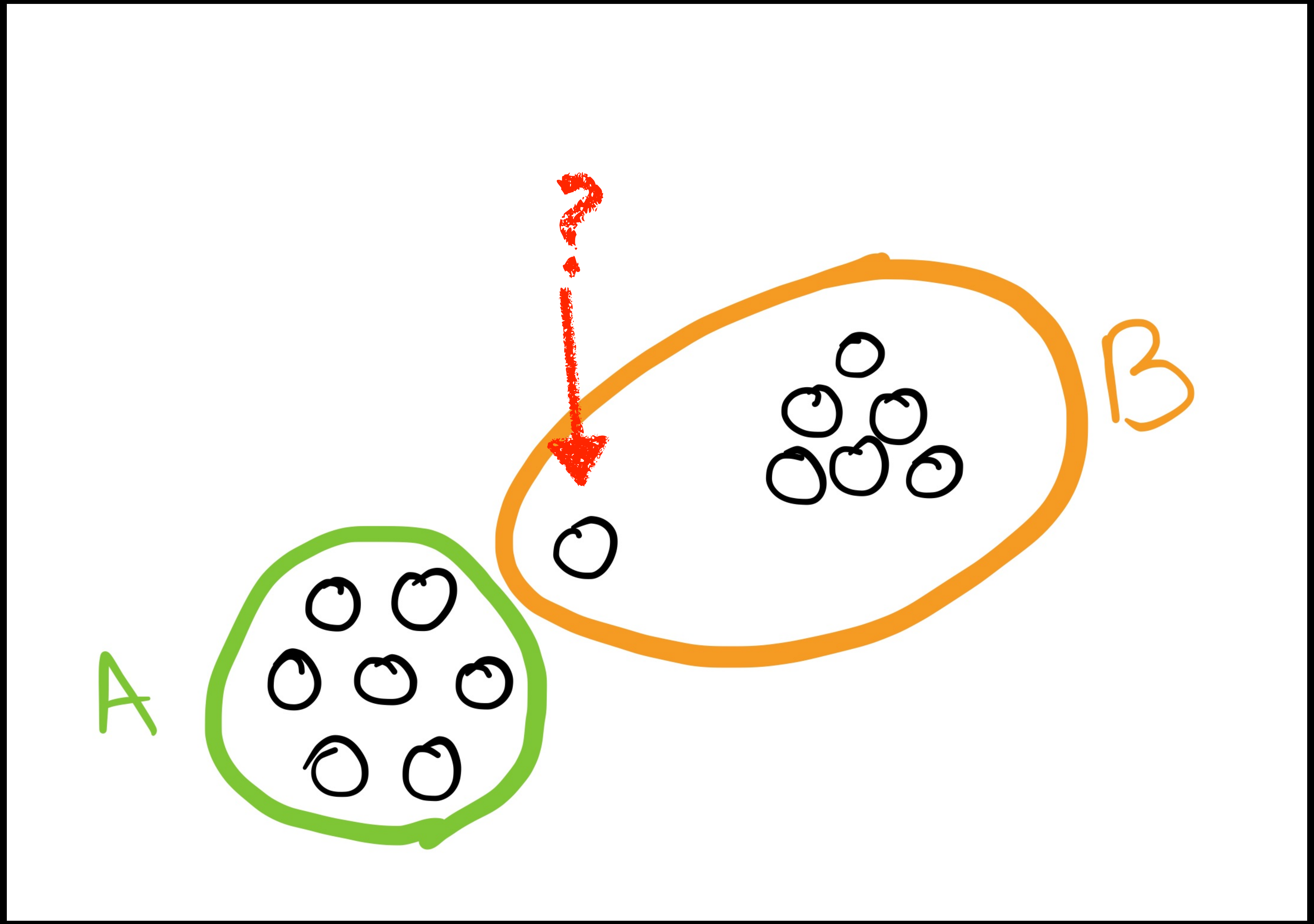
Why?



Is this object allocated to the proper cluster?

Objects should be allocated to the cluster they are nearest to, according to some distance.

Can we quantify this?



Silhouette coefficient

For a data set, D, of n objects, suppose D is partitioned into k clusters, C_1, …, C_k. For each object $o \in D$, we calculate $a(o)$ as the average distance between o and all other objects in the cluster to which o belongs. Similarly, b(o) is the minimum average distance from o to all clusters to which o does not belong. Formally, suppose $o \in C(1 \leq i \leq k)$; then

$$a(o) = \frac{\sum_{o' \in C_i, o \neq o'} dist(o, o')}{|C_i| - 1} \qquad b(o) = \min_{c_j : 1 \leq j \leq k, j \neq i} \frac{\sum_{o' \in C_j} dist(o, o')}{|C_j|}$$

Silhouette coefficient

$$a(o) = \frac{\sum_{o' \in C_i, o \neq o'} dist(o, o')}{|C_i| - 1} \qquad b(o) = \min_{c_j : 1 \leq j \leq k, j \neq i} \frac{\sum_{o' \in C_j} dist(o, o')}{|C_j|}$$

- a(o) reflects the compactness score of the cluster to which o belongs. Smaller values → cluster of o is compact

- b(o) reflects shows how separated o is from other clusters. Large values → o is more separated from other clusters than its own

Silhouette coefficient

The Silhouette Coefficient of o is then defined as:

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

$s(o) \in [-1,1]$:

- Values close to 1 indicate that o is close to its own cluster and far from other clusters.

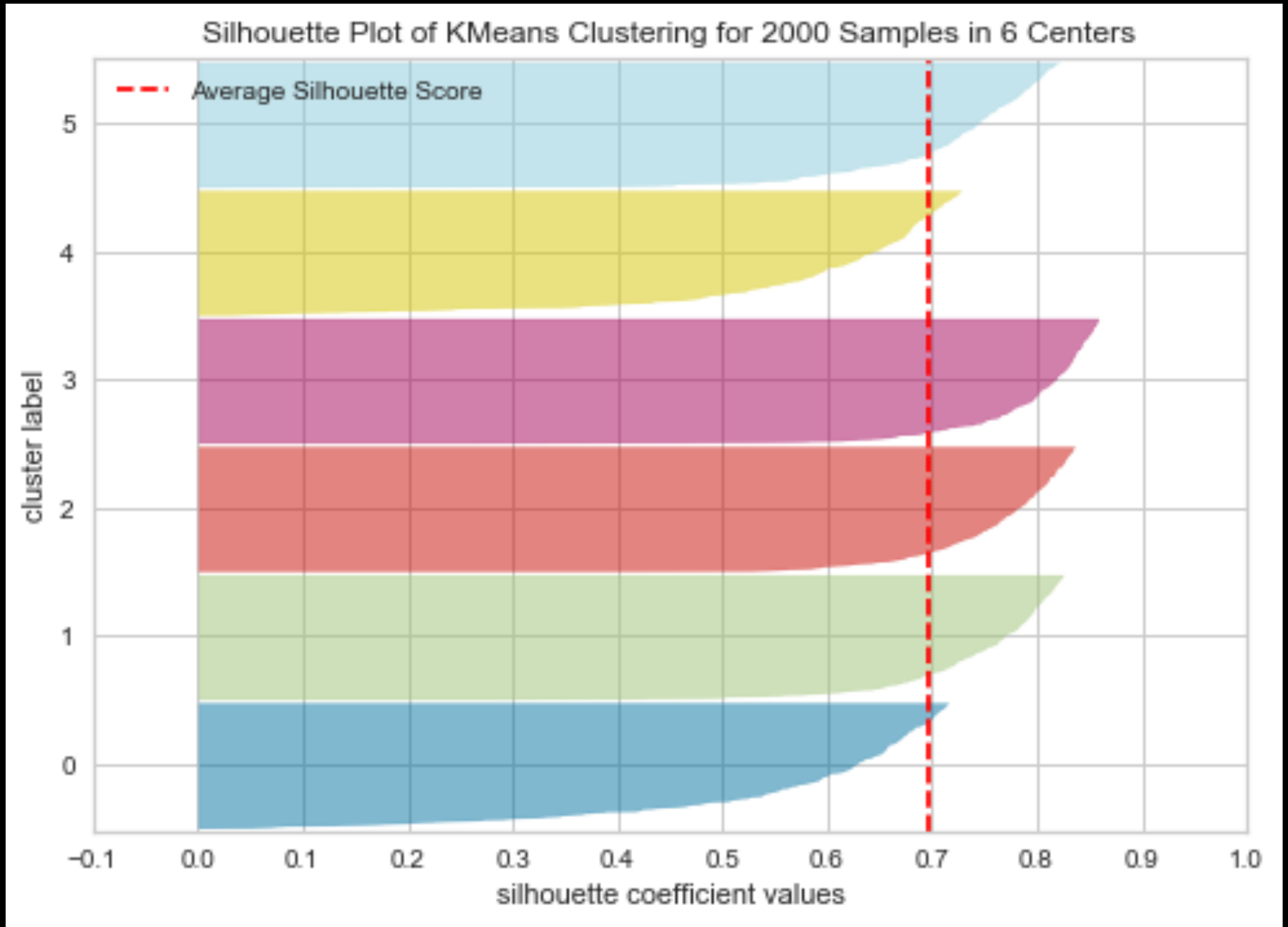
- Values approaching -1 indicate that o is far from its own cluster and close to points from another cluster.

Silhouette coefficient

We can average the Silhouette score to get scores per cluster, or per clustering.

We can also create Silhouette Coefficient plots.

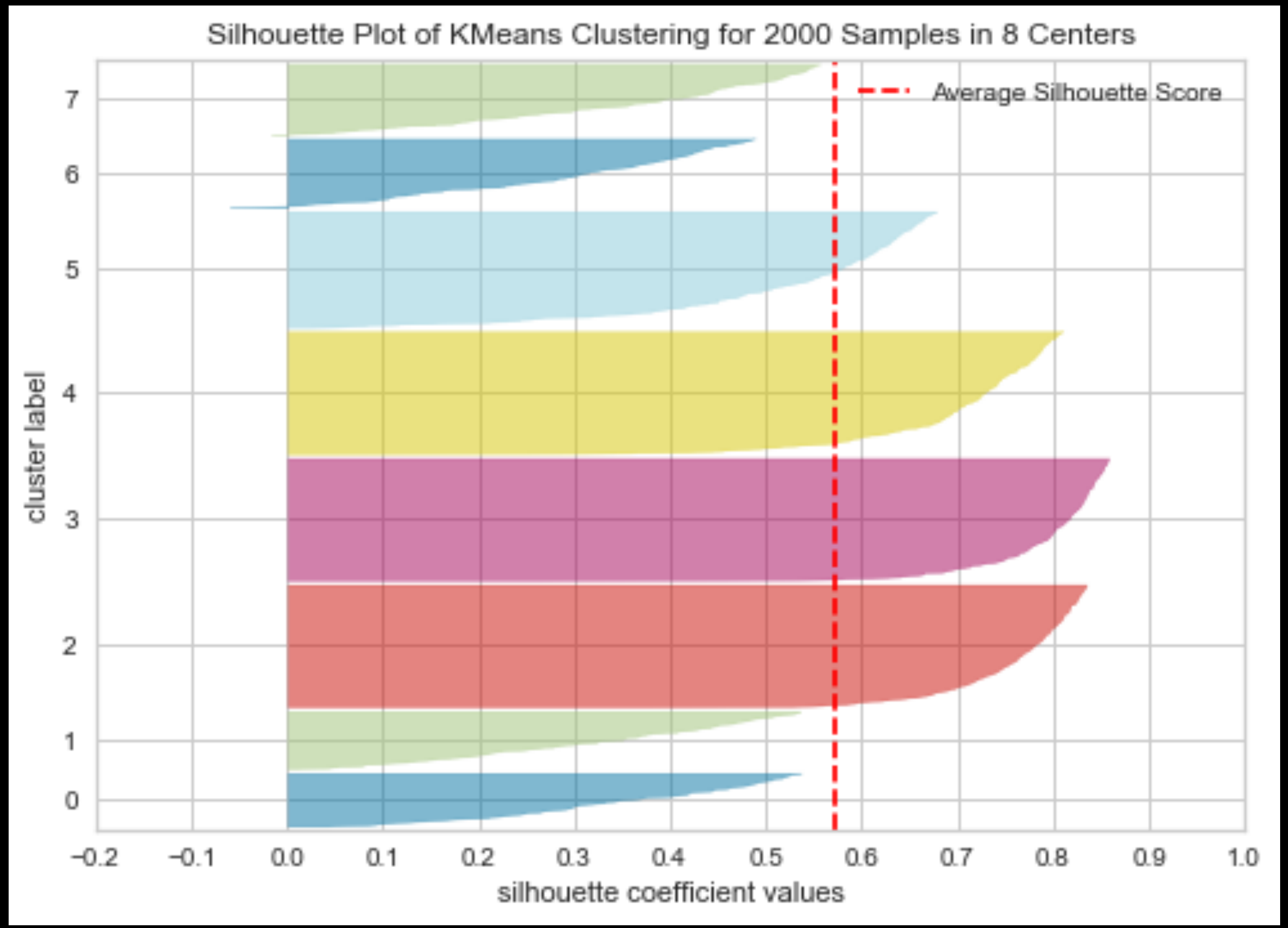
We can use the per clustering Silhouette Coefficient together with the Elbow method.



k-means, k = 6 **Silhouette Coefficient: 0.70**

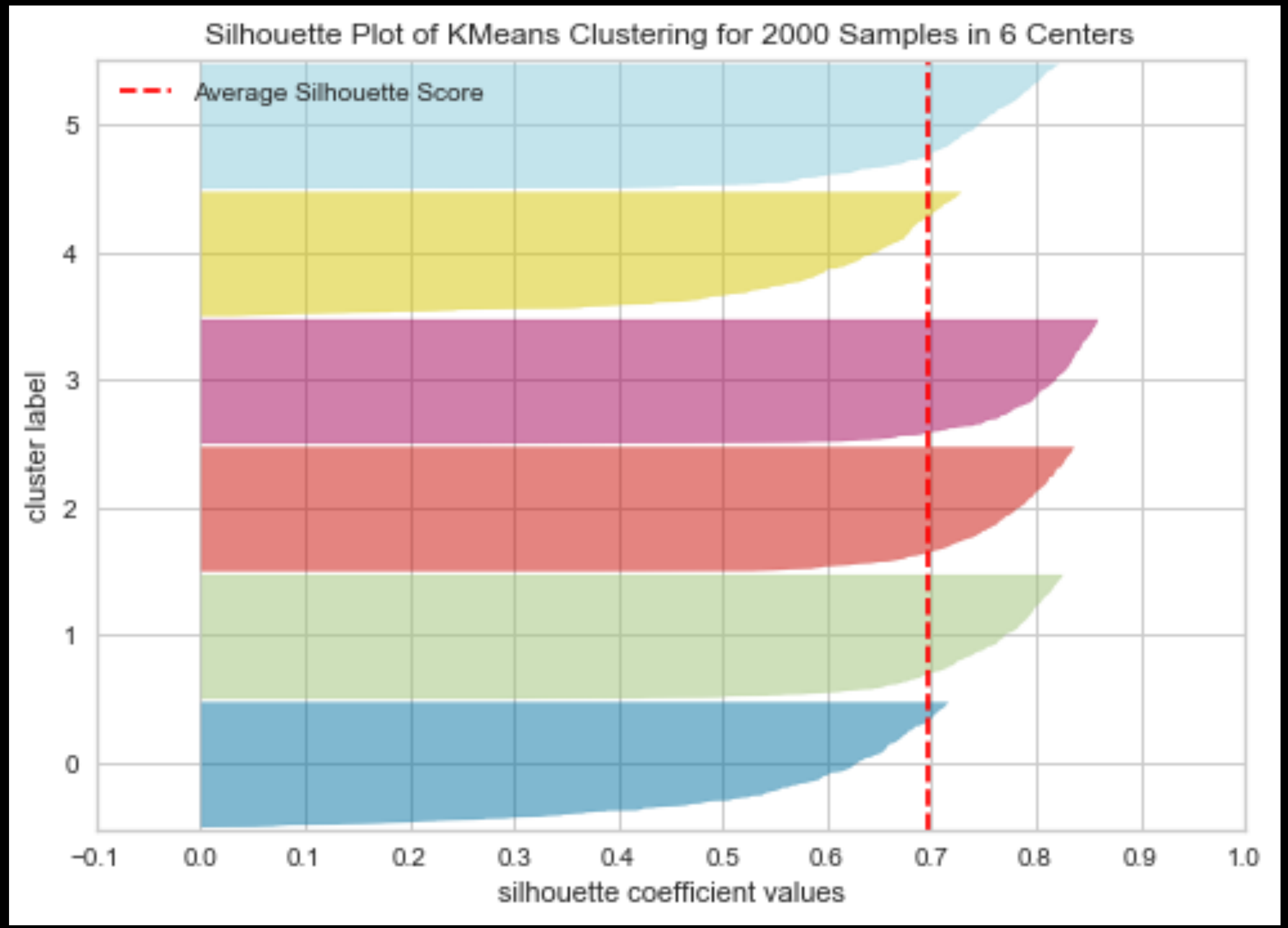
Silhouette coefficient

GOOD



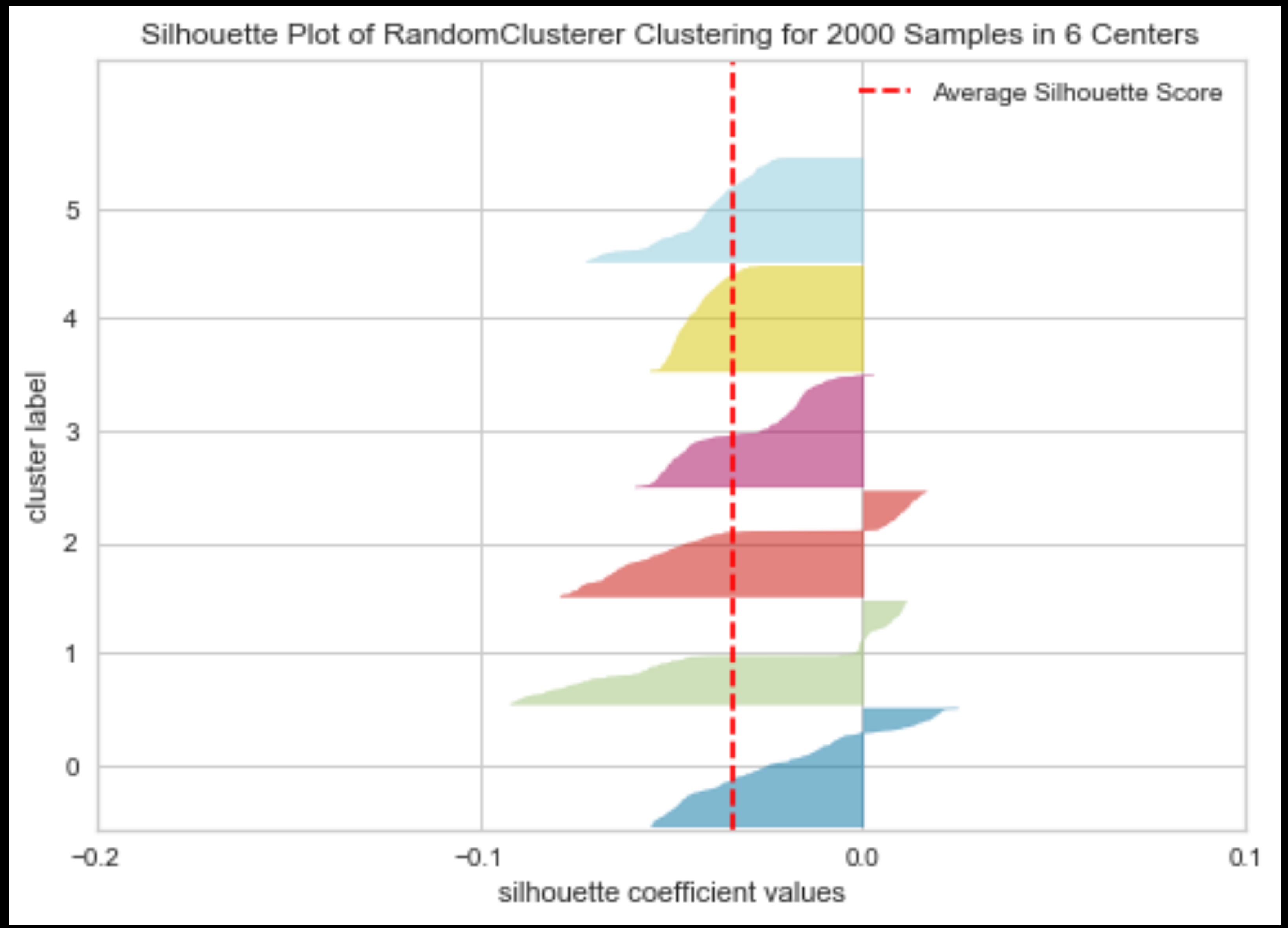
k-means
k = 8
Silhouette Coefficient: 0.57

BETTER



k-means
k = 6
Silhouette Coefficient: 0.70

BAD



Random clustering
k = 6
Silhouette Coefficient: -0.03

Extrinsic evaluation

- We have access to the ground truth data, i.e. for every point $x_i \in X$, we have $y_i \in Y$ that is the index of the correct cluster for x_i.
- Extrinsic evaluation tries to capture the extent to which points from the same partition appear in the same cluster, and that to which points from different partitions are separated into different clusters.
- There is usually a trade-off between the two goals. Each measure aims for a different balance of the two goals.

Extrinsic methods

- We can use extrinsic methods to:
  - Validate clustering algorithms (using real-world or synthetic data).
  - Validate clusterings using cross-validation (train on unlabeled data, test on the labeled data).
- All extrinsic measures rely on the $r \times k$ *contingency table N* induced by a clustering *C* and the ground-truth partitioning *T:*

$$N(i,j) = n_{ij} = |C_i \cap T_j|$$

(n_{ij} is the number of elements in C_i that are in partition T_j)

Purity

- Measures to what degree does each cluster C_i contain only objects from one partition (label).

- For cluster *i, purity* is defined as:

$$purity_i = \frac{1}{n_i} \max_{j=1}^{k} \{nij\}$$

- For a clustering C, it is the weighted sum of each cluster's purity:

$$purity = \sum_{i=1}^{r} \frac{n_i}{n} purity_i = \frac{1}{n} \sum_{i=1}^{r} max_{j=1}^{k} \{n_{ij}\}$$

Purity

- For a clustering C, it is the weighted sum of each cluster's purity:

$$purity = \sum_{i=1}^{r} \frac{n_i}{n} purity_i = \frac{1}{n} \sum_{i=1}^{r} max_{j=1}^{k} \{n_{ij}\}$$

- $purity \in [0,1]$, the larger the value, the better the clustering matches the ground-truth partitioning
  - purity = 1 → All clusters contain elements of the same ground-truth partitioning
  - purity $<$ 1 → Percentage of elements in the clustering that are assigned to a cluster whose predominant ground-truth label they share
- For k $>$ number of partitions in ground-truth data, purity has a tendency to be optimistic. What is the value of purity for a clustering where every element is allocated to its own cluster, despite the fact that the ground-truth data only contains a small number of partitions p?

Purity

- For a clustering C, it is the weighted sum of each cluster's purity:

$$purity = \sum_{i=1}^{r} \frac{n_i}{n} purity_i = \frac{1}{n} \sum_{i=1}^{r} max_{j=1}^{k} \{n_{ij}\}$$

- What is the value of purity for a clustering where every element is placed in its own cluster, despite the fact that the ground-truth data only contains a small number of partitions p?

Break

(Questions and answers after the break)

Questions?

What if we didn’t have to specify the number of clusters?

Hierarchical clustering

- We might want to partition data into groups at different levels of granularity and construct a hierarchy.
- The hierarchy contains information about cluster relationships.
- It can be useful for data summarization or visualization.

Hierarchical clustering

We can cluster hierarchically using two approaches:

1. Start by assuming each object is its own cluster and iteratively merge clusters — *Bottom-up clustering (or Agglomerative clustering)*

2. Start by assuming there is only one cluster containing all objects and then iteratively split one cluster at a time until each object is its own cluster — *Top-down clustering (or Divisive clustering)*

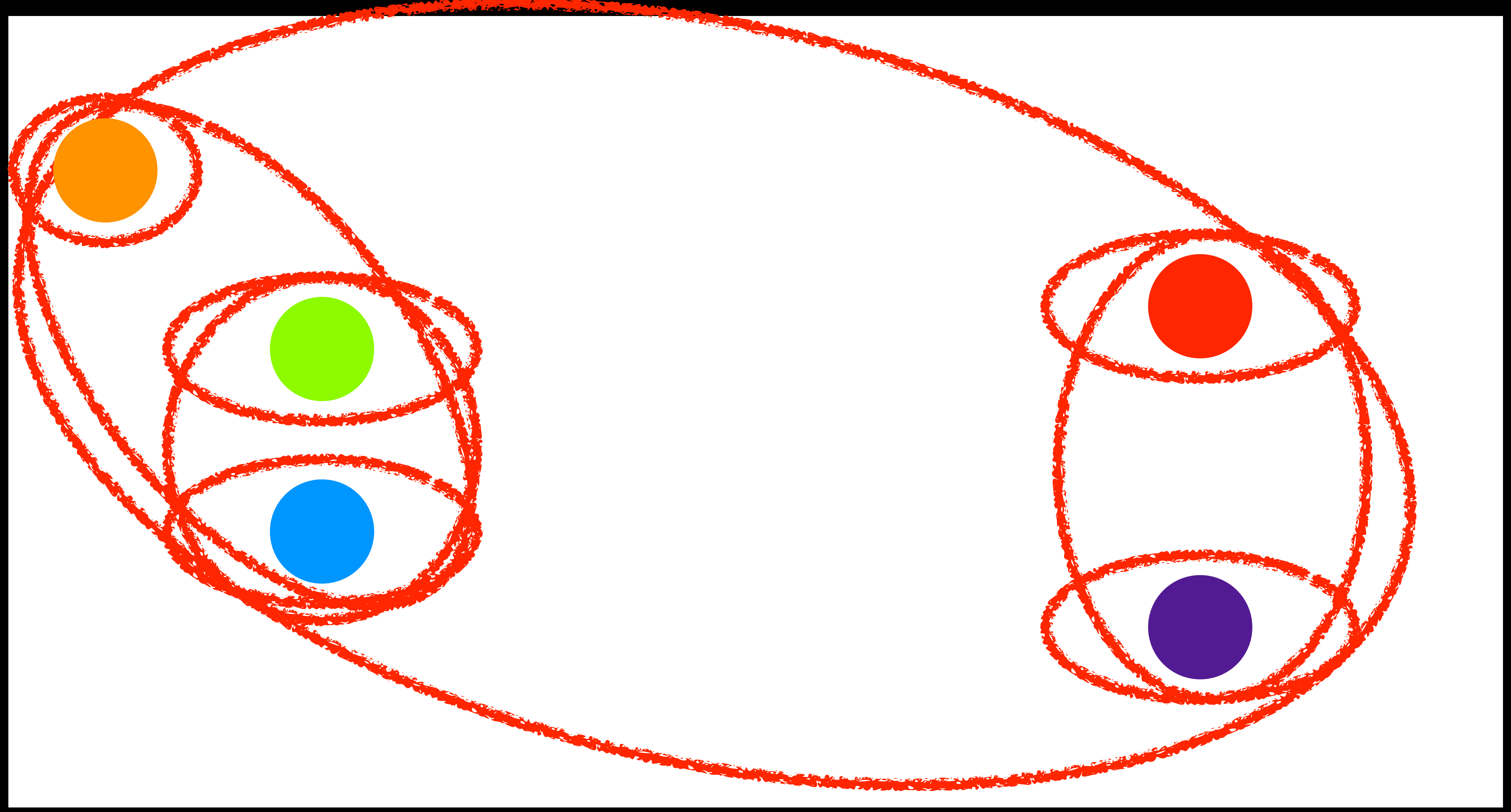
Hierarchical Agglomerative Clustering

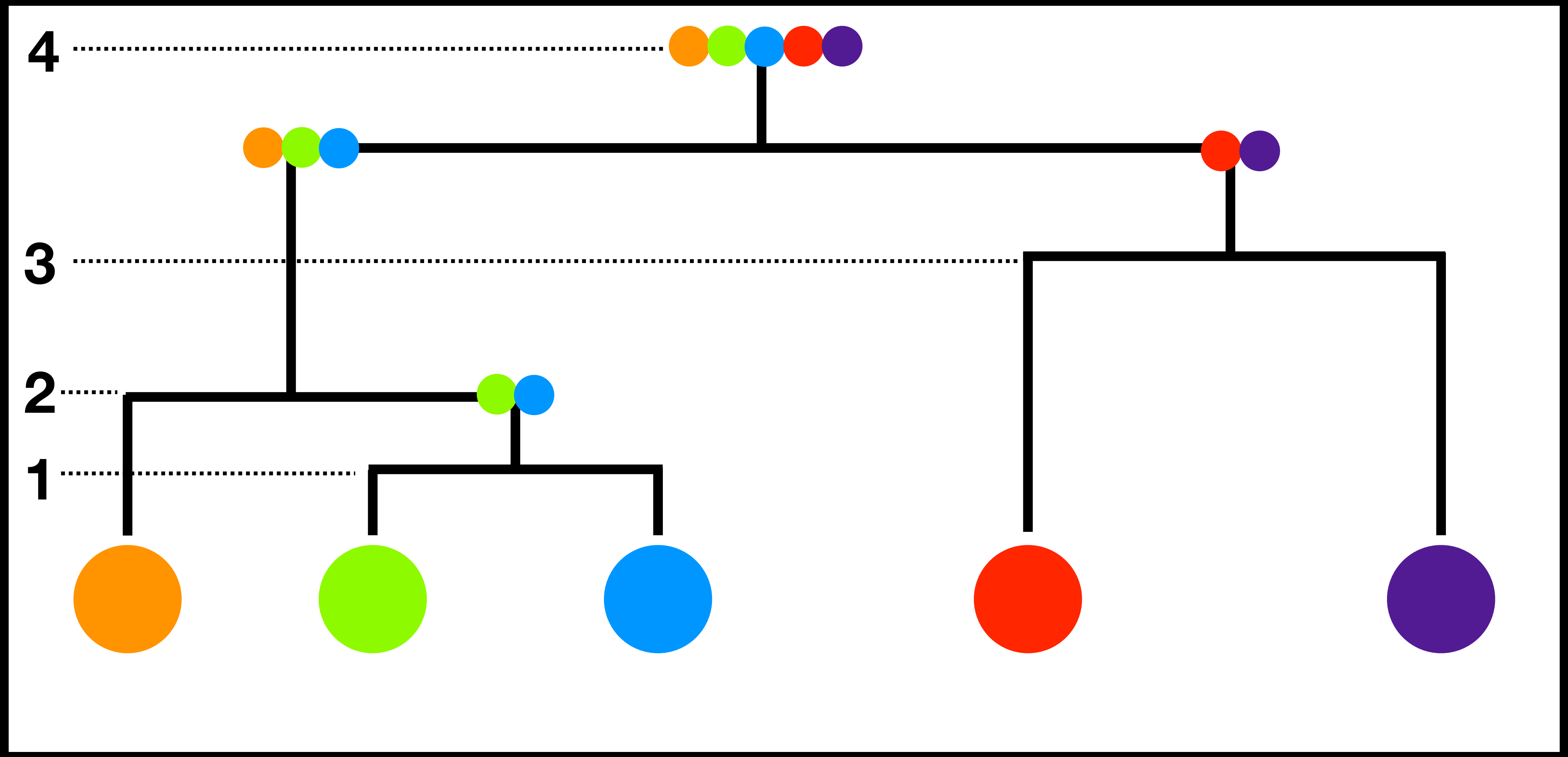
1. Start by considering each point as its own cluster. I.e., we have $k = 5$ clusters.

2. While there are more than 1 cluster left:

   - Compute the distance between each pair of clusters.

   - Merge the two clusters closest to each other.





Challenges for hierarchical clustering

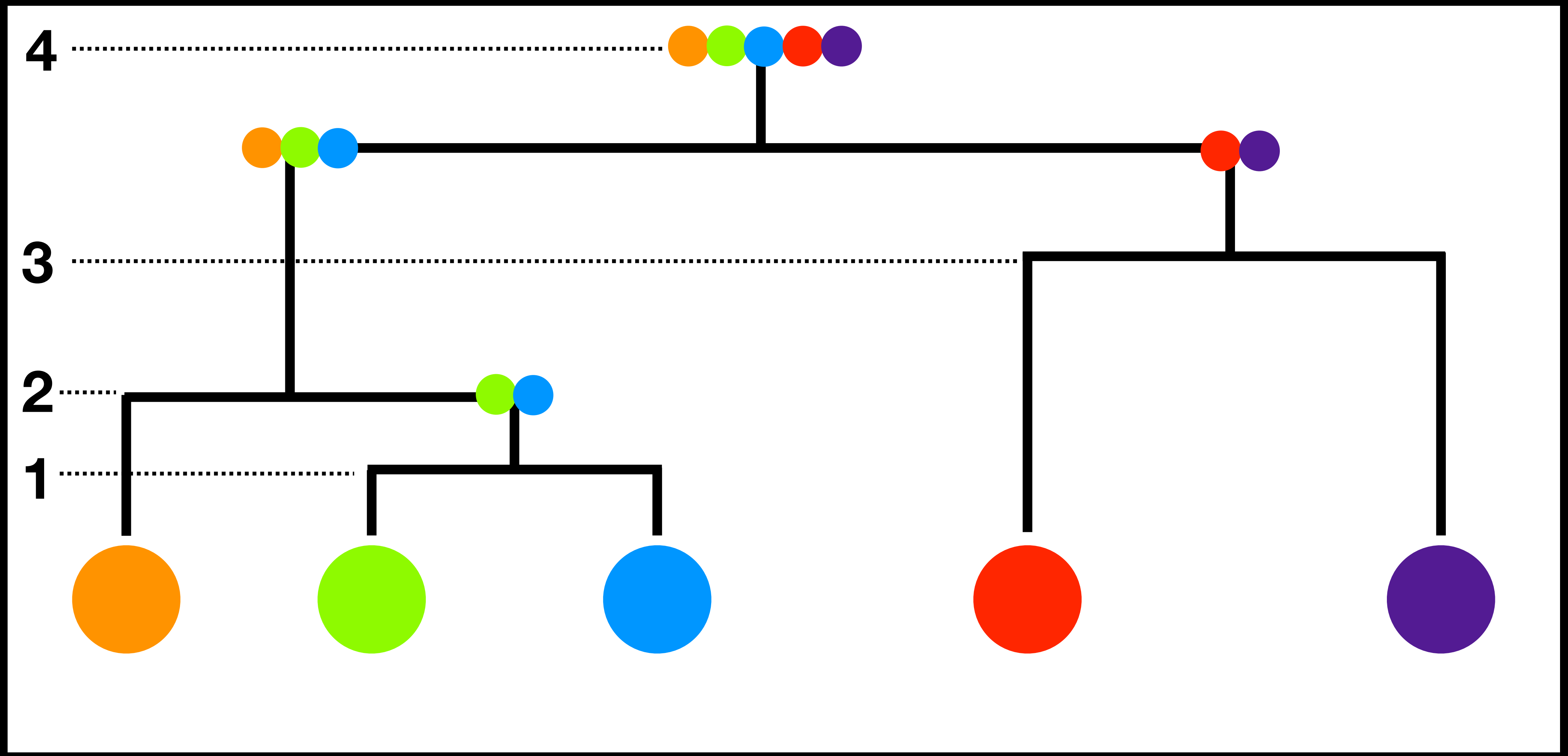
- Once a merge / division decision is made, it cannot be reverted.
- Finding the clusters to merge, or the proper division can be compute intensive.
- There can be more than one possible hierarchy.

Dendrograms

We can represent hierarchical clusters graphically using a tree structure called a *dendrogram.*

The dendrogram provides information about:

- The type of clustering (bottom-up/top-down)
- Hierarchy
- Order of merges/splits

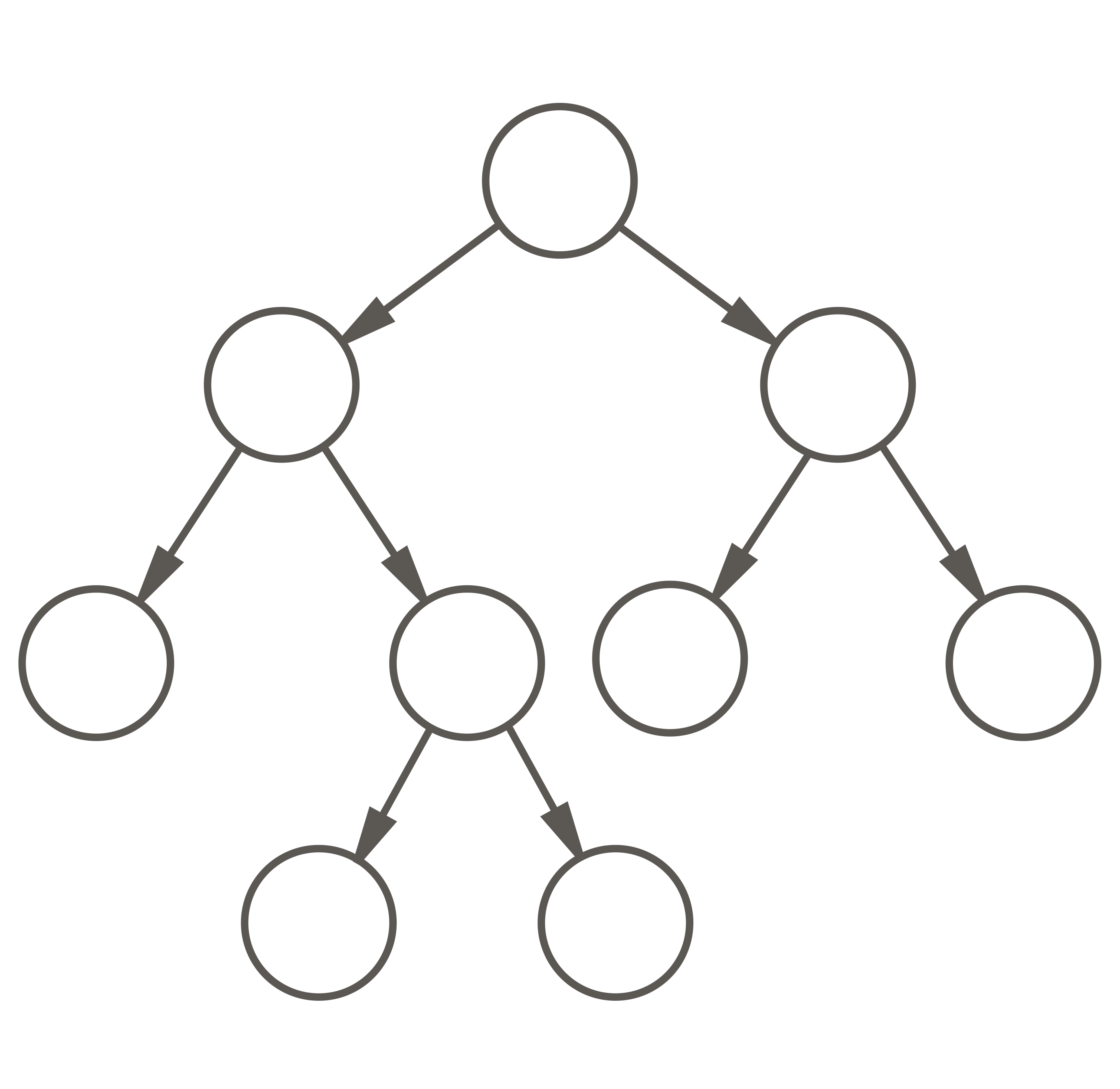


Discussion

- We can convert addresses of all people in Denmark into latitude, longitude, and altitude.
    - E.g. Hans Jensens Stræde 45, 1TH 5000 Odense → 55.39878, 10.39064, 4m
- What would we get at the different layers if we clustered these points using bottom-up hierarchical clustering? (Discuss with your neighbor)

Hierarchical clustering of words

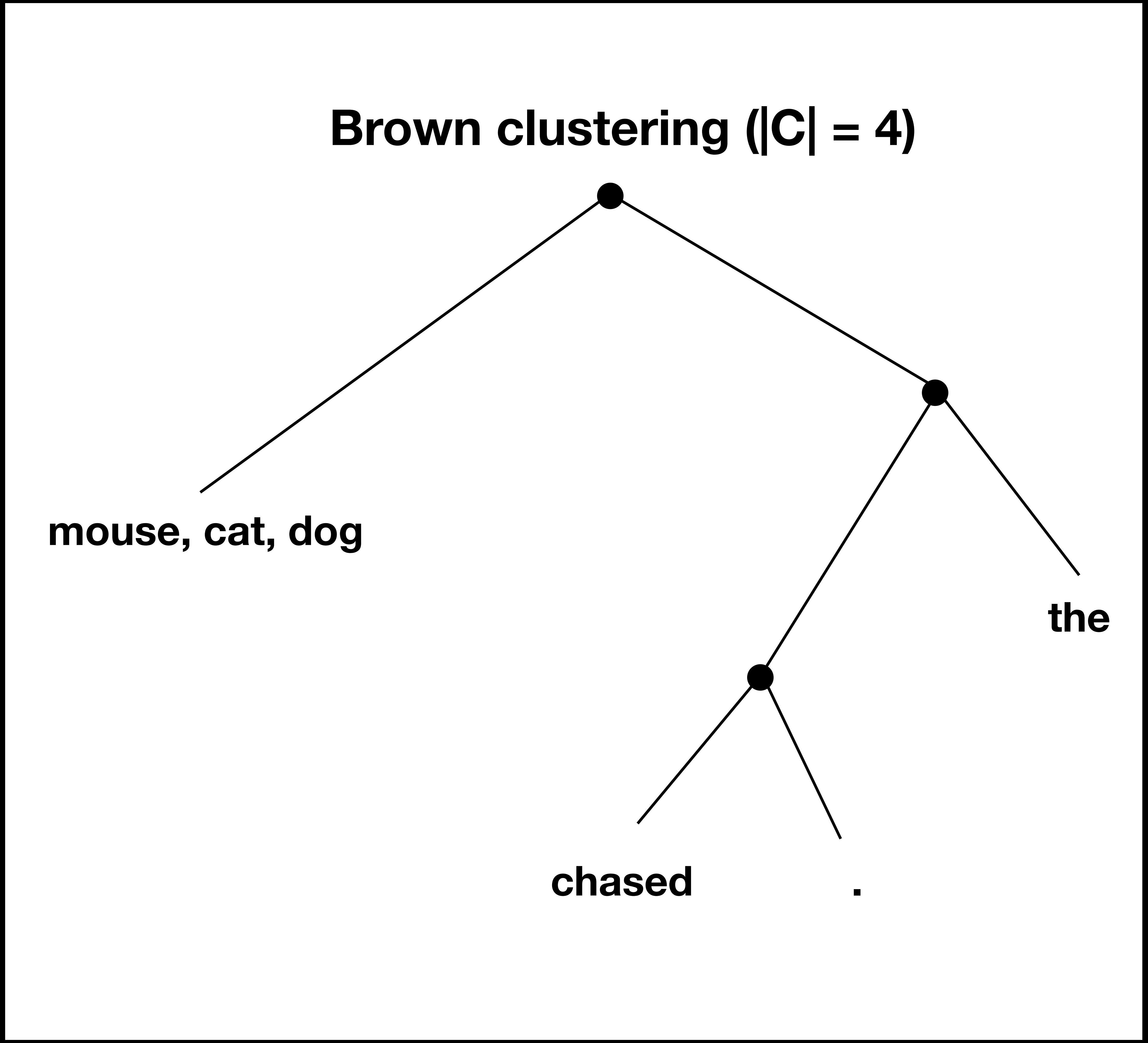
- Hierarchical clustering can provide information about word role and meaning.
- Words do not exist in a metric space, so we need a new way to gain information about their similarity.
- We will consider words to be similar if they are used similarly.
- The algorithm is called Brown clustering



Brown clustering

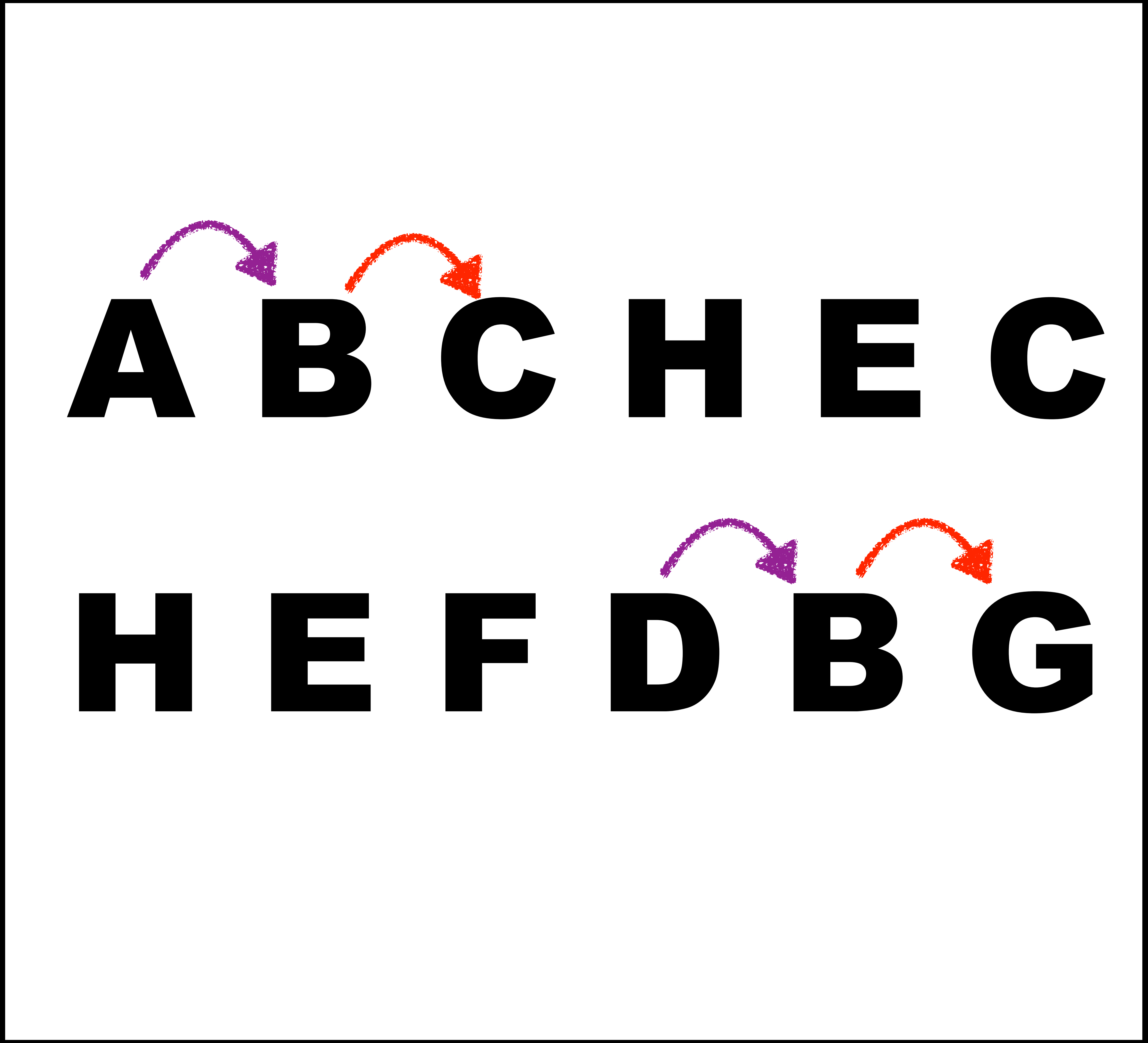
We take a corpus of text and look at word usage patterns:

```
the dog chased the cat .
the cat chased the mouse .
the mouse chased the dog .
```



Brown clustering

- We take a corpus of text and look at word usage patterns.
- Example:
  - Letters stand for words.
  - We follow word B with: C 50% of the time and G 50% of the time. We can cluster C and G.
  - Similarly, word B is preceded by A 50% of the time and D 50% of the time. We can cluster A and D.



A B C H E C

H E F D B G

Brown clustering

- What is the probability of seeing word B based on the corpus? How about word C?

$$p(B) = \frac{N(B)}{length\_corpus} = \frac{2}{11} = 0.18$$

$$p(C) = \frac{N(C)}{length\_corpus} = \frac{2}{11} = 0.18$$

- What is the probability of seeing the word B followed by word C?

$$p(B, C) = \frac{N(B, C)}{length\_corpus} = \frac{1}{11} = 0.09$$

- Is B followed by C more often than we would expect if they had no correlation? How about H followed by E?

$$pmi(B, C) = \log_2 \frac{p(B, C)}{p(B)p(C)} = \log_2 \frac{0.09}{0.18 * 0.18} = 1.47$$

$$pmi(H, E) = \log_2 \frac{p(H, E)}{p(H)p(E)} = \log_2 \frac{0.18}{0.18 * 0.18} = 2.47$$

A B C H E C

H E F D B G

Brown clustering

- Is B followed by C more often than we would expect if they had no correlation?

$$pmi(B, C) = \log_2 \frac{p(B, C)}{p(B)p(C)} = \log_2 \frac{0.09}{0.18 * 0.18} = 1.47$$

- What if C and G were in the same cluster?

$$pmi(B, [C, G]) = \log_2 \frac{p(B, [C, G])}{p(B)p([C, G])} = \log_2 \frac{0.18}{0.18 * 0.27} = 1.88$$

- Let's scale this up for the entire clustering.

A B C H E C

H E F D B G

Brown clustering

- For an entire clustering we can define Average Mutual Information (AMI) as:

$$AMI(C) = \sum_{c_i, c_j \in C} p(c_i, c_j) \log_2 \frac{p(c_i, c_j)}{p(c_i) p(c_j)}$$

- For bottom-up clustering we want to always merge those clusters that lead to the highest AMI.

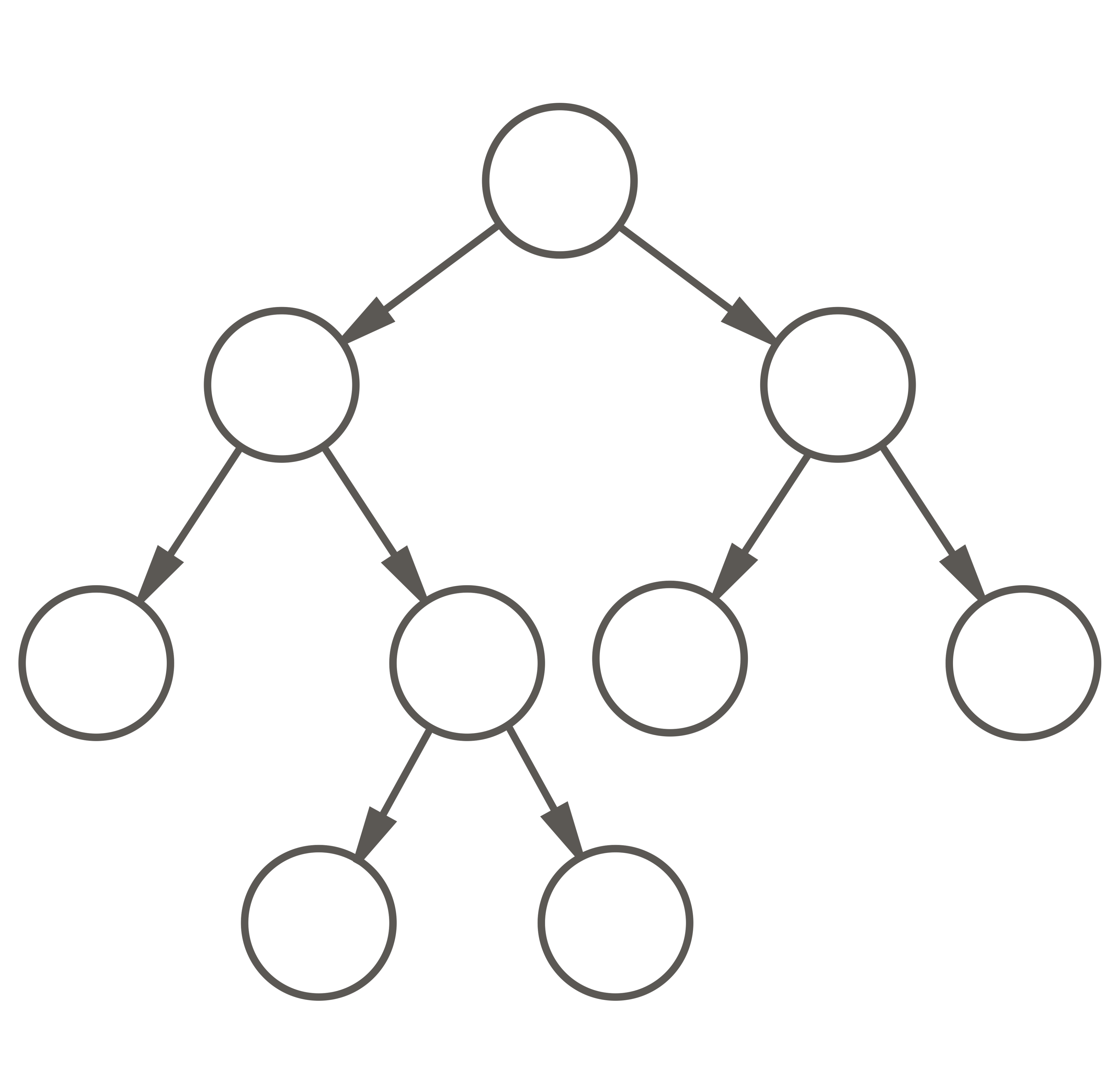
Brown clustering

1. Start by considering word in the vocabulary V as its own cluster.

2. While there are more than 1 cluster left:

    - For each pair of clusters, compute the AMI of the clustering that would result if we merged the clusters.

    - Merge the two clusters that lead to the best AMI.



Brown clustering

- Considering every combination of clusters takes time $O(|V|^2)$.

- Computing AMI takes time $O(|V|^2)$

- We have to perform $|V|$ merges to obtain the hierarchy.

- Total time complexity becomes $O(|V|^5)$, and V can be large.

- We can build an approximation.

Brown clustering

We can construct an approximation of the algorithm as follows:

1. Assume we want to build a hierarchy with base |C|. We consider every word as its own cluster.
2. We set an active window of size |C| and fill it with the most frequent |C| clusters.
3. While the active window contains more than 1 cluster:
    1. We find the best pair of clusters to merge based on AMI and merge them.
    2. If there are clusters that are not in the active window, we take the most frequent one and include it in the active window.

What if the active window is the same size as |V|?

Brown clustering

If the active window is equal to |V|, then we have the non-approximated algorithm.

Active windows a little larger than |C| result in better clusters.

Example Brown clusters

friday monday thursday wednesday tuesday saturday sunday weekends sundays saturdays

june march july april january december october november september august

iraq london texas canada washington houston paris california australia earth

n't nt n’t n`t conceivably believe notionally

1980s 1970s 1960s 1990s 1950s 1930s 1920s 1940s 1890s 1800s 1880s 1870s 1830s 1860s 1900s 1850s 1820s 1600s nineties thirties eighties

Discussion

Why are bottom-up agglomerative clustering algorithm (like Brown clustering) more demanding than k-means?

(discuss with your neighbor)

Global vs Local metrics

In an algorithm like k-means we allocate each point to its closest cluster (i.e. a local measure) which has complexity $O(num\_objects \times k \times i)$.

In order for HAC to obtain the hierarchy it performs $num\_objects - 1$ merges, and for every merge it must find the best pair of clusters. This gives a time complexity of $O(num\_objects^3)$.

Various methods exist to speed up HAC, e.g. caching of distances.

Summary

- Before we cluster data we should make sure it has a structure and find the ideal number of clusters.

- Clustering methods can be evaluated intrinsically (based on properties of the clusters themselves) or extrinsically (based on labels).

- Hierarchical clustering provides clusters at different granularities which provides information about relationships between clusters.

- Hierarchical clustering is computationally more expensive than partitioning methods.