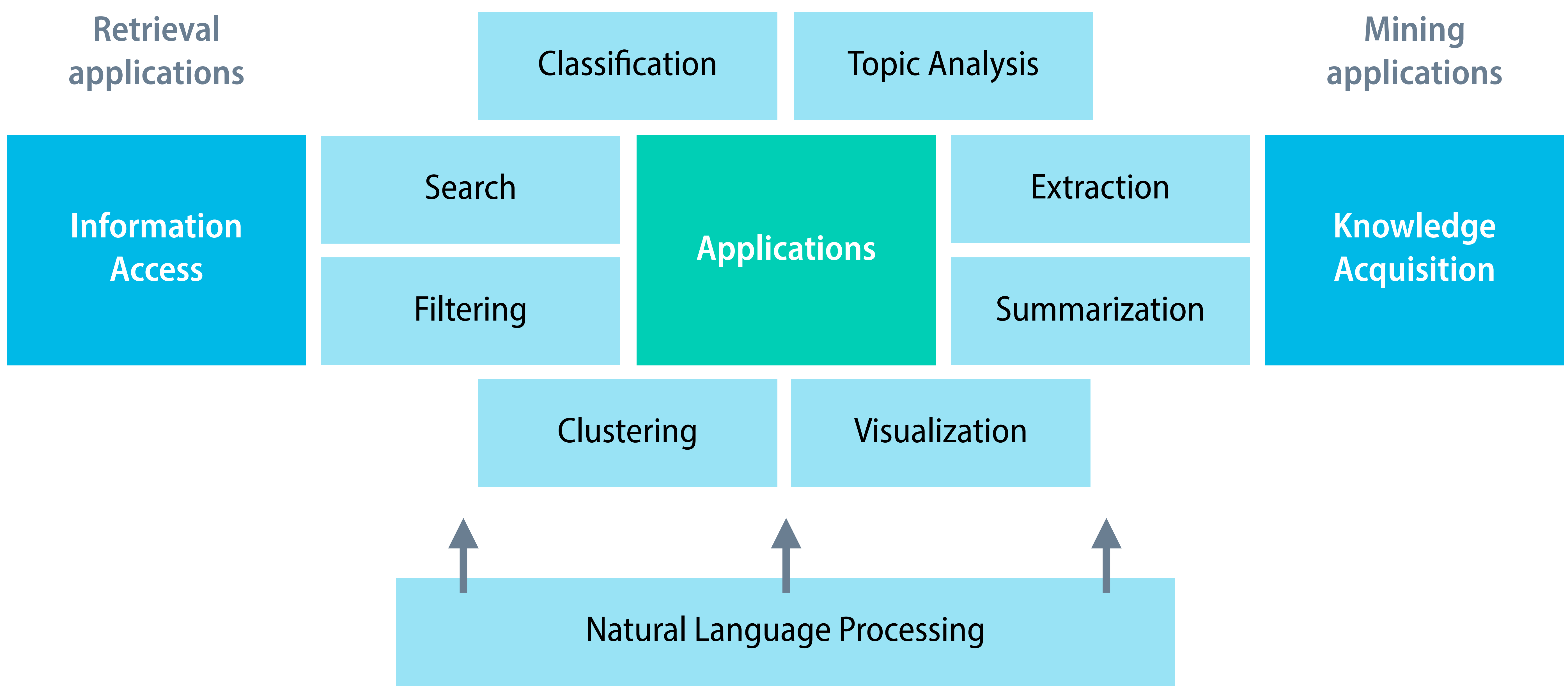
732A92/TDDE16 Text Mining (2020)

# Text clustering and topic modelling

Marco Kuhlmann
Department of Computer and Information Science

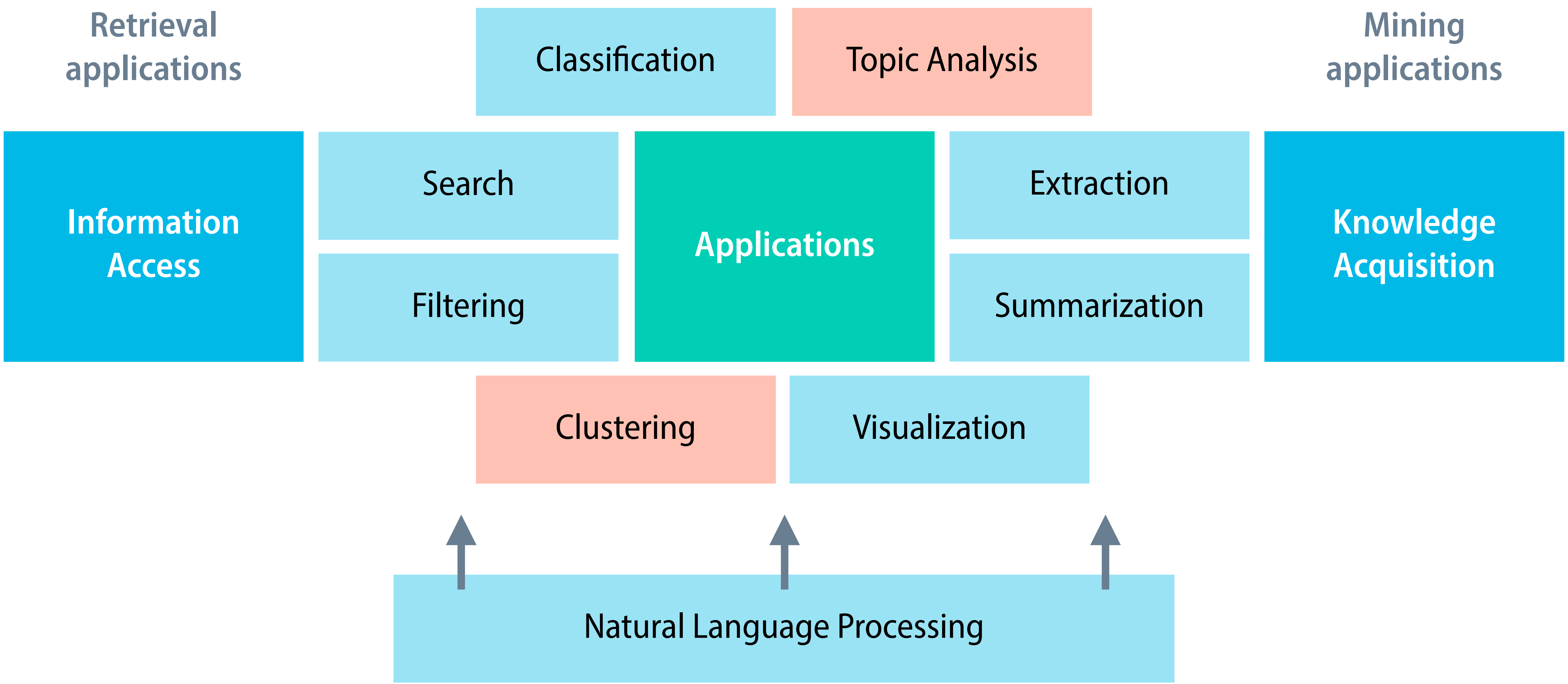
LINKÖPING UNIVERSITY

# Reminder: Conceptual framework for text mining




Adapted from Zhai and Massung (2016)

# Reminder: Conceptual framework for text mining



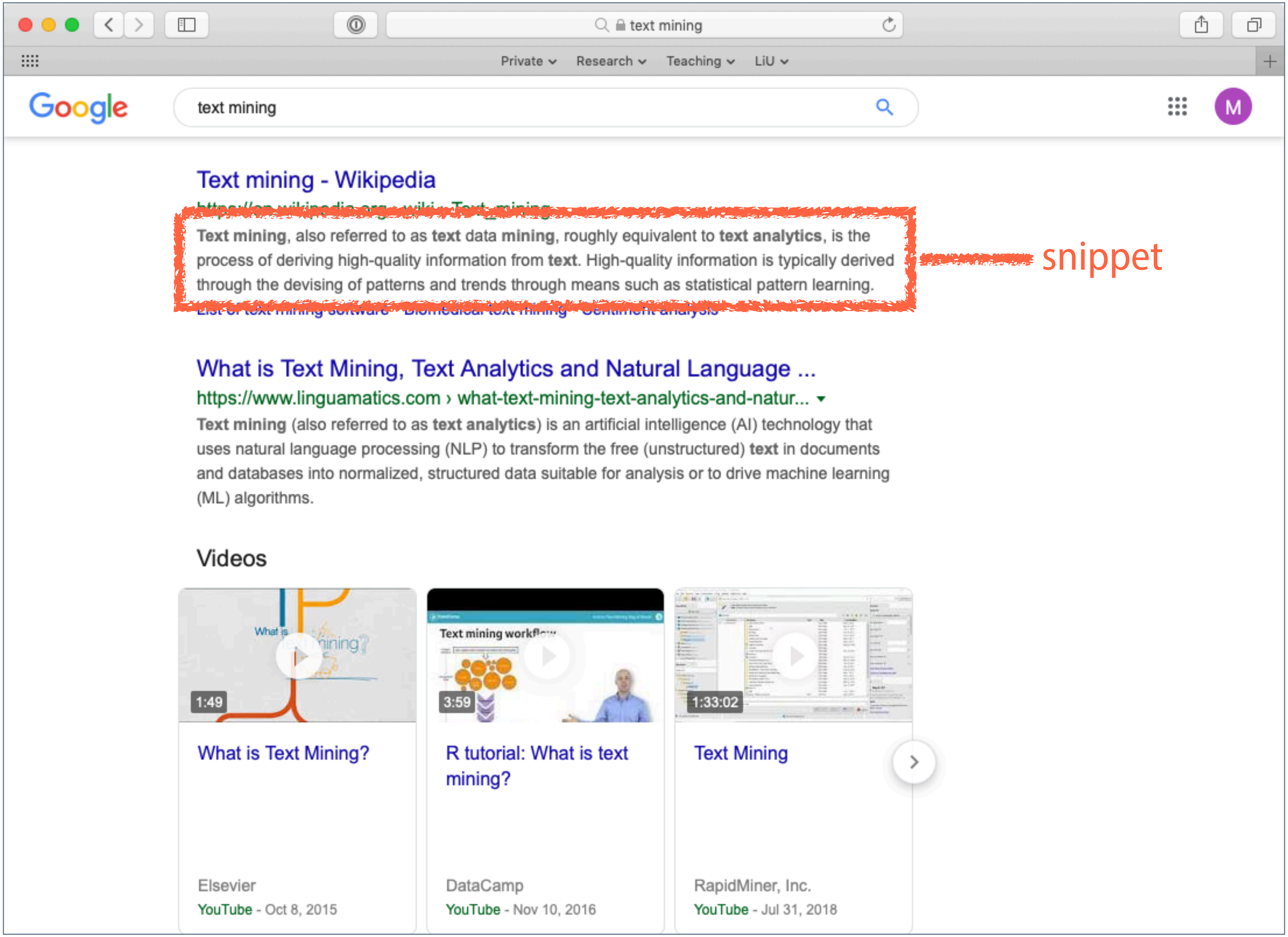

Adapted from Zhai and Massung (2016)

# Text clustering

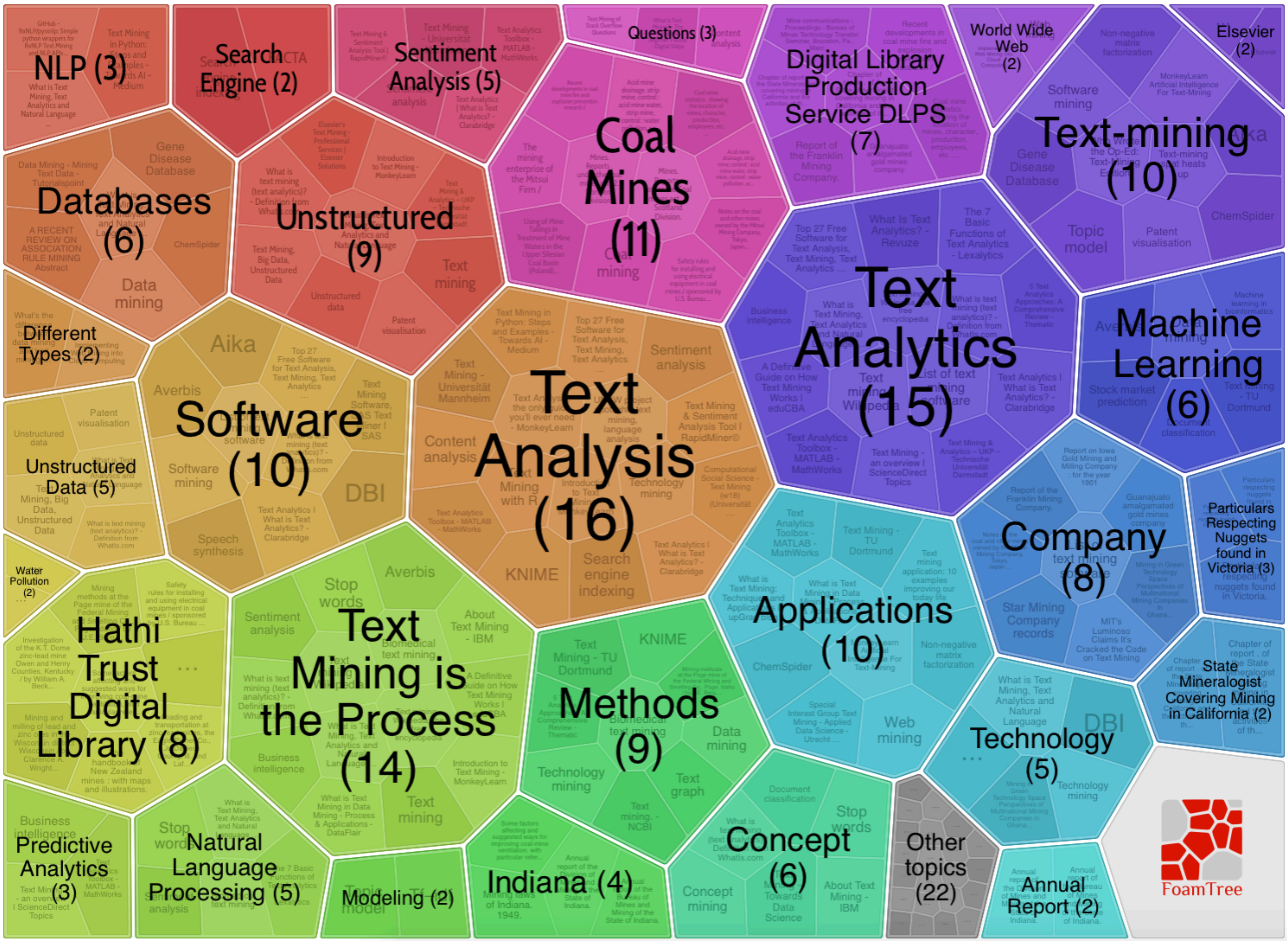
- **Text clustering** is the task of grouping similar texts together. What is considered 'similar' depends on the application.
- Clustering is a central tool in exploratory data analysis, where it can help us to get insights into the distribution of a data set.

  Example: Clustering of search results
- Clustering is also useful as a pre-processing technique in knowledge-focused applications.

  Example: Brown clustering

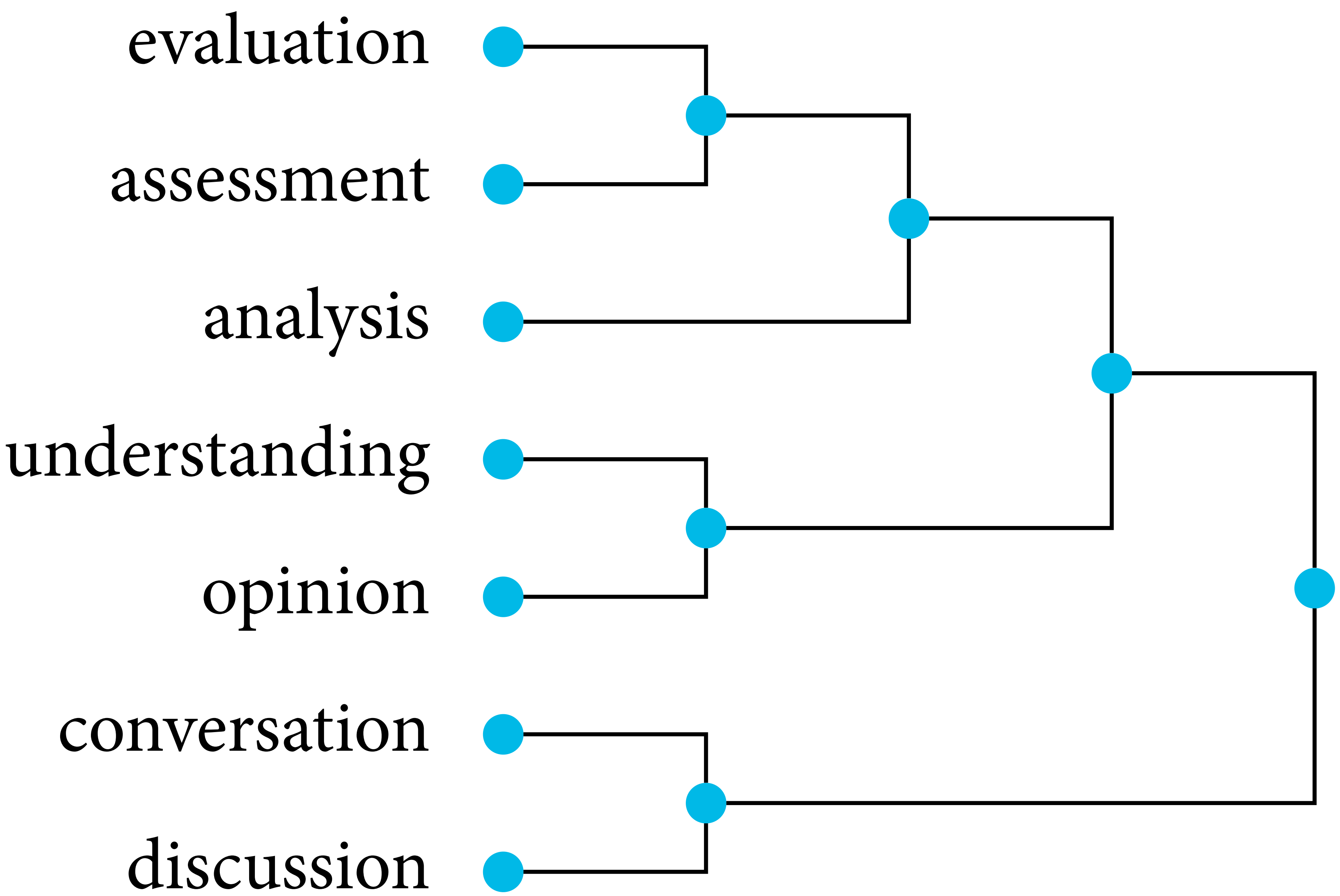


text mining
Private
Research
Teaching
LiU
Google
text mining
M
Text mining - Wikipedia
https://en.wikipedia.org › wiki › Text_mining
Text mining, also referred to as text data mining, roughly equivalent to text analytics, is the process of deriving high-quality information from text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning.
List of text mining software · Biomedical text mining · Sentiment analysis
snippet
What is Text Mining, Text Analytics and Natural Language ...
https://www.linguamatics.com › what-text-mining-text-analytics-and-natur...
Text mining (also referred to as text analytics) is an artificial intelligence (AI) technology that uses natural language processing (NLP) to transform the free (unstructured) text in documents and databases into normalized, structured data suitable for analysis or to drive machine learning (ML) algorithms.
Videos
1:49
What is Text Mining?
Elsevier
YouTube - Oct 8, 2015
Text mining workflow
3:59
R tutorial: What is text mining?
DataCamp
YouTube - Nov 10, 2016
1:33:02
Text Mining
RapidMiner, Inc.
YouTube - Jul 31, 2018



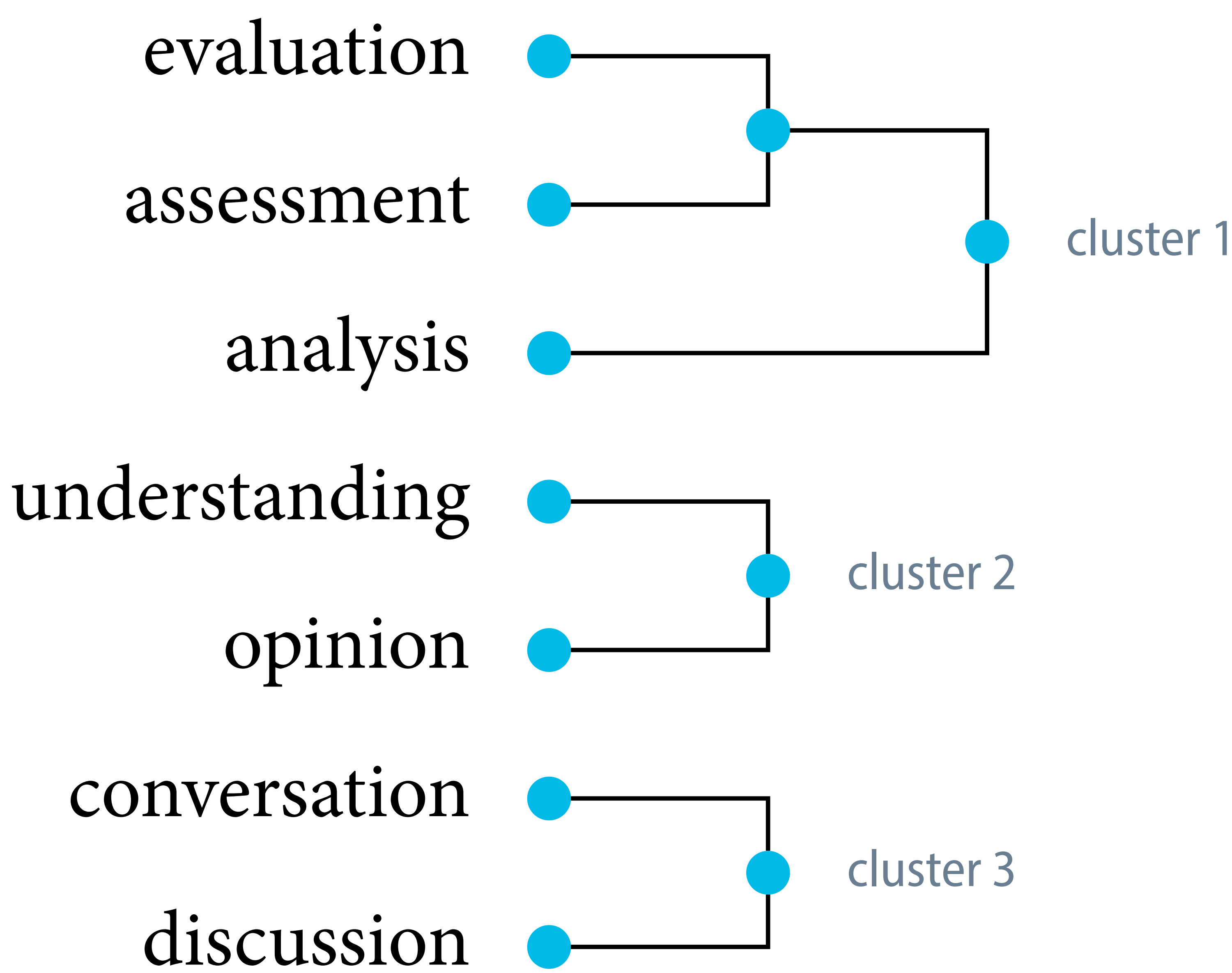

Source: carrot2.org, accessed 2019-11-17

# Brown clustering



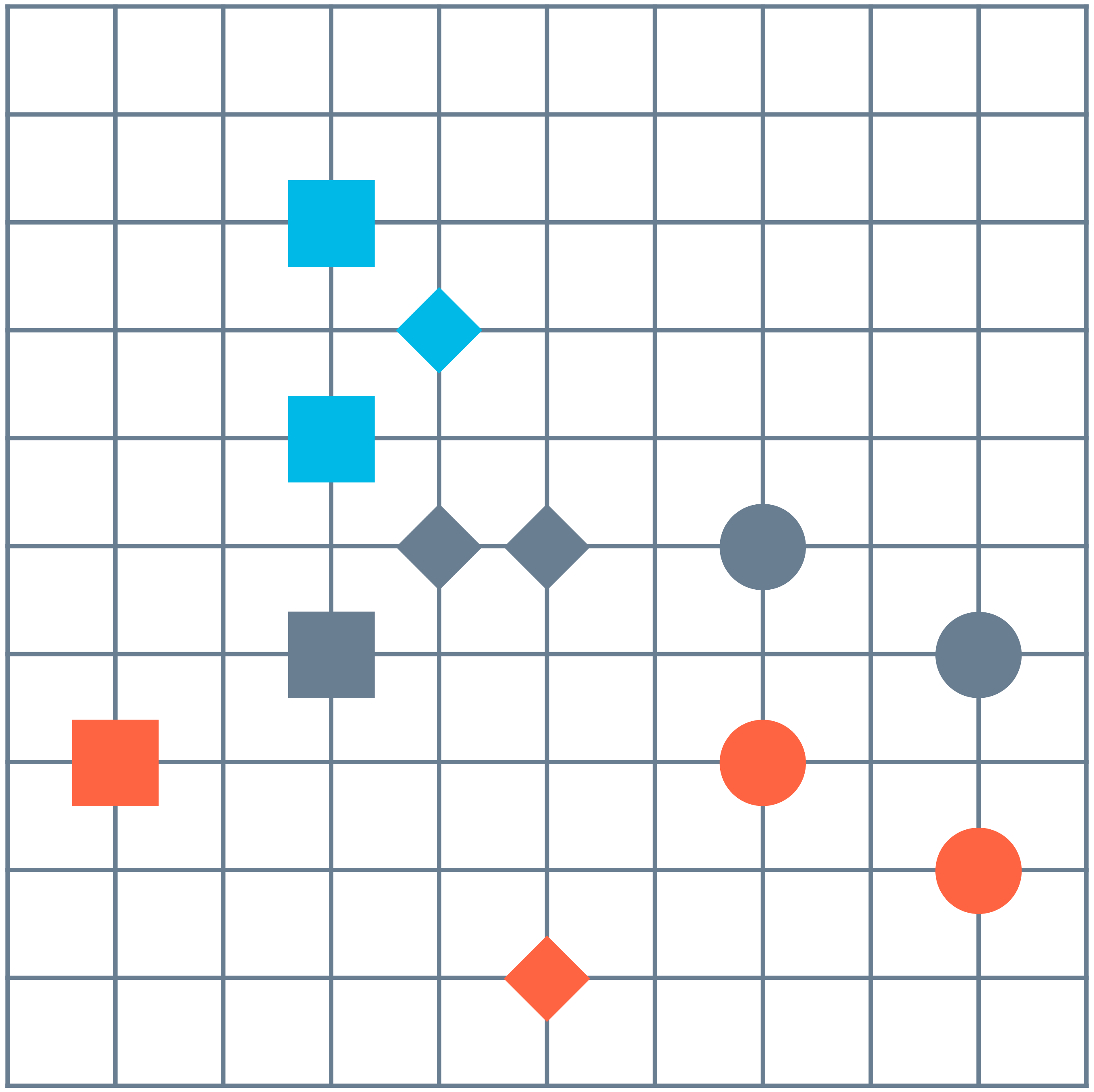

Source: Brown et al. (1992)

# Brown clustering

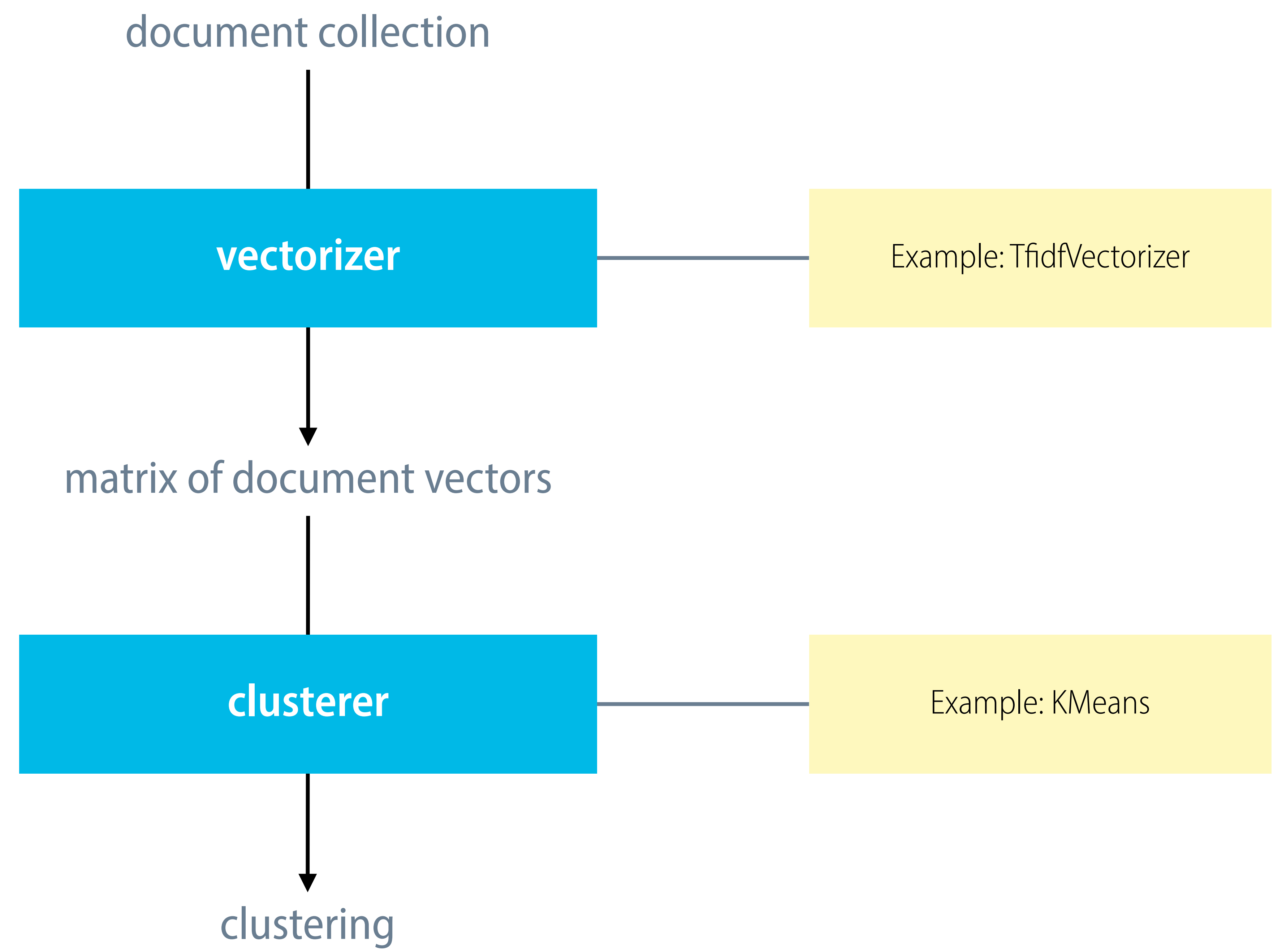



Source: Brown et al. (1992)

# Different notions of similarity



# The standard text clustering pipeline



# Reminder: Two challenges in text classification

- Standard document representations such as the bag of words easily yield tens of thousands of features.

  computational challenge, data sparsity

- Many document collections are highly imbalanced with respect to the represented classes.

  frequency bias, problems for evaluation

# Hard clustering and soft clustering

- **Hard clustering**

  Each document either belongs to a cluster or not.

  hierarchical clustering, k-means, DBSCAN

- **Soft clustering**

  Each document belongs to each cluster to a certain degree.

  LDA (topic model)

# This lecture

- Introduction to text clustering
- Similarity measures
- An overview of hard clustering methods
- Evaluation of hard clustering
- Soft clustering: Topic models

# Similarity measures

# Similarity measures

- Informally speaking, a **similarity measure** is a real-valued function that quantifies the similarity between two objects.

- There is no single definition of these functions, but they often appear as the complements of distance functions.

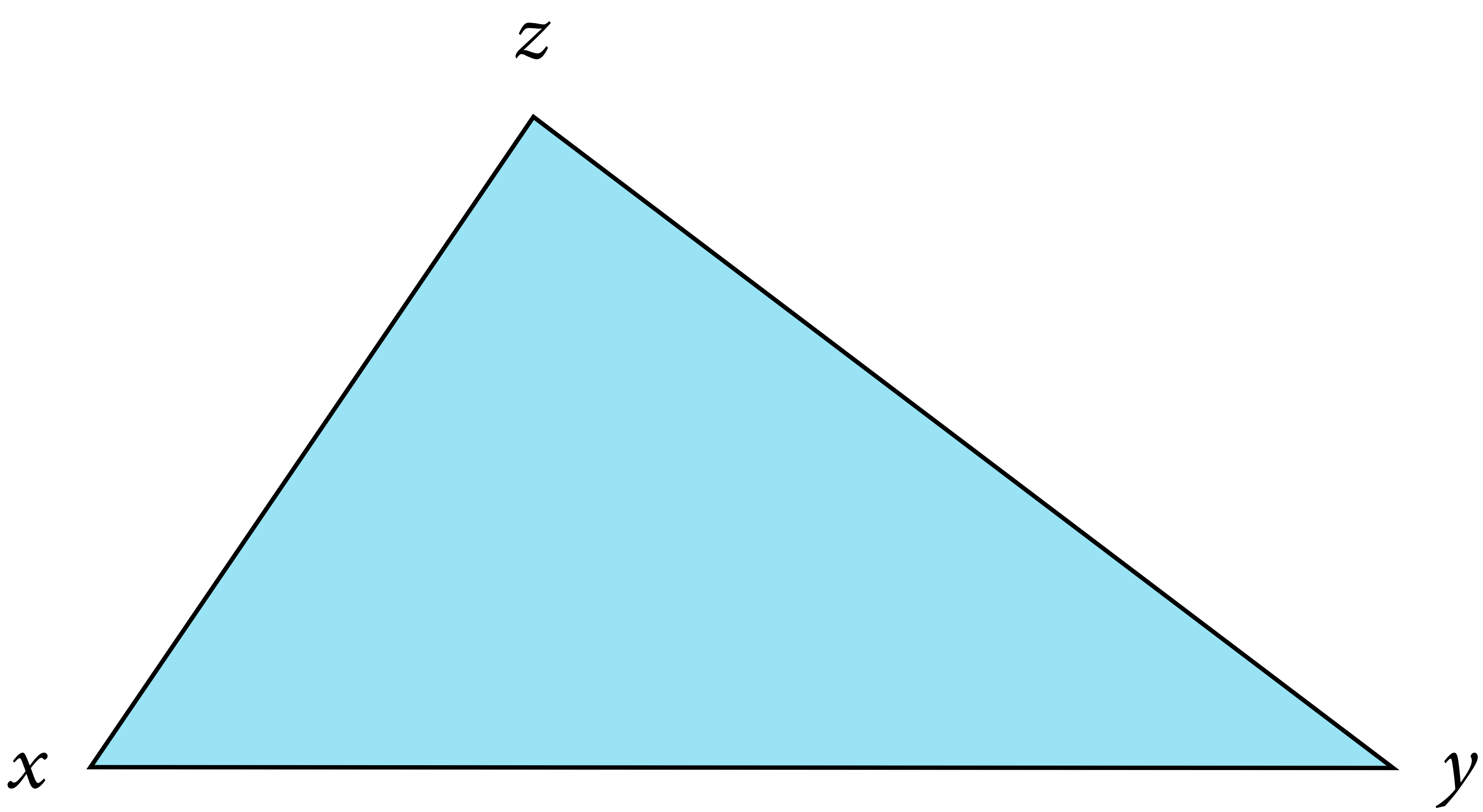
# Distance functions

A **distance function** on a set $X$ is a function $d : X \times X \to [0, \infty)$ that satisfies the following properties:

1. $d(x, y) \geq 0$ (non-negativity)
2. $d(x, y) = 0 \iff x = y$ (identity of indiscernibles)
3. $d(x, y) = d(y, x)$ (symmetry)
4. $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality)

Such a function establishes a **metric** on $X$.

# Triangle inequality




$$d(x, y) \leq d(x, z) + d(z, y)$$

# Accuracy for symmetric binary vectors

| | vector w 1 | vector w 0 |
|---|---|---|
| vector v 1 | $a$ | $b$ |
| vector v 0 | $c$ | $d$ |

$$\mathrm{sim}(\boldsymbol{v}, \boldsymbol{w}) = \frac{a + d}{a + b + c + d}$$

The corresponding distance function is sometimes called error rate.

# Jaccard similarity for asymmetric binary vectors

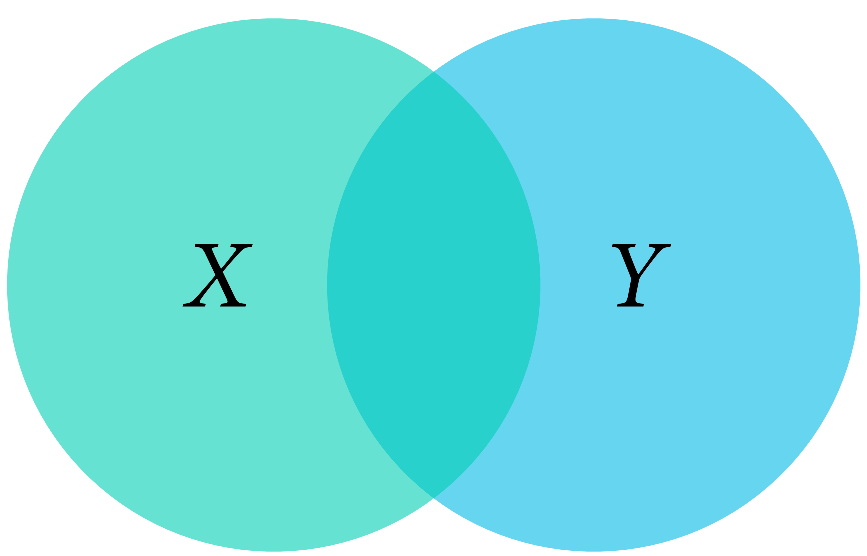
| | vector w 1 | vector w 0 |
|---|---|---|
| vector v 1 | $a$ | $b$ |
| vector v 0 | $c$ | $d$ |

$$\mathrm{sim}(\boldsymbol{v}, \boldsymbol{w}) = \frac{a}{a + b + c}$$

# Jaccard similarity for sets

Under the set view, Jaccard similarity measures the number of shared elements in two sets, relative to the size of the two sets:

$$\mathrm{sim}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$



# Cosine similarity for general vectors

- **Cosine similarity** measures the cosine of the angle between two non-zero vectors, independently of the length of the vectors.

$$\cos(\boldsymbol{v}, \boldsymbol{w}) = \frac{\boldsymbol{v}}{|\boldsymbol{v}|} \cdot \frac{\boldsymbol{w}}{|\boldsymbol{w}|} = \frac{\boldsymbol{v} \cdot \boldsymbol{w}}{|\boldsymbol{v}||\boldsymbol{w}|} = \frac{\sum_{i=1}^{d} v_i w_i}{\sqrt{\sum_{i=1}^{d} v_i^2}\sqrt{\sum_{i=1}^{d} w_i^2}}$$

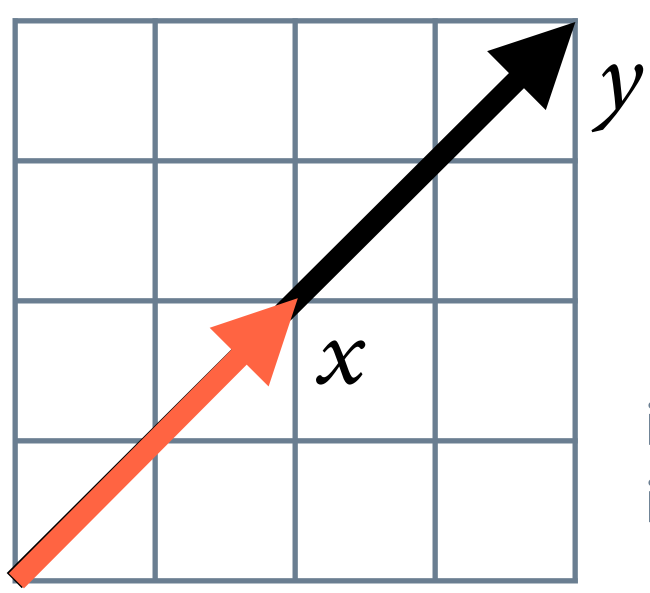
- Cosine similarity can take negative values. However, when restricted to non-negative vectors, it is in the range [0, 1].

  frequency vectors, tf–idf weights

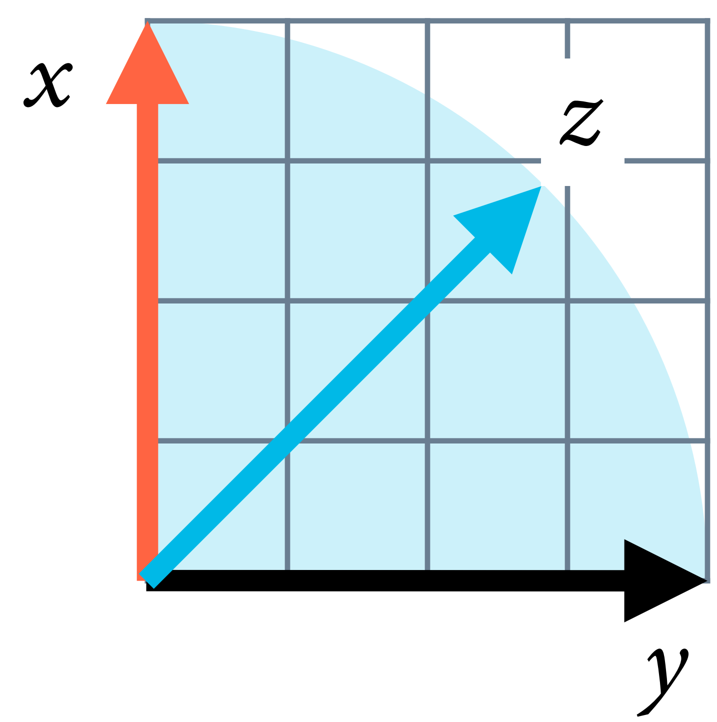
# Cosine distance

- The term **cosine distance** is often used for the complement of cosine similarity, 1 – cos($\boldsymbol{vw}$).

- However, this 'distance' is not a proper metric, as it violates the identity of indiscernibles and the triangle inequality.




identity of indiscernibles




triangle inequality

# Pointwise mutual information for random variables

- **Pointwise mutual information** measures the distributional similarity between outcomes of two discrete random variables.

$$\mathrm{pmi}(x, y) = \log \frac{p(x, y)}{p(x)\,p(y)}$$

- In the context of text clustering, PMI is frequently used to measure the associative strength between word occurrences.

  *ice cream*, *ice hockey* > *the cream*, *bad hockey*

# This lecture

- Introduction to text clustering
- Similarity measures
- An overview of hard clustering methods
- Evaluation of hard clustering
- Soft clustering: Topic models

# An overview of hard clustering methods

# Hierarchical clustering

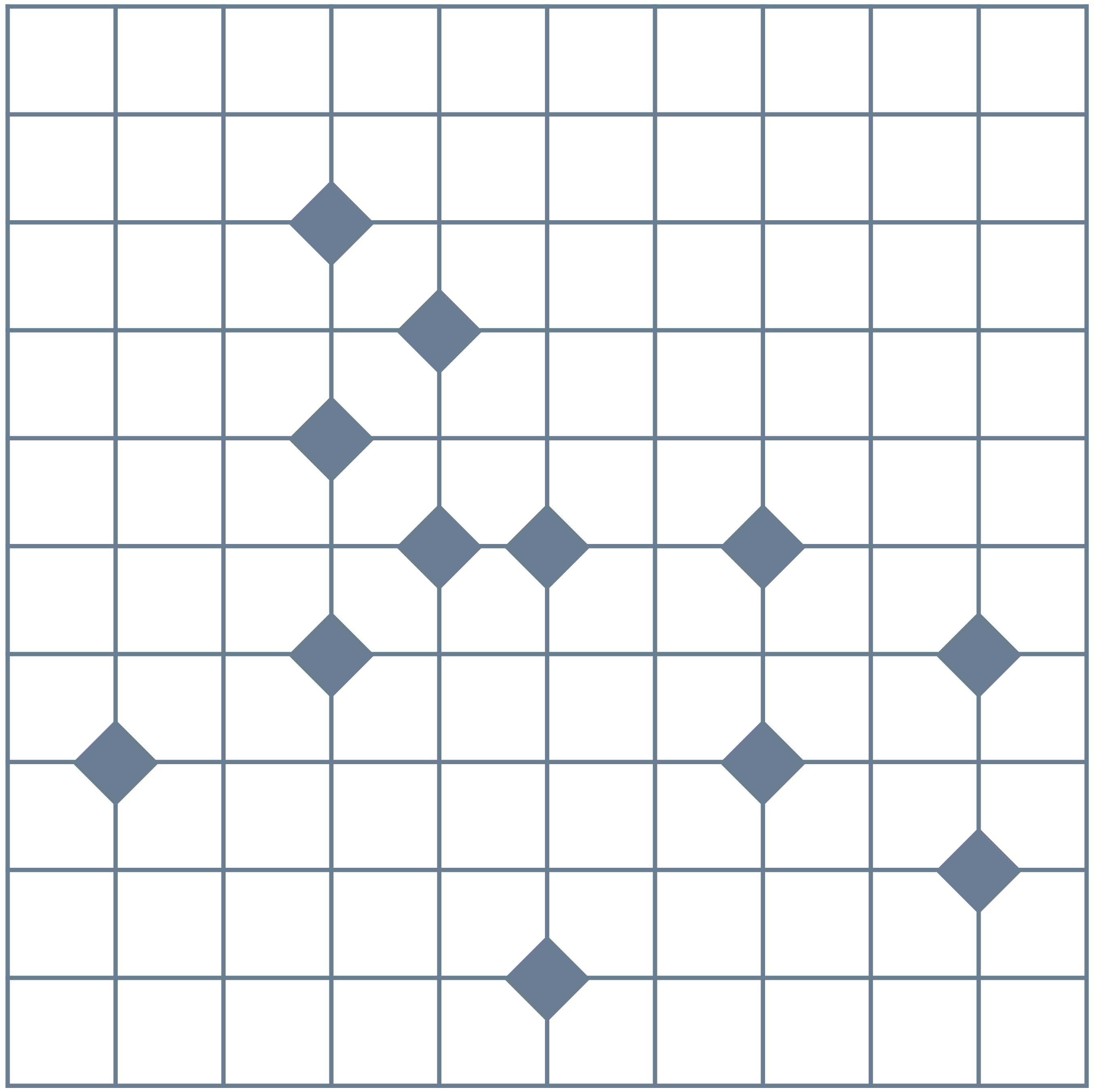
The term **hierarchical clustering** refers to clustering methods that seek to build a hierarchy of clusters. There are two kinds:

- **Agglomerative.** Each document starts in its own cluster. Hierarchy is created by merging pairs of clusters.
- **Divisive clustering.** All documents start in one cluster. Hierarchy is created by splitting clusters recursively.

# Agglomerative hierarchical clustering

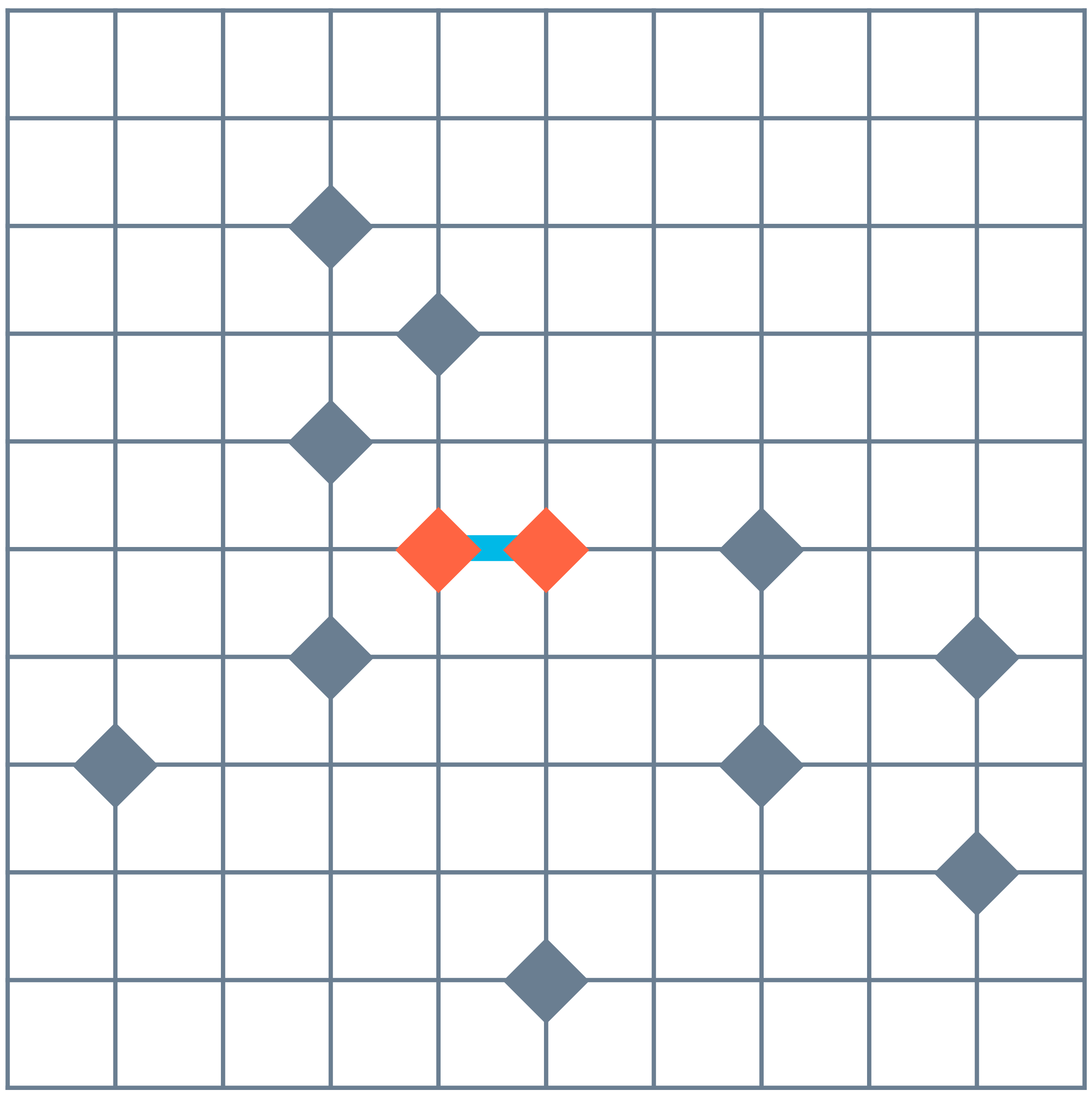
- Algorithm: While there is more than one cluster, find the two most similar clusters and merge them.
- Stop the process when all documents belong to one cluster, or when the desired number of clusters is found.
- Note that this algorithm requires a measure of similarity between *clusters*, rather than between individual documents.

# Hierarchical agglomerative clustering



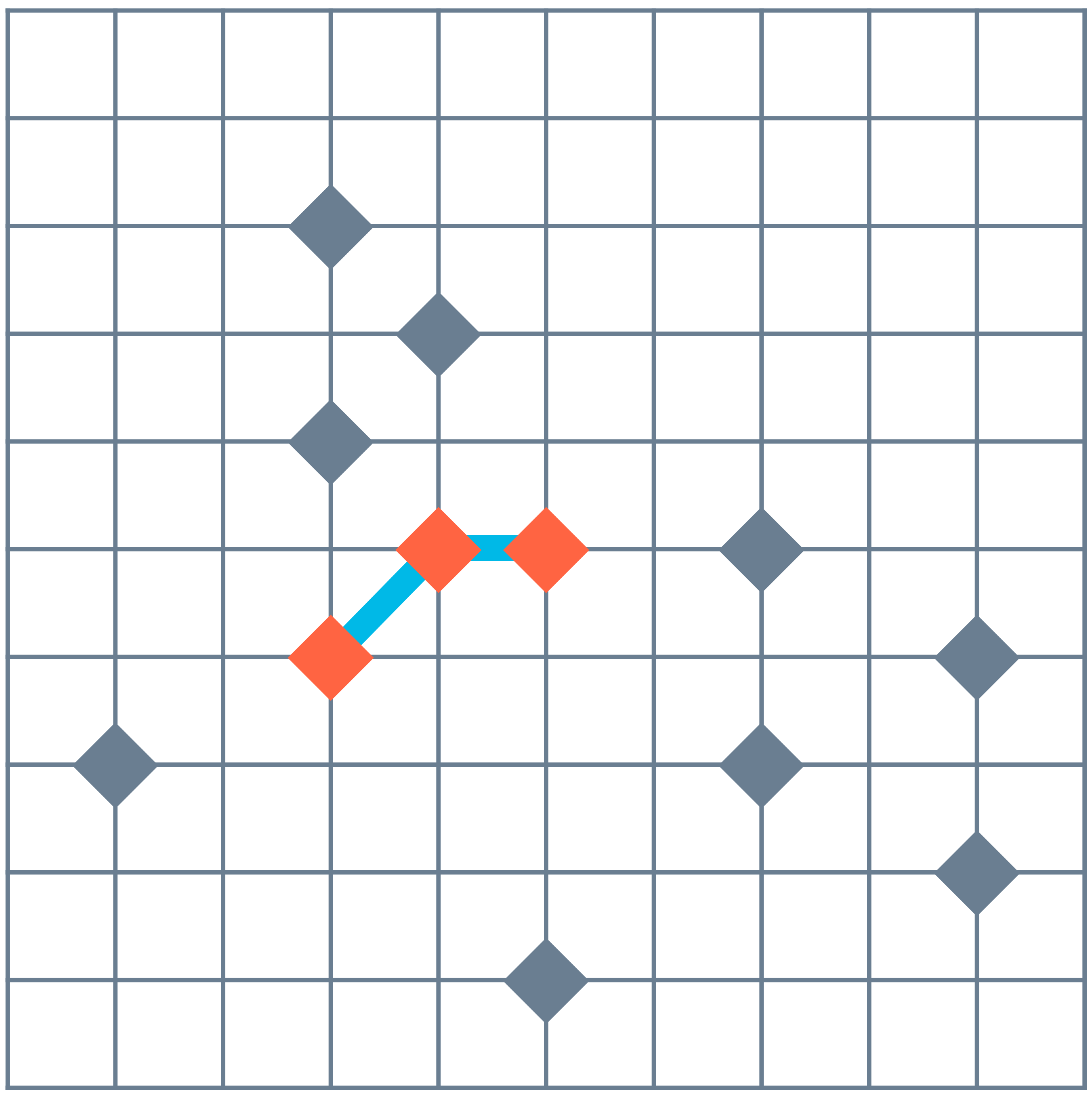
Initially, each document lives in its own cluster.

# Hierarchical agglomerative clustering



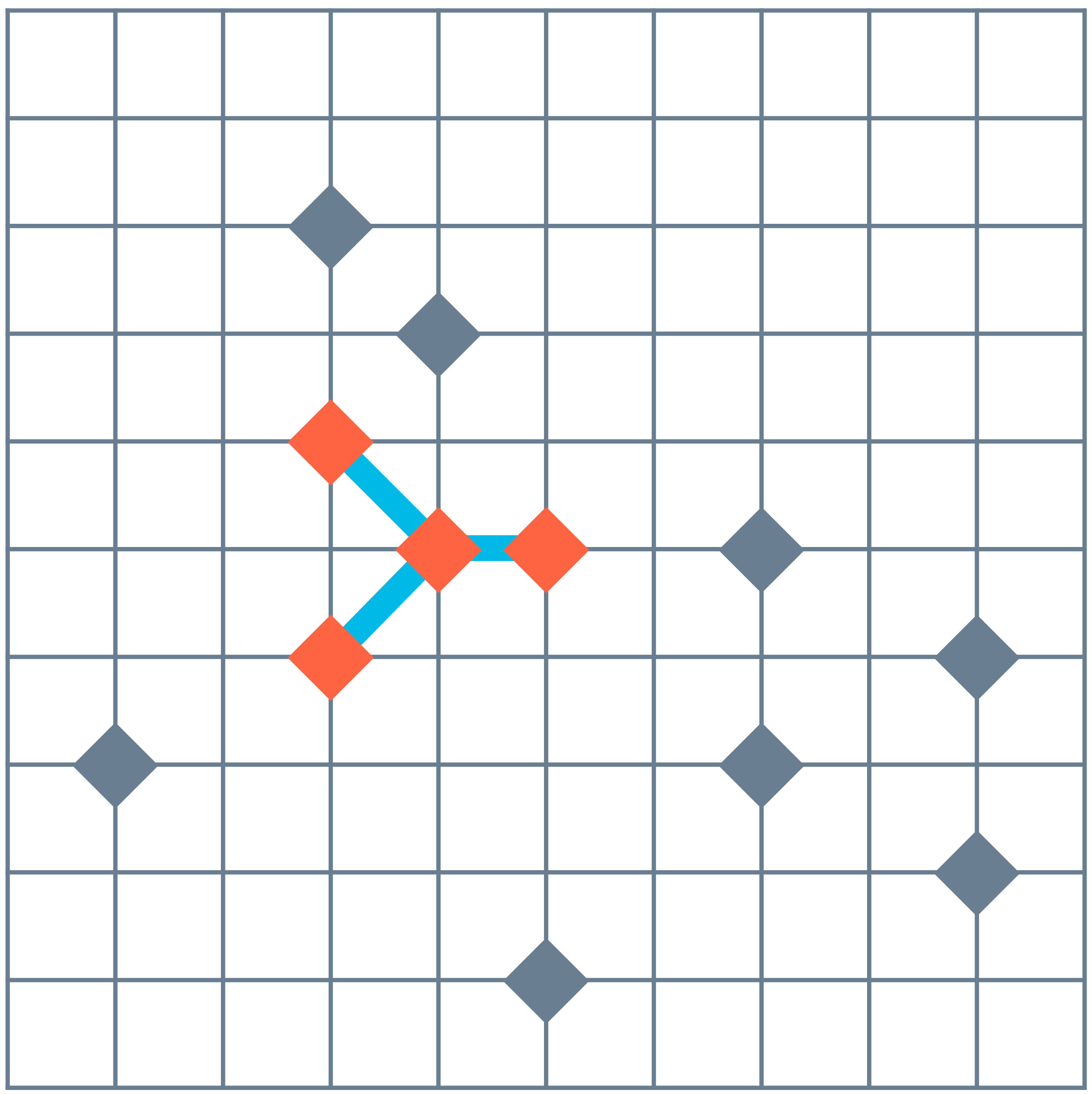
Find the two most similar clusters and merge them.

# Hierarchical agglomerative clustering



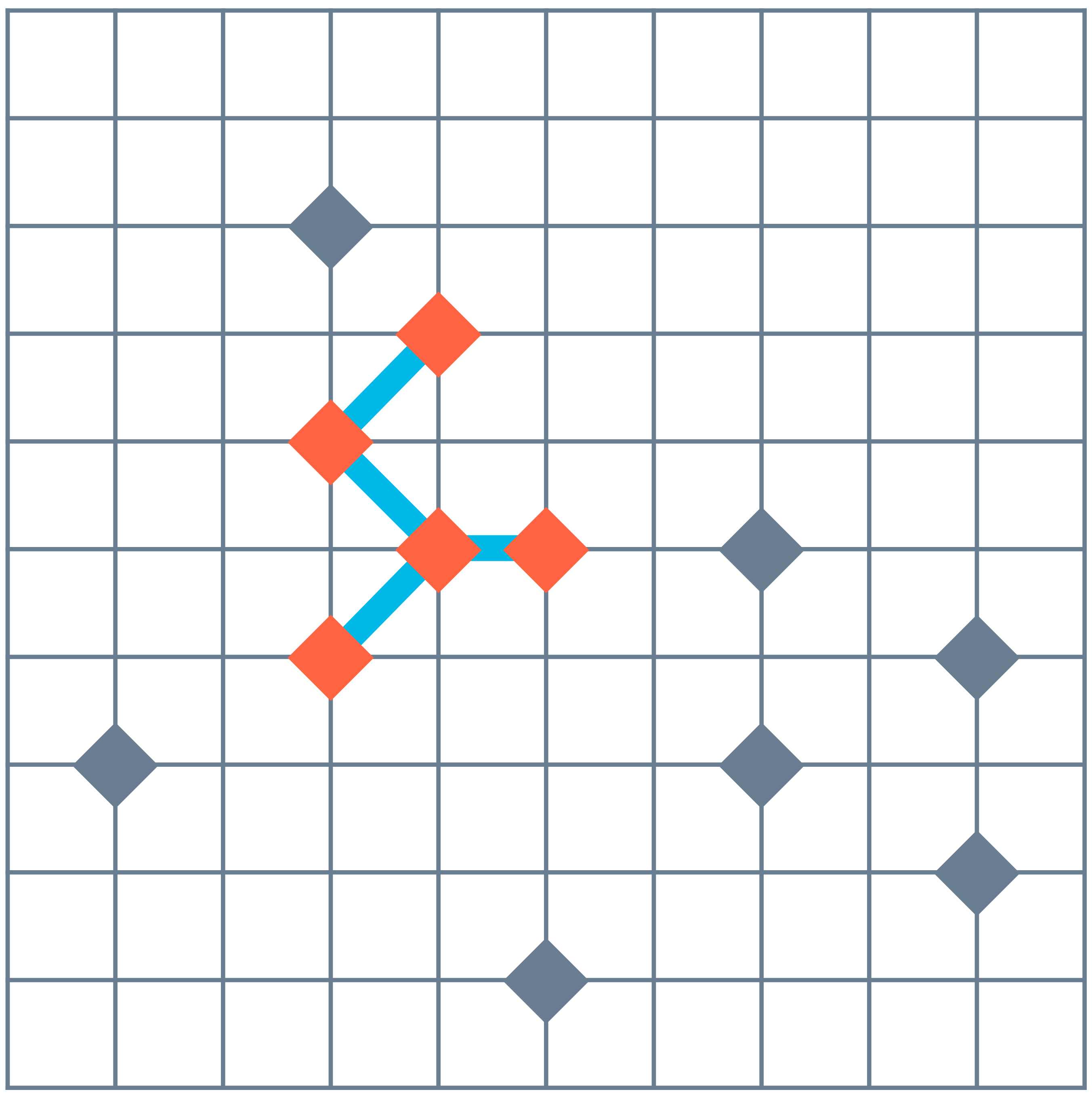
Find the two most similar clusters and merge them.

# Hierarchical agglomerative clustering



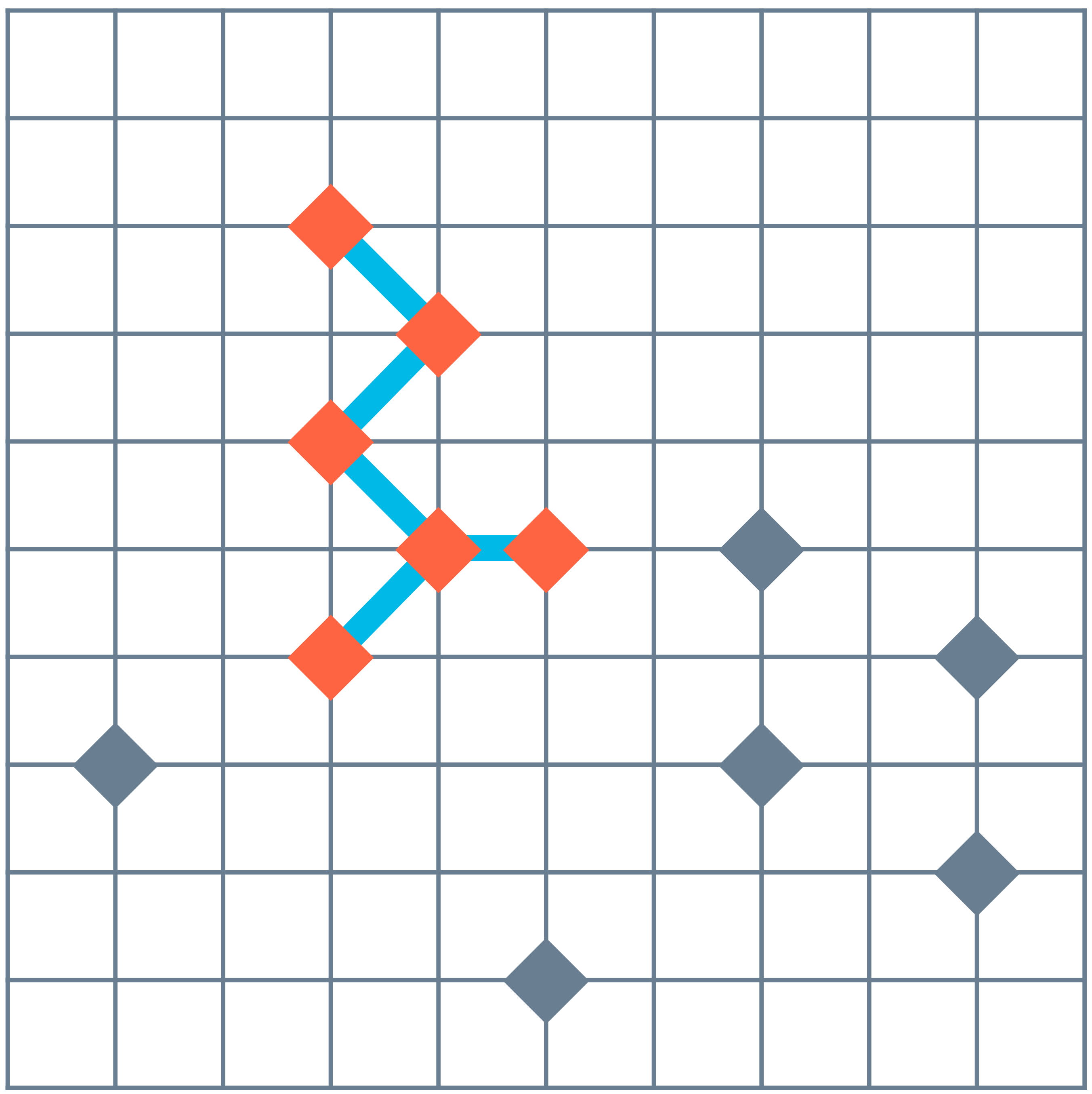
Find the two most similar clusters and merge them.

# Hierarchical agglomerative clustering



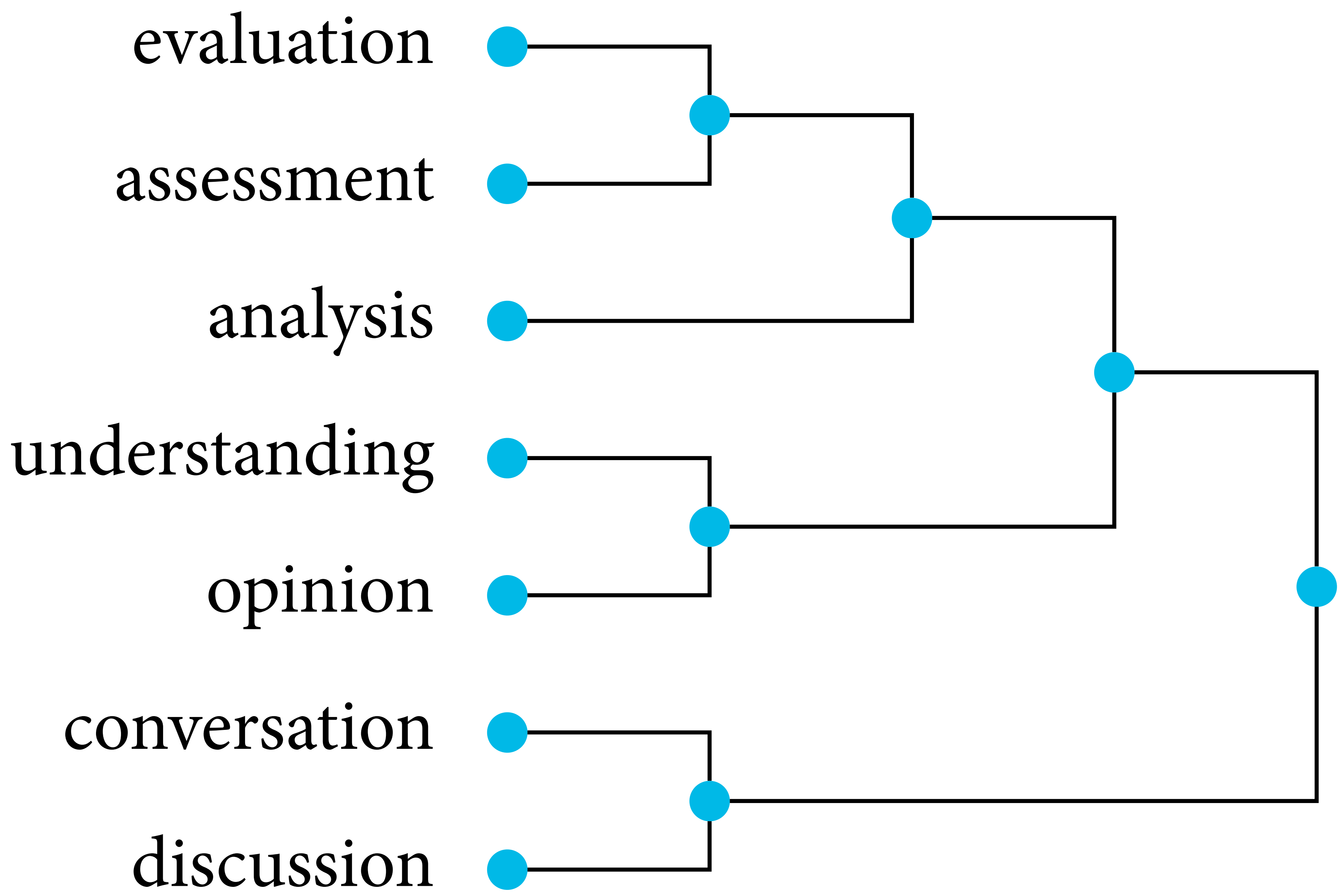
Find the two most similar clusters and merge them.

# Hierarchical agglomerative clustering



Find the two most similar clusters and merge them.

# Dendrograms show how clustered are merged




Source: Brown et al. (1992)

# Linkage criteria

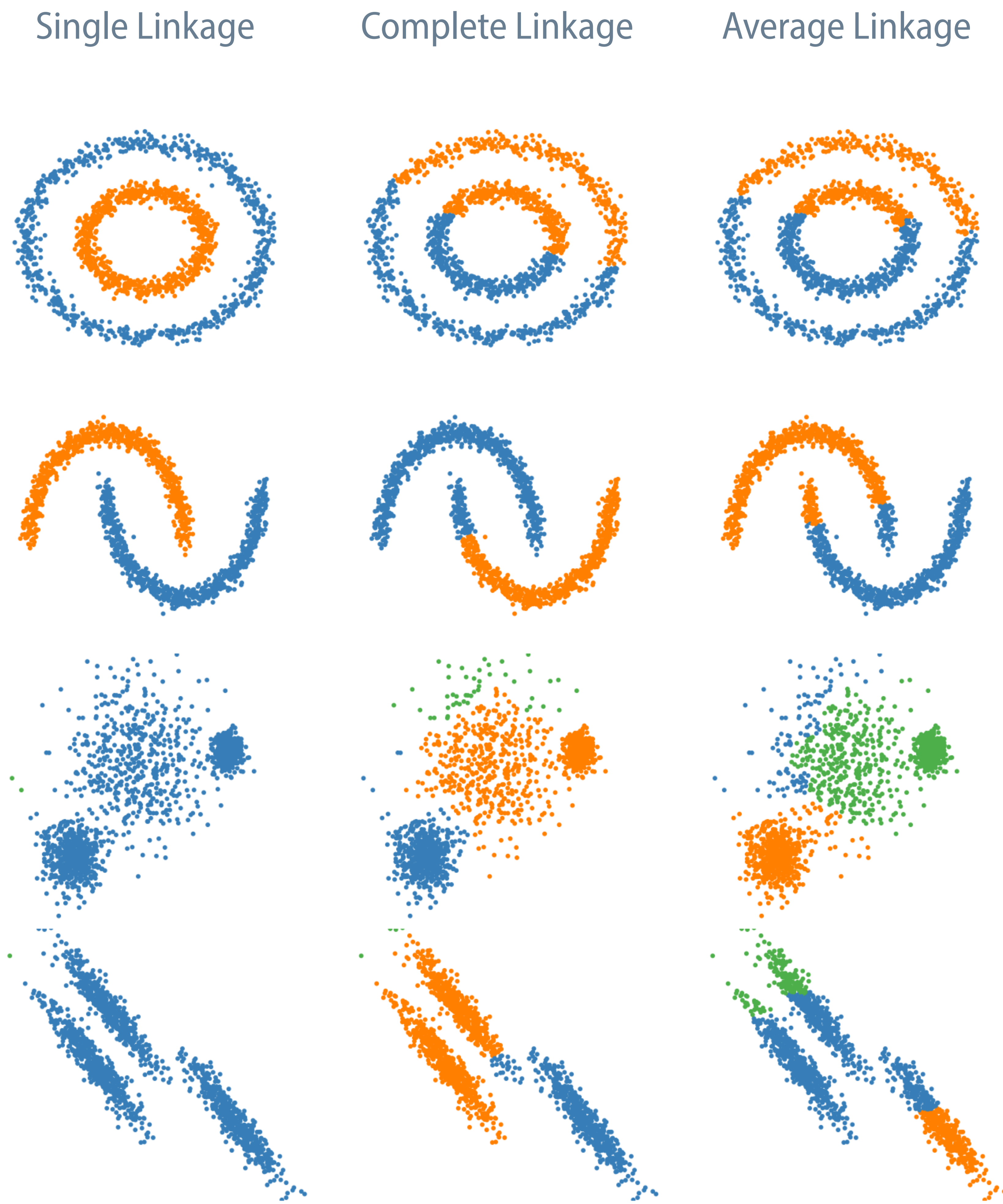
- **Single-link** merges two clusters with the smallest *minimum* distance. This criterion yields 'loose' clusters.

  To trigger a merge, it suffices to find one document pair with high similarity.

- **Complete-link** merges two clusters with the smallest *maximum* distance. This criterion yields 'compact' clusters.

  A merge is only triggered if there are many pairs with high similarity.

- **Average-link** merges two clusters with the smallest *average* distance. This is less sensitive to outliers than the other two.




Source: Scikit-learn

# K-means clustering

- The ***k*-means algorithm** aims to partition a document collection into *k* clusters, minimising within-cluster variance in distance.

  distance variance = squared Euclidean distances

- Each document, represented by its vector, will be put into the cluster with the nearest centroid (mean).

# Centroids and medoids

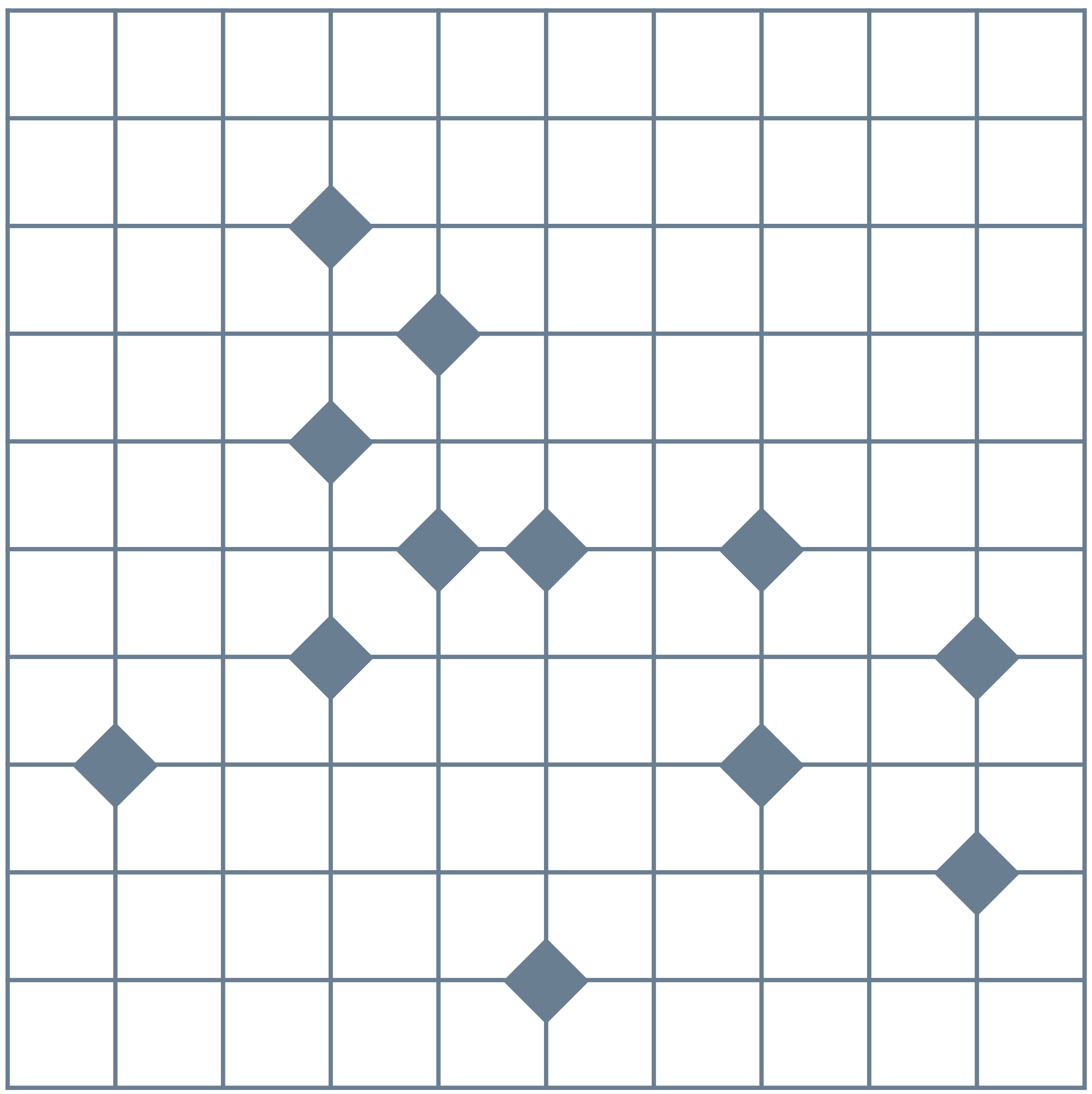
- The **centroid** of a cluster is the arithmetic mean of the document vectors in the cluster.

  not necessarily the vector of an actual document

- The **medoid** of a cluster is a vector in the cluster whose average distance to all the other vectors is minimal.

  not the same as a geometric median

# K-means clustering

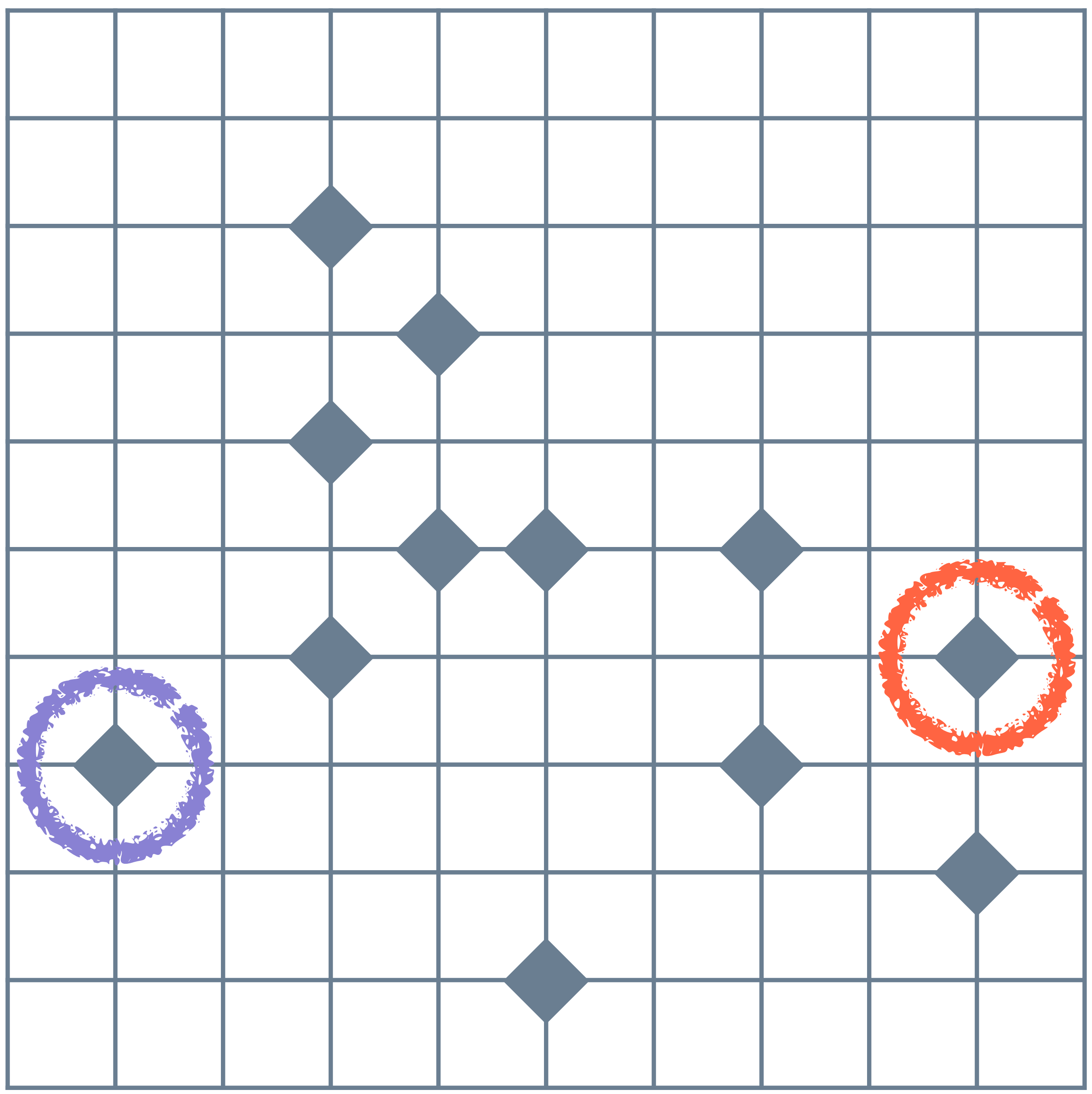


**Step 1:** Randomly choose $k$ documents as initial centroids.

**Step 2:** Assign each document to the closest centroid.

**Step 3:** Update the cluster centroids.

# K-means clustering

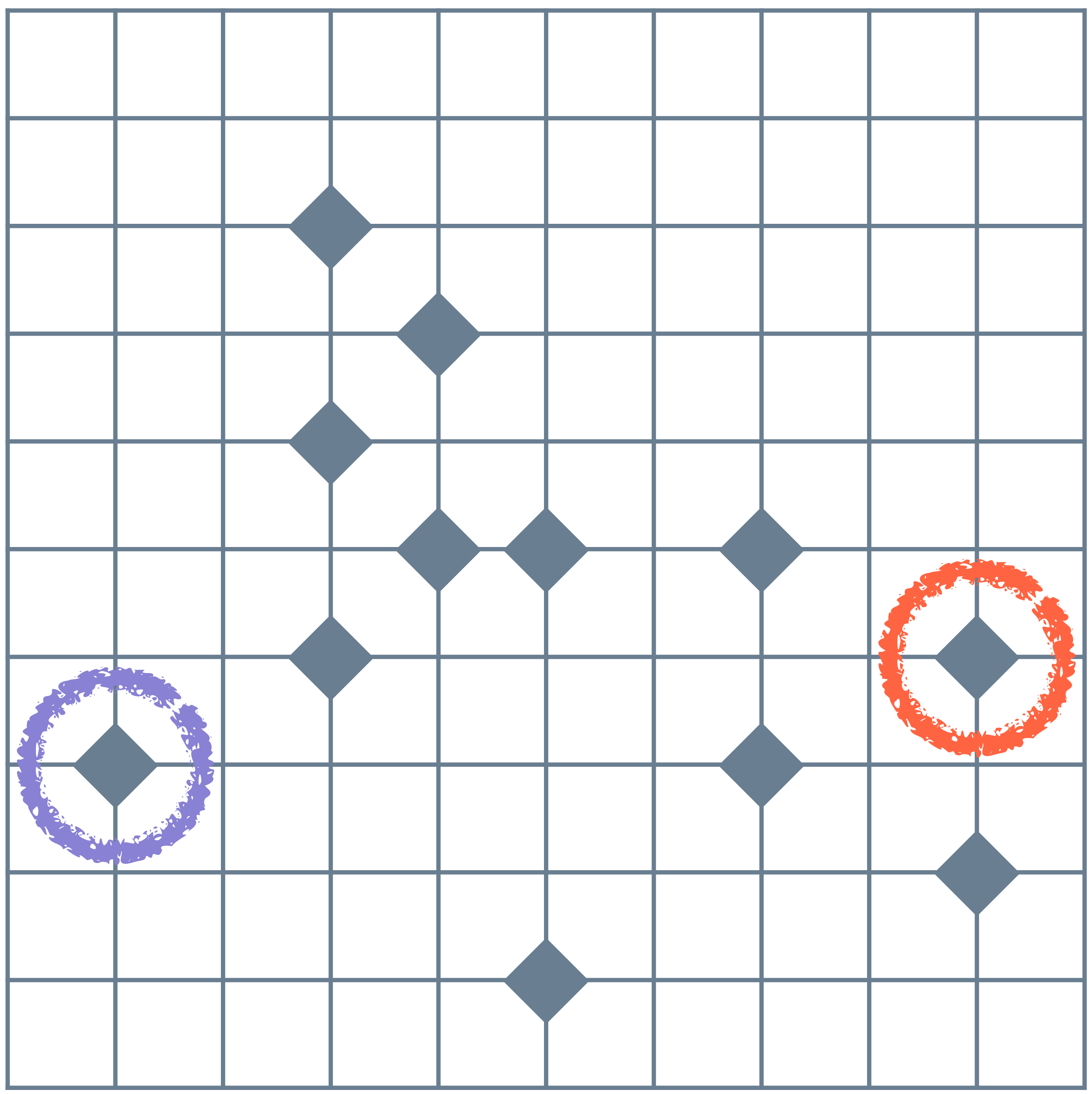


**Step 1:** Randomly choose $k$ documents as initial centroids.

**Step 2:** Assign each document to the closest centroid.

**Step 3:** Update the cluster centroids.

# K-means clustering

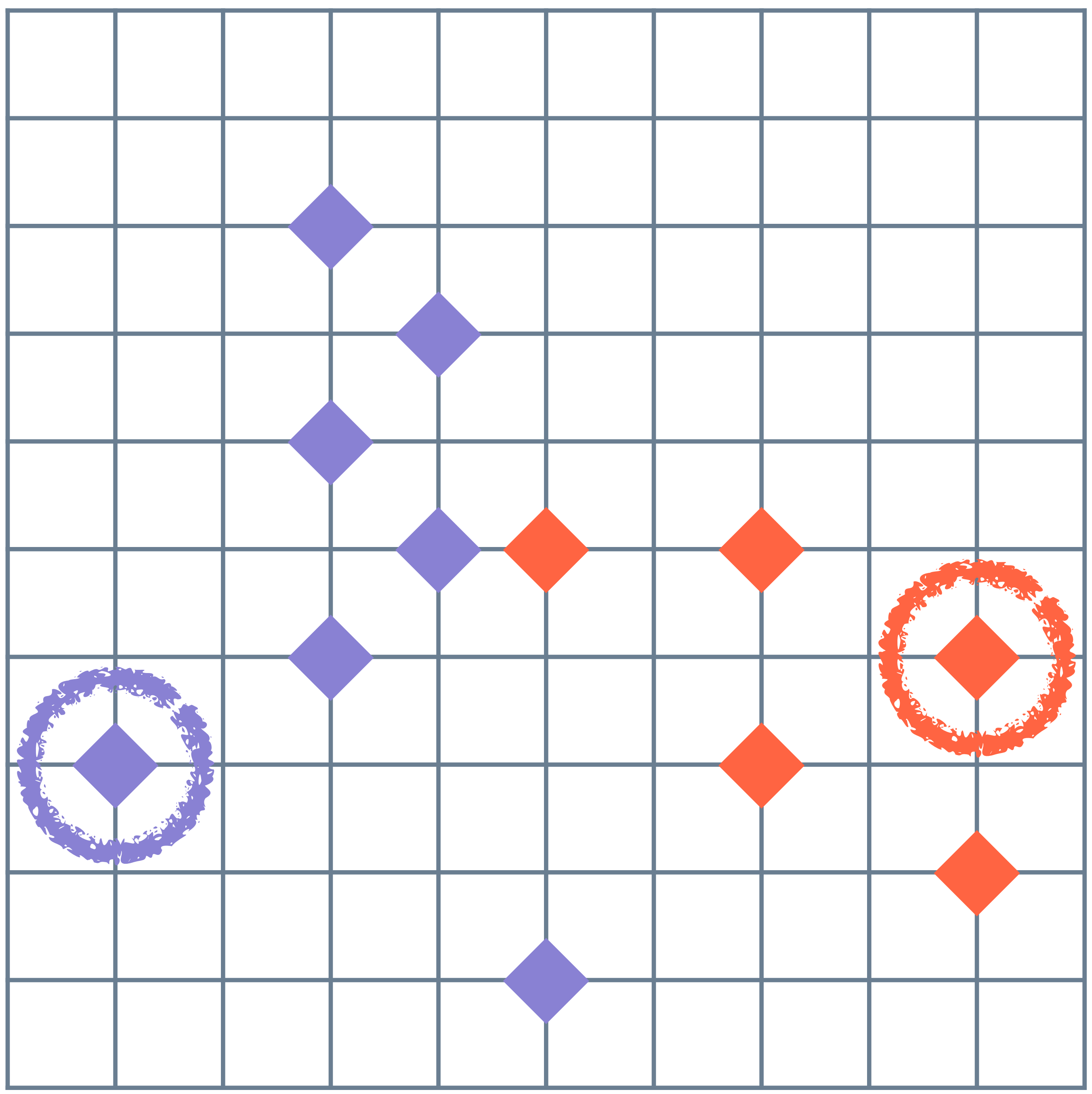


**Step 1:** Randomly choose *k* documents as initial centroids.

**Step 2:** Assign each document to the closest centroid.

**Step 3:** Update the cluster centroids.

# K-means clustering

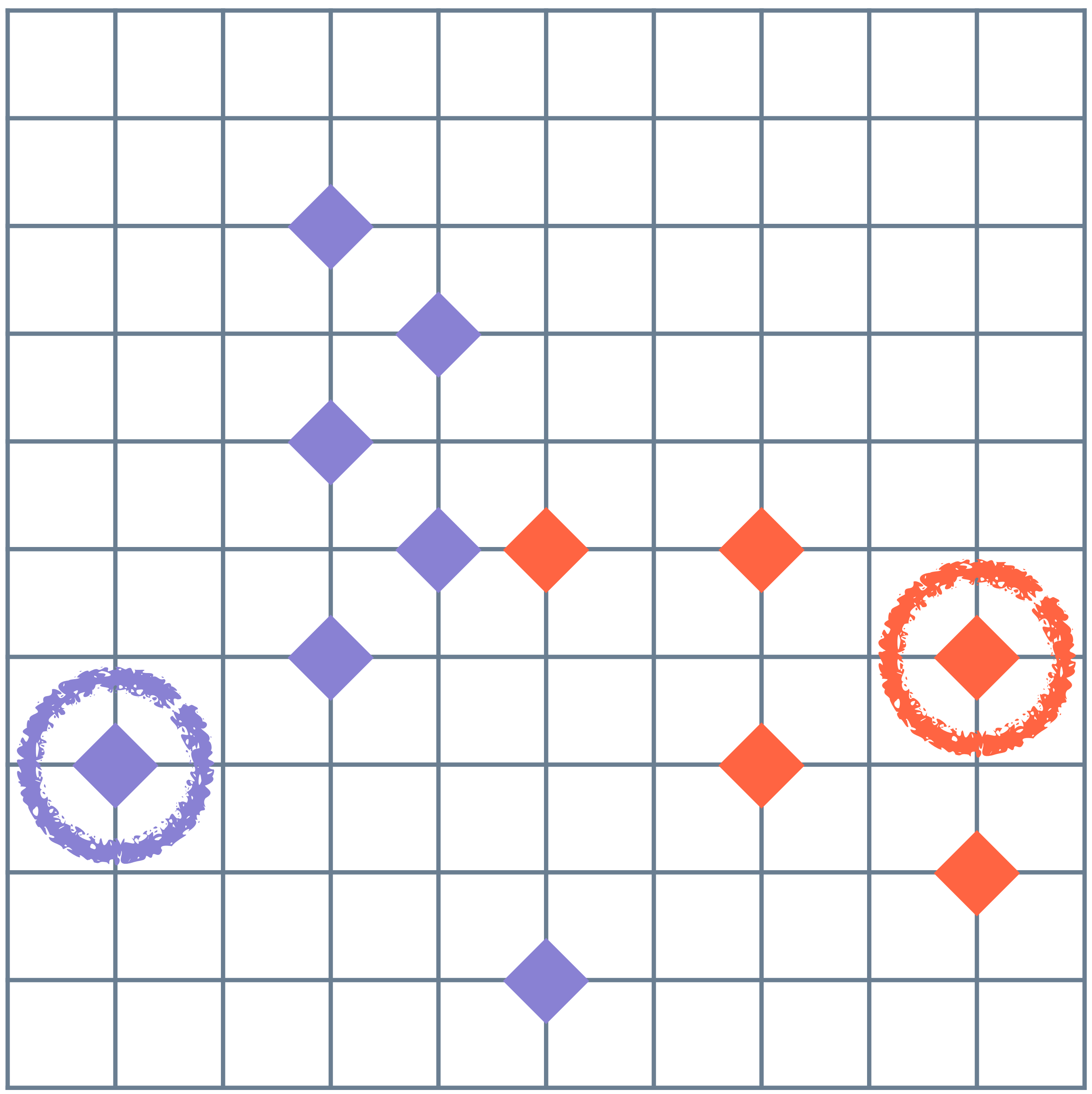


**Step 1:** Randomly choose *k* documents as initial centroids.

**Step 2:** Assign each document to the closest centroid.

**Step 3:** Update the cluster centroids.

# K-means clustering

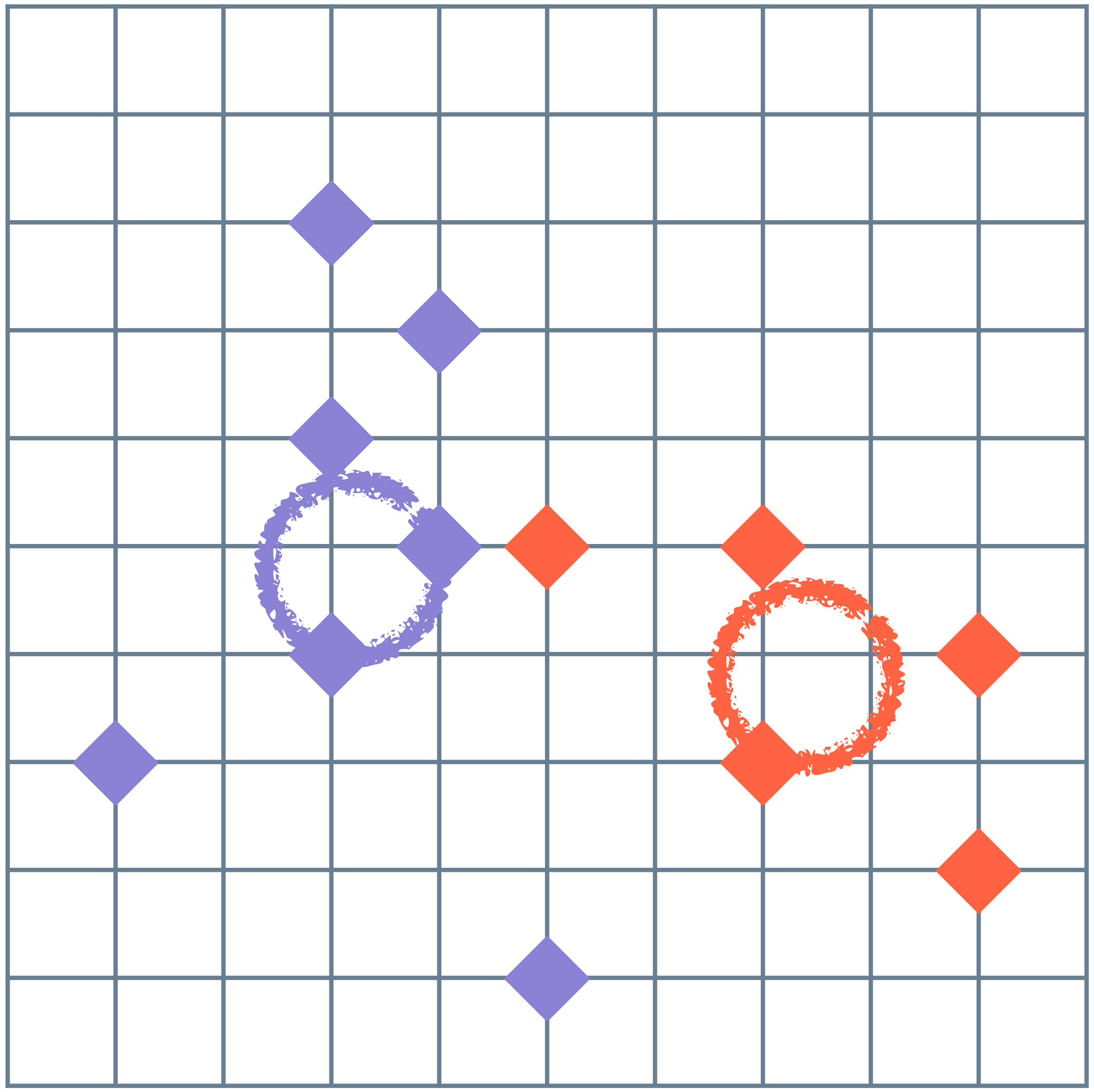


**Step 1:** Randomly choose *k* documents as initial centroids.

**Step 2:** Assign each document to the closest centroid.

**Step 3:** Update the cluster centroids.

# K-means clustering

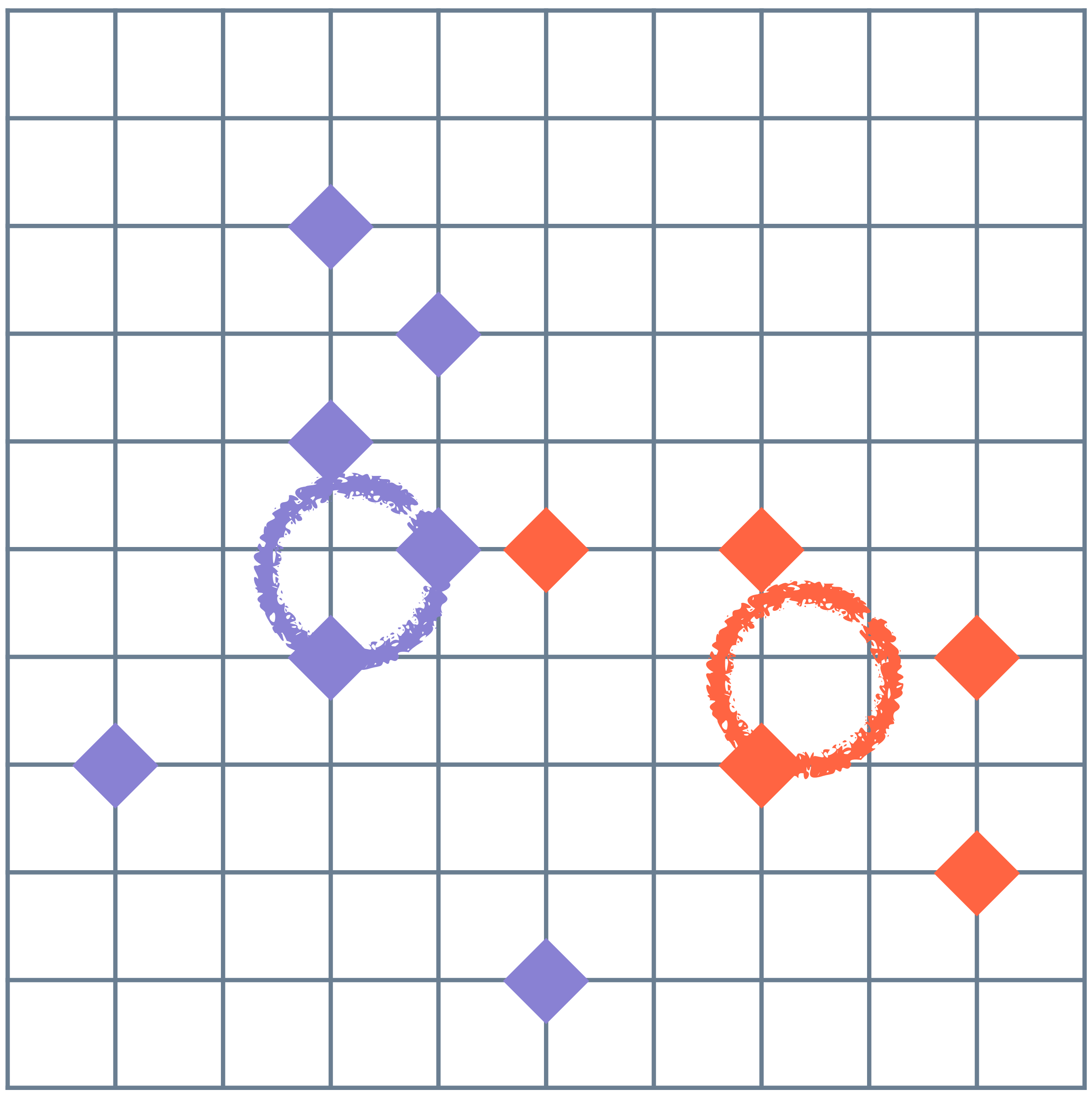


**Step 1:** Randomly choose *k* documents as initial centroids.

**Step 2:** Assign each document to the closest centroid.

**Step 3:** Update the cluster centroids.

# K-means clustering

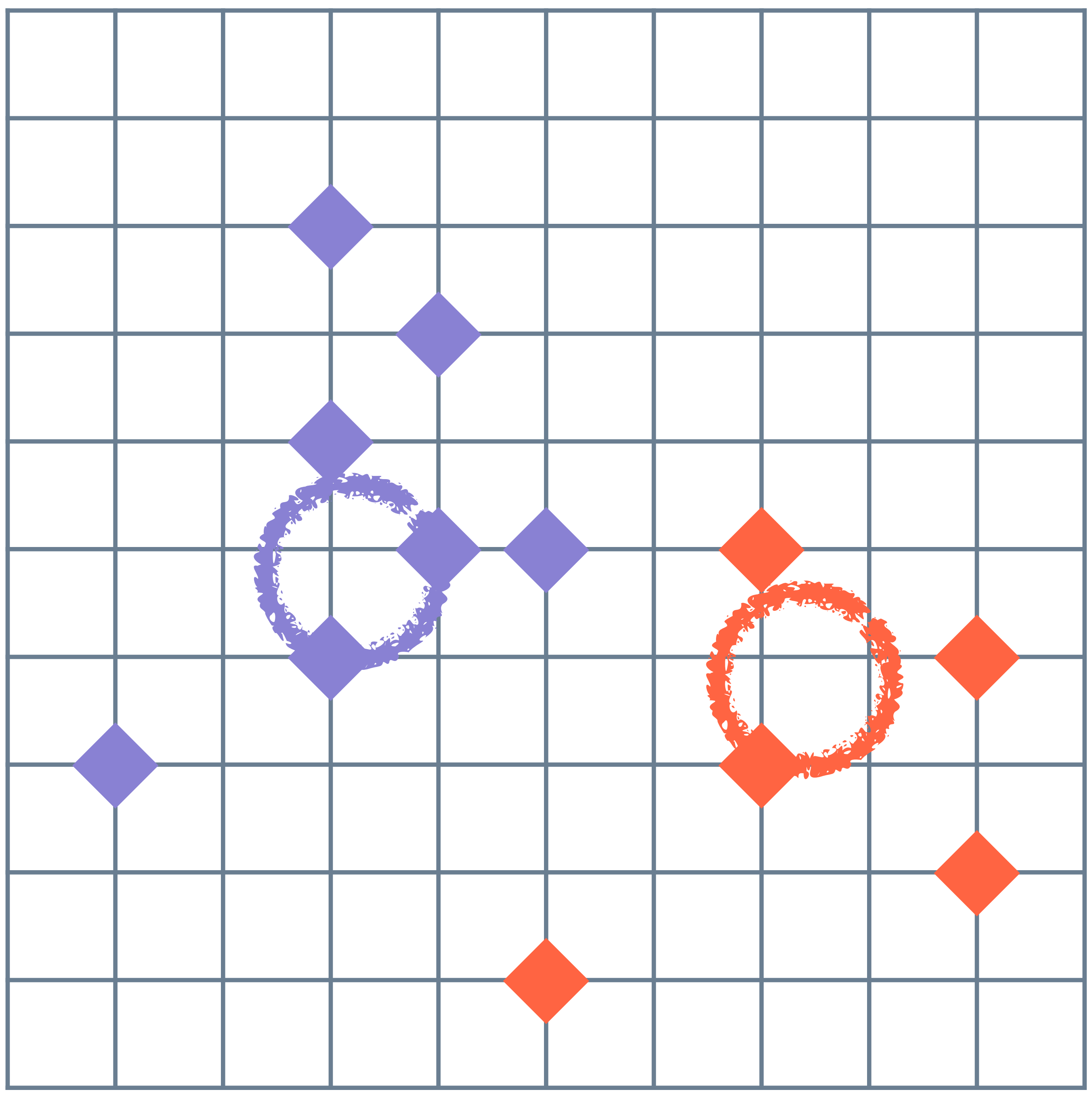


**Step 1:** Randomly choose *k* documents as initial centroids.

**Step 2:** Assign each document to the closest centroid.

**Step 3:** Update the cluster centroids.

# K-means clustering

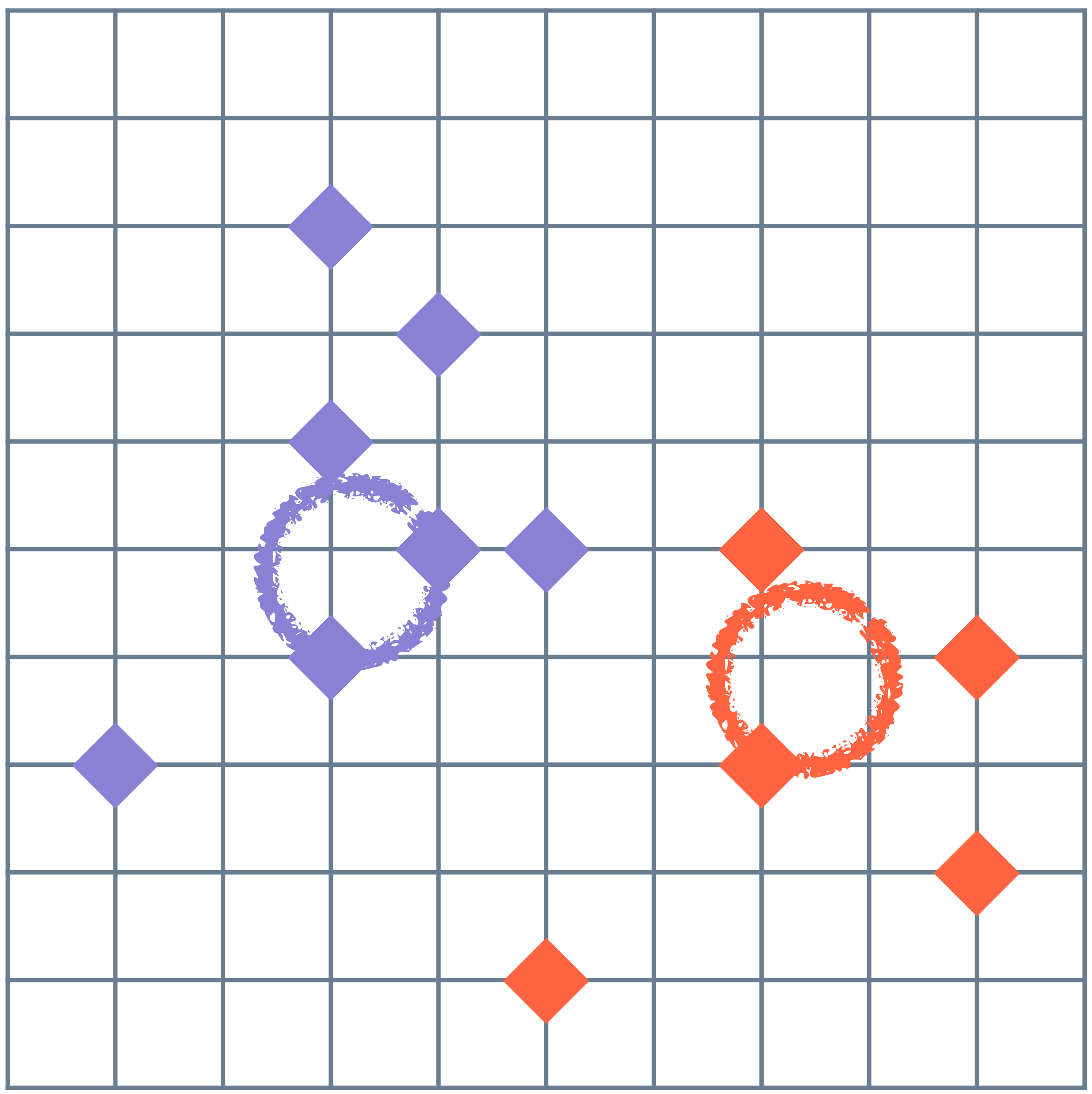


**Step 1:** Randomly choose *k* documents as initial centroids.

**Step 2:** Assign each document to the closest centroid.

**Step 3:** Update the cluster centroids.

# K-means clustering

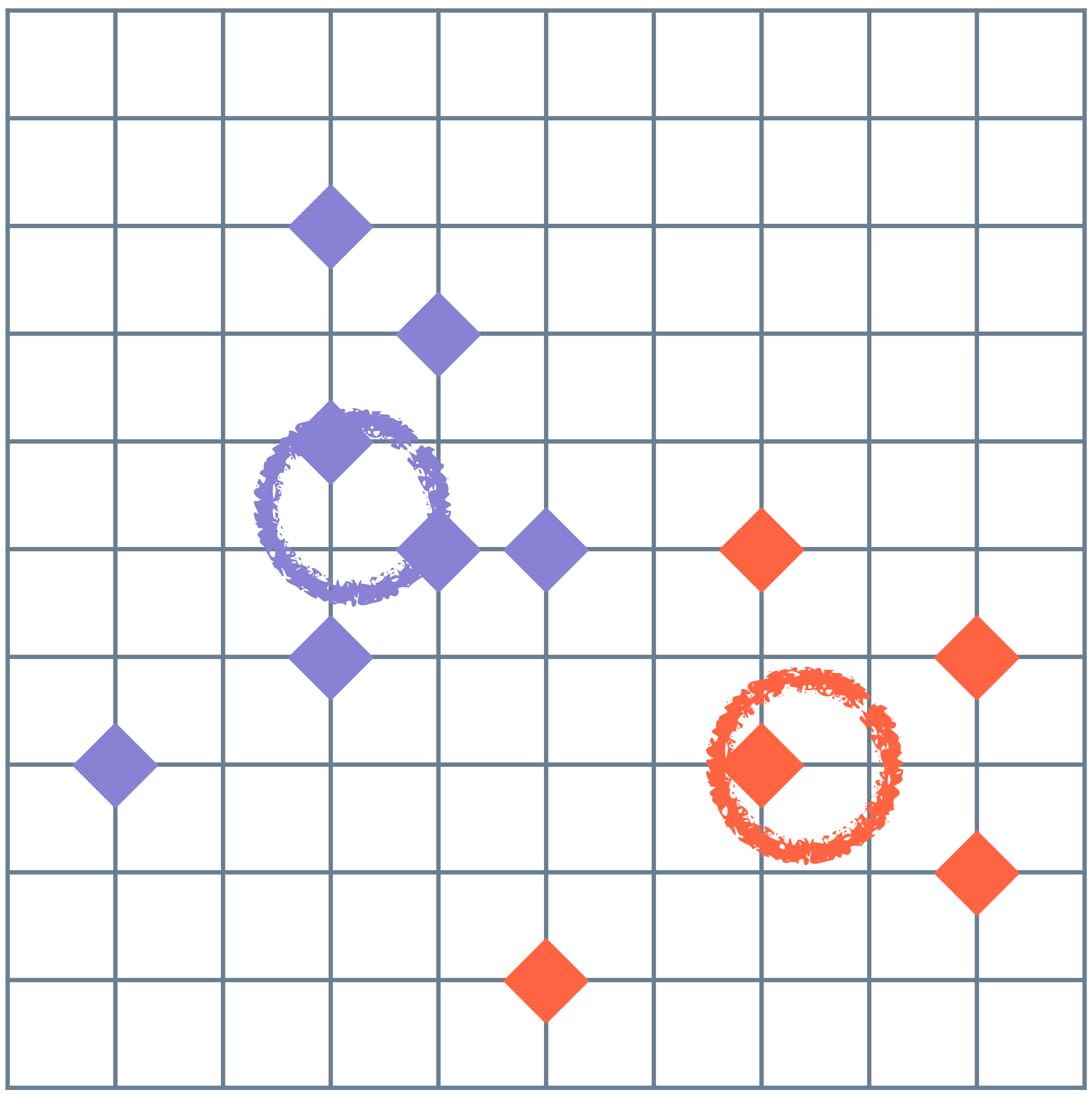


**Step 1:** Randomly choose *k* documents as initial centroids.

**Step 2:** Assign each document to the closest centroid.

**Step 3:** Update the cluster centroids.

# K-means clustering

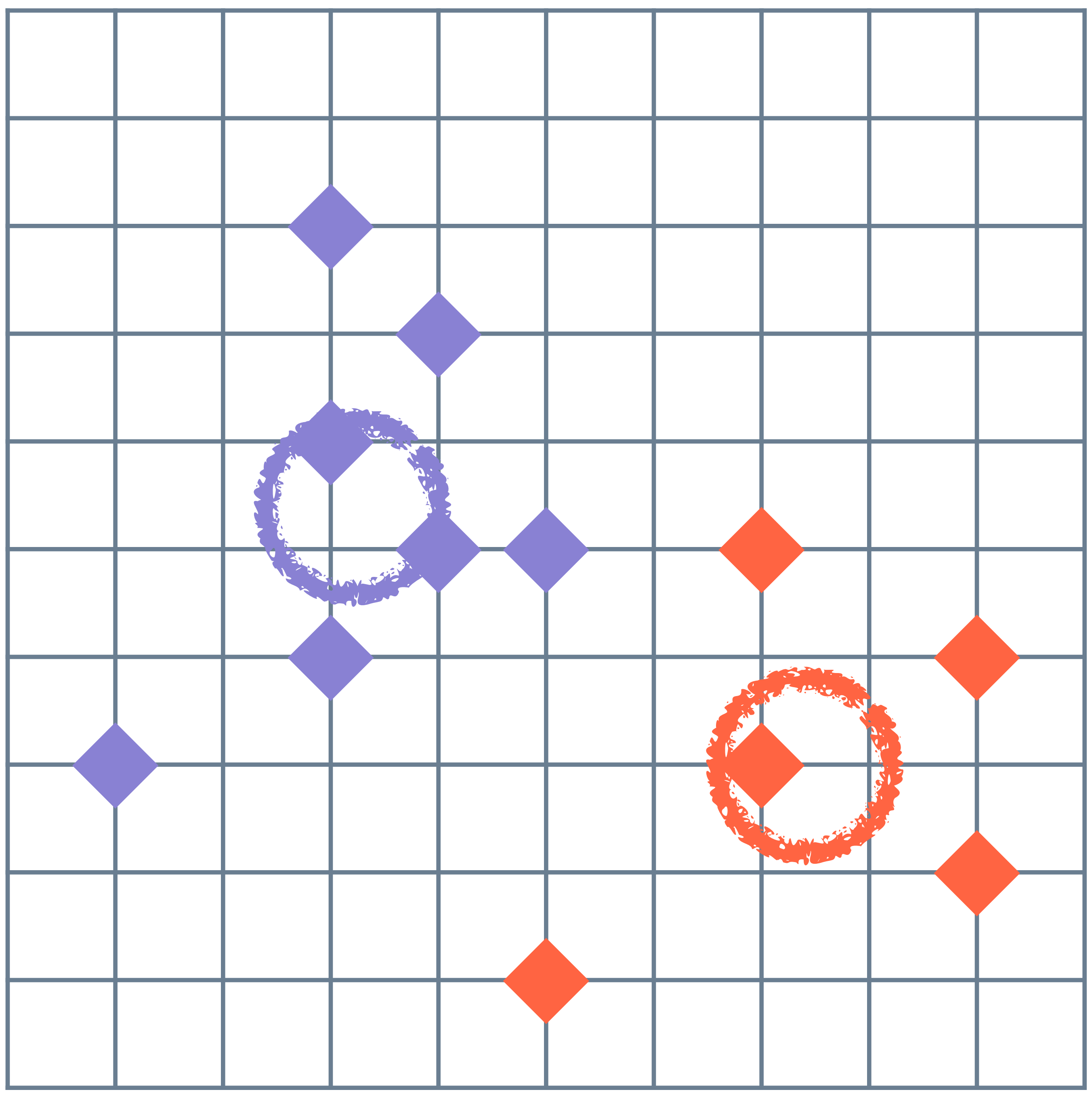


**Step 1:** Randomly choose *k* documents as initial centroids.

**Step 2:** Assign each document to the closest centroid.

**Step 3:** Update the cluster centroids.

# K-means clustering



At this point, each document belongs to the cluster with the closest centroid, and the algorithm terminates.

# Issues with the k-means algorithm

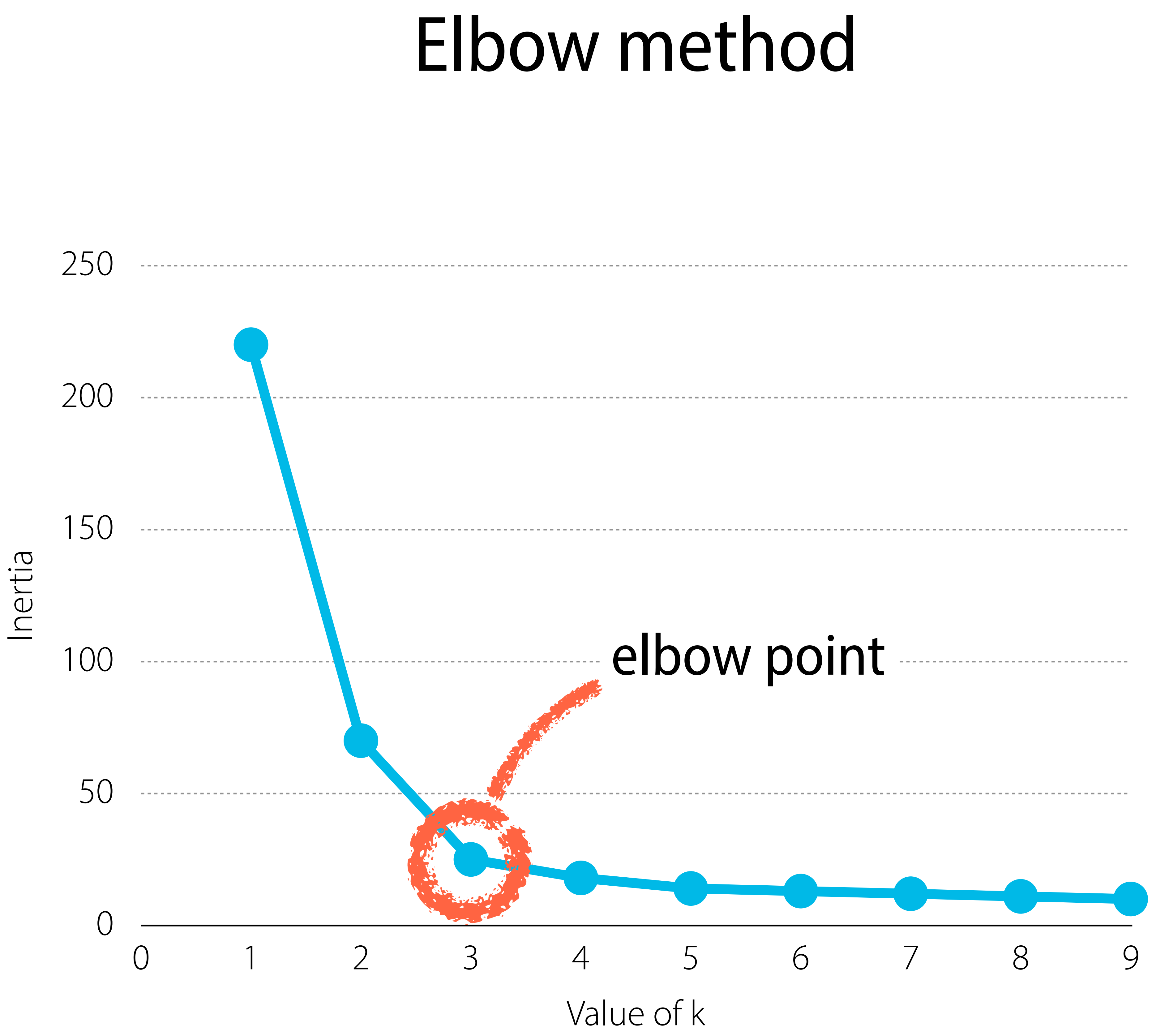
- The $k$-means algorithm always converges, but there is no guarantee that it finds a global optimum.

  Solution: random restarts

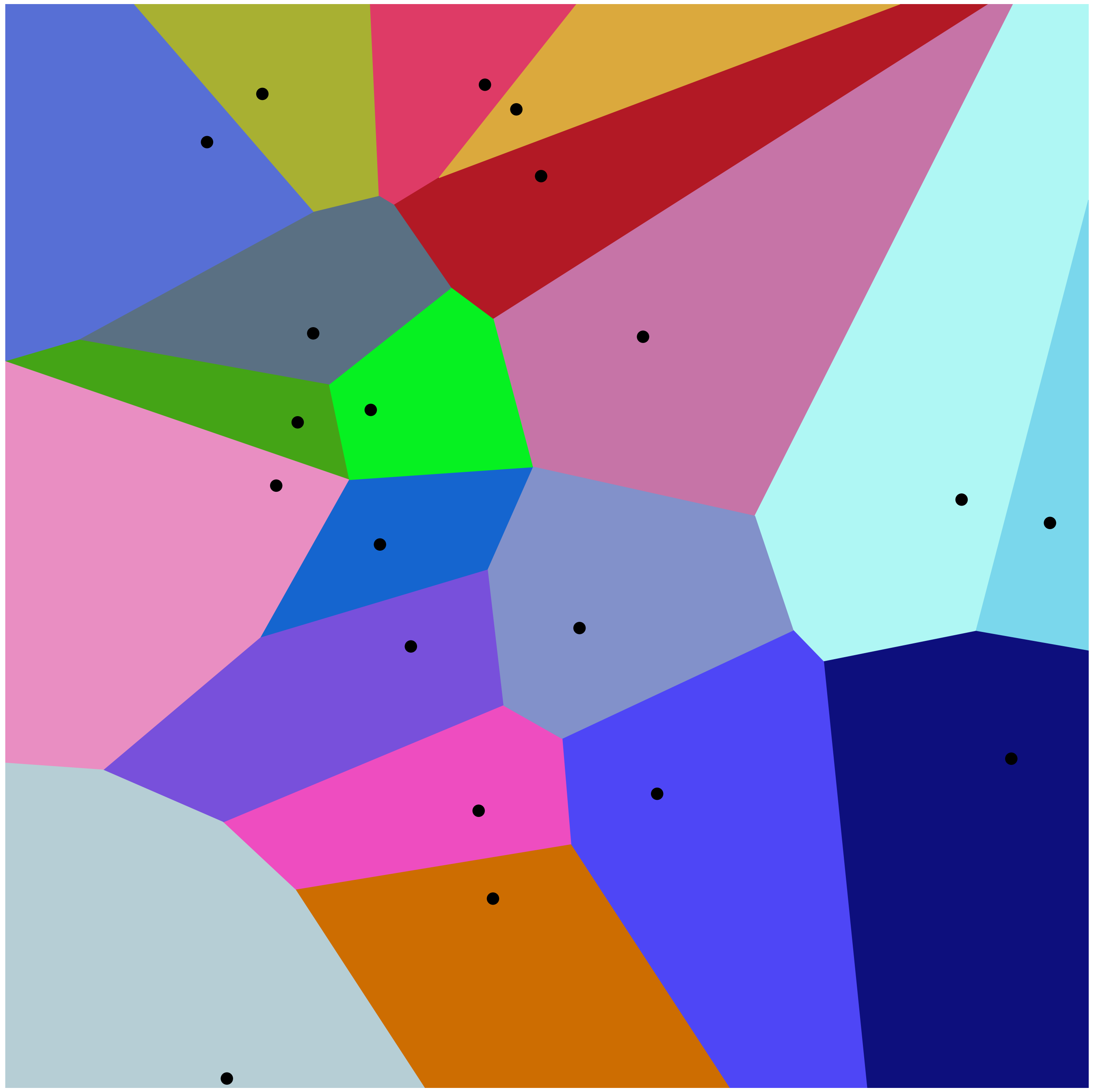
- The number of clusters needs to be specified in advance, or chosen based on heuristics and cross-validation.

  Example: elbow method

- The $k$-means algorithm is not good at handling outliers – every document will eventually belong to some cluster.


Elbow method
250
200
150
100
50
0
Inertia
elbow point
0
1
2
3
4
5
6
7
8
9
Value of k

# K-means is restricted to clusters with convex shapes



# Density-based clustering

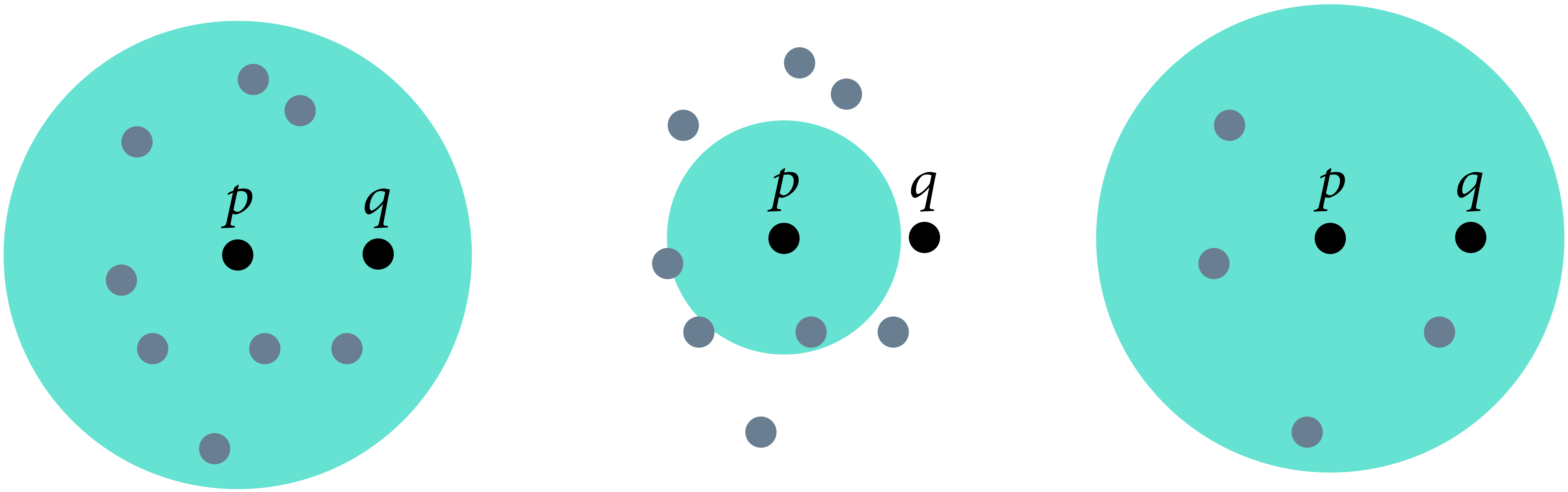
- The basic idea behind **density-based algorithms** is that different regions of the vector space can be more or less densely populated.

- Under this view, clusters can take any shape; they are not constrained to convex clusters as in $k$-means.

## Directly density-reachable

- Informally, a point $q$ should be in the same cluster as a point $p$ if $q$ is close to $p$ and the space between them is densely populated.
- Formally, we define the **$\varepsilon$-neighbourhood** around $p$, denoted by $N_\varepsilon(p)$, as the set of points whose distance from $p$ is at most $\varepsilon$.
- We also set a minimum number of points, denoted by $m$.
- We say that $q$ is **directly density-reachable** from $p$ if (1) $q$ belongs to $N_\varepsilon(p)$ and (2) $N_\varepsilon(p)$ contains at least $m$ points.

# Directly density-reachable




Point q is directly density-reachable from point p.

Point q does not belong to the neighbourhood of p.

↓

Point q is not directly density-reachable from point p.

The neighbourhood of p contains too few points.

↓

Point q is not directly density-reachable from point p.

**eps = 1**
**min_samples = 10**

# Density-reachability and density-connectedness

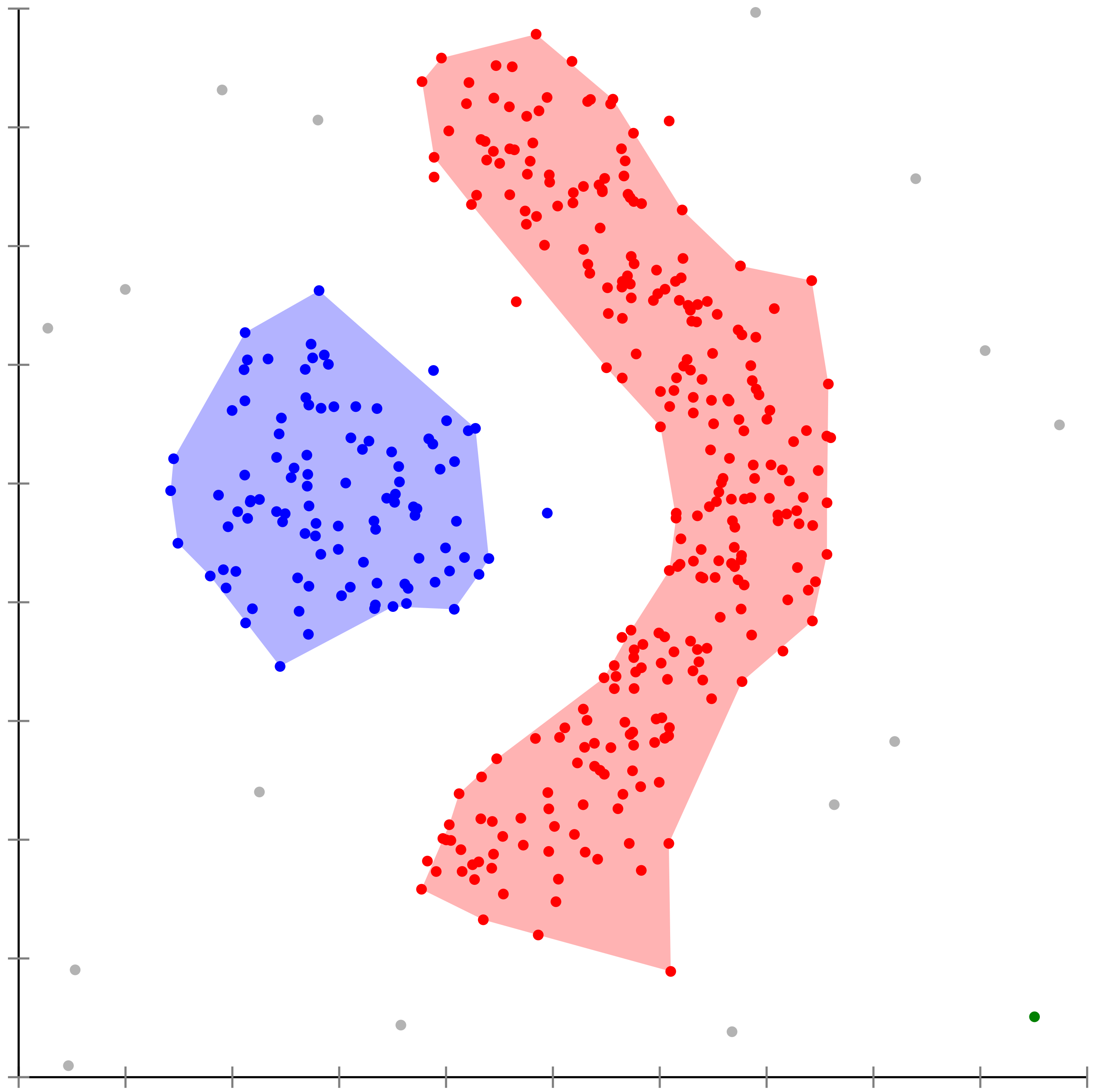
- **Density-reachability** is the reflexive–transitive closure of direct density-reachability.

  Chain of directly density-reachable points.

- Two points *p* and *q* are **density-connected** if there is a third point *r* such that both *p* and *q* are density-reachable from *r*.

- Based on this definition, a cluster can be viewed as a maximal set of density-connected points.

# DBSCAN – Density-based spatial clustering with noise



# Issues with DBSCAN

- DBSCAN can find clusters of arbitrary shape. They do not need to be convex as in $k$-means.
- DBSCAN can handle outliers: Points that do not have a sufficiently large neighbourhood are labelled as 'noise'.
- The size of the neighbourhood and the minimum number of samples in the neighbourhood need to be set in advance.

# This lecture

- Introduction to text clustering
- Similarity measures
- An overview of hard clustering methods
- Evaluation of hard clustering
- Soft clustering: Topic models

# Evaluation of hard clustering

# Intrinsic and extrinsic evaluation

- In **intrinsic evaluation**, a clustering is evaluated based on internal measures such as coherence and separation.

  Are documents in the same cluster similar? Are clusters well-separated?

- In **extrinsic evaluation**, a clustering is evaluated based on data that was not used for the clustering, such as known class labels.

  cluster purity, Rand index

# Manual evaluation using cluster summaries

- One way to manually evaluate the quality of a cluster is to generate a short summary of each cluster.
- To do so, we take the centroid of the cluster, and identify the *k* highest-weighted terms in that vector.

0: product great good use just like does hair time did

1: album cd music songs quot song like just great band

2: movie film movies like story watch just good great acting

3: book read books author reading story like quot just written

4: software program version product computer use support windows microsoft easy

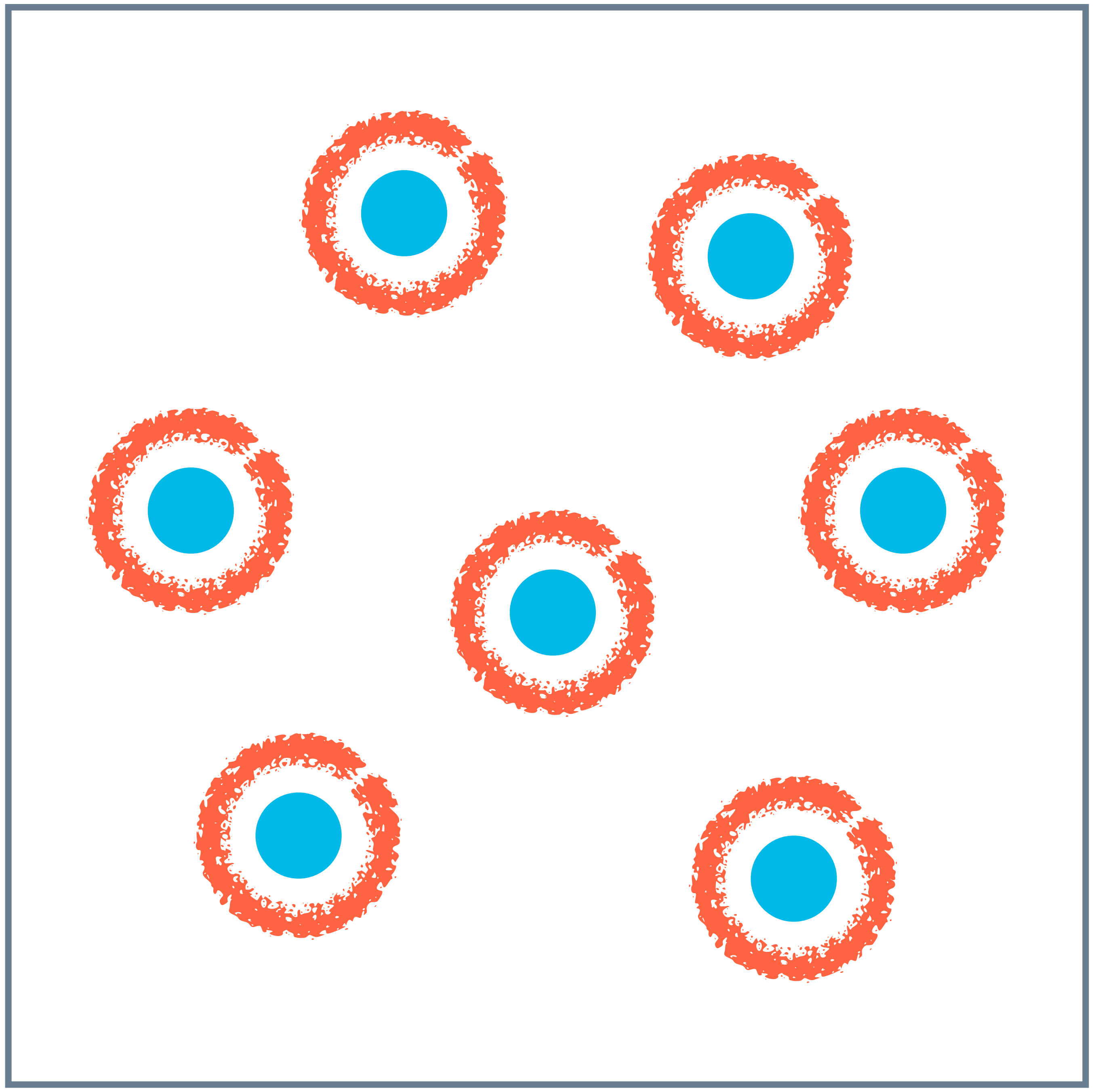
5: camera lens pictures canon digital use flash battery quality great

## Cluster purity

- Suppose that we have gold-standard class labels, perhaps only for a subset of the data (evaluation set).
- Intuitively, a cluster whose elements are distributed over few classes is better than one that contains many different classes.
- Formally, let $N$ be the number of documents, let $M$ be the set of clusters, and let $D$ be the partitioning into classes. Then

$$\text{purity} = \frac{1}{N} \sum_{m \in M} \max_{d \in D} |m \cap d|$$

# Purity does not penalise trivial clusters



# Rand index

- We can view a clustering as a binary classifier that maps a *pair* of documents to 'true' if and only if they belong to the same cluster.
- The **Rand index** of a clustering measures the accuracy of this classifier relative to the gold-standard class assignment.

  true positive = same cluster and same gold-standard class label

# Qualitative evaluation

- In the absence of relevant measures for the evaluation of clusterings, one alternative is to do a qualitative evaluation.
- In a first step, one generates a set of hypotheses about the clustering, based on knowledge about a domain.

  Example: movies and books should be in different clusters
- Then, one inspects the clustering and checks whether it actually exhibits the hypothesised properties.

  Important to do this *after* generating the hypotheses!

# This lecture

- Introduction to text clustering
- Similarity measures
- An overview of hard clustering methods
- Evaluation of hard clustering
- Soft clustering: Topic models

# Topic models

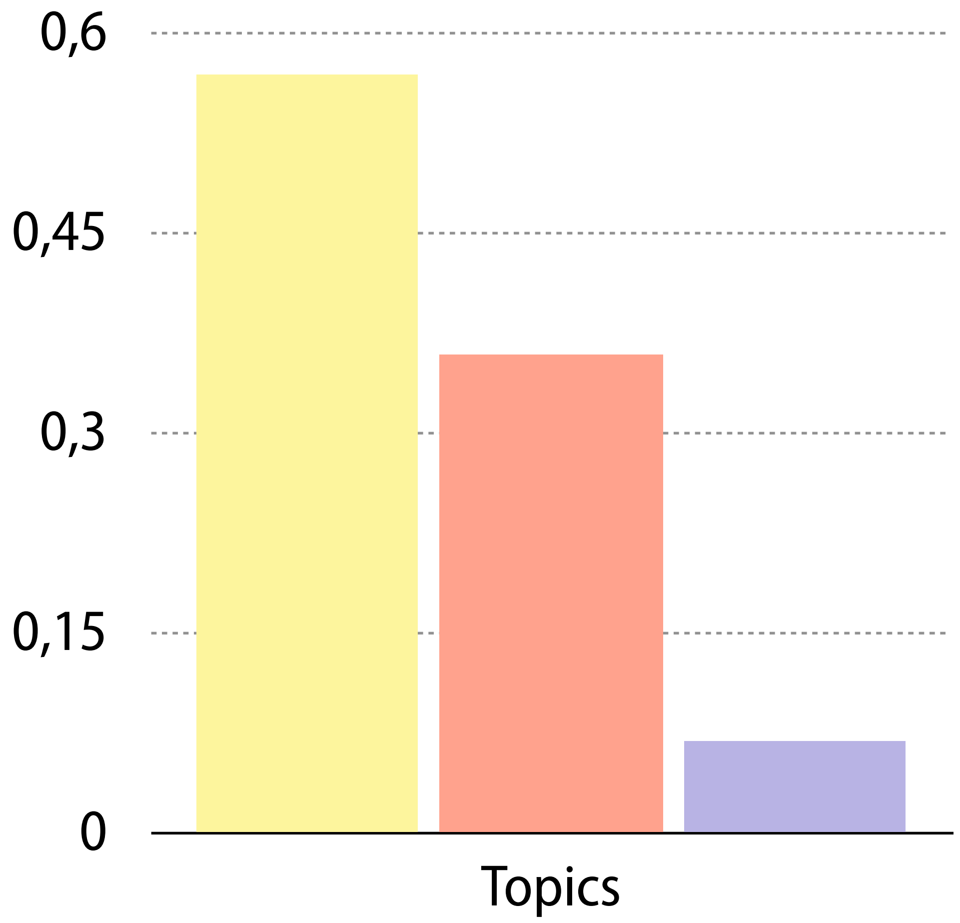
# Topic models

- A **topic model** is a statistical model for representing the abstract topics that are expressed in a collection of documents.
- Topic models are examples of soft clustering techniques – each document belongs to each cluster (topic) to a certain degree.
- This lecture will focus on **Latent Dirichlet Allocation (LDA)**, the most common topic model currently in use.

  Blei, Ng, and Jordan (2002)

# Topic models

How many genes does an organism need to survive? Last week at the genome meeting here, two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes …




Source: Blei (2012)

# Topic models

| | | |
|---|---|---|
| human | evolution | computer |
| genome | evolutionary | models |
| dna | species | information |
| genetic | organisms | data |
| genes | life | computers |
| sequence | origin | system |
| gene | biology | network |
| molecular | groups | systems |
| sequencing | phylogenetic | model |
| map | living | parallel |
| information | diversity | methods |
| genetics | group | networks |
| mapping | new | software |
| project | two | new |
| sequences | common | simulations |

Source: Blei (2012)

# Prelude: Language models

- A (probabilistic) **language model** is a probability distribution over sequences of words in some language.
- In a **unigram language model**, the probability of a sequence is broken down into a product of single words.

  special case of a more general family of *n*-gram language models

text

count of *w* in the text

$$P(w_1 \cdots w_N) = \prod_{k=1}^{N} P(w_k) = \prod_{w \in V} P(w)^{\#(w)}$$

# Naive Bayes as a collection of unigram models

choose that class $c$ which maximises
the term to the right of the ‘arg max’

unigram model

$$\hat{c} = \arg\max_{c \in C} P(c) \cdot \boxed{\prod_{w \in V} P(w \mid c)^{\#(w)}}$$

predicted class
for the document

count of the word $w$
in the document

# Generative story for Naive Bayes

How does Naive Bayes generate a corpus?

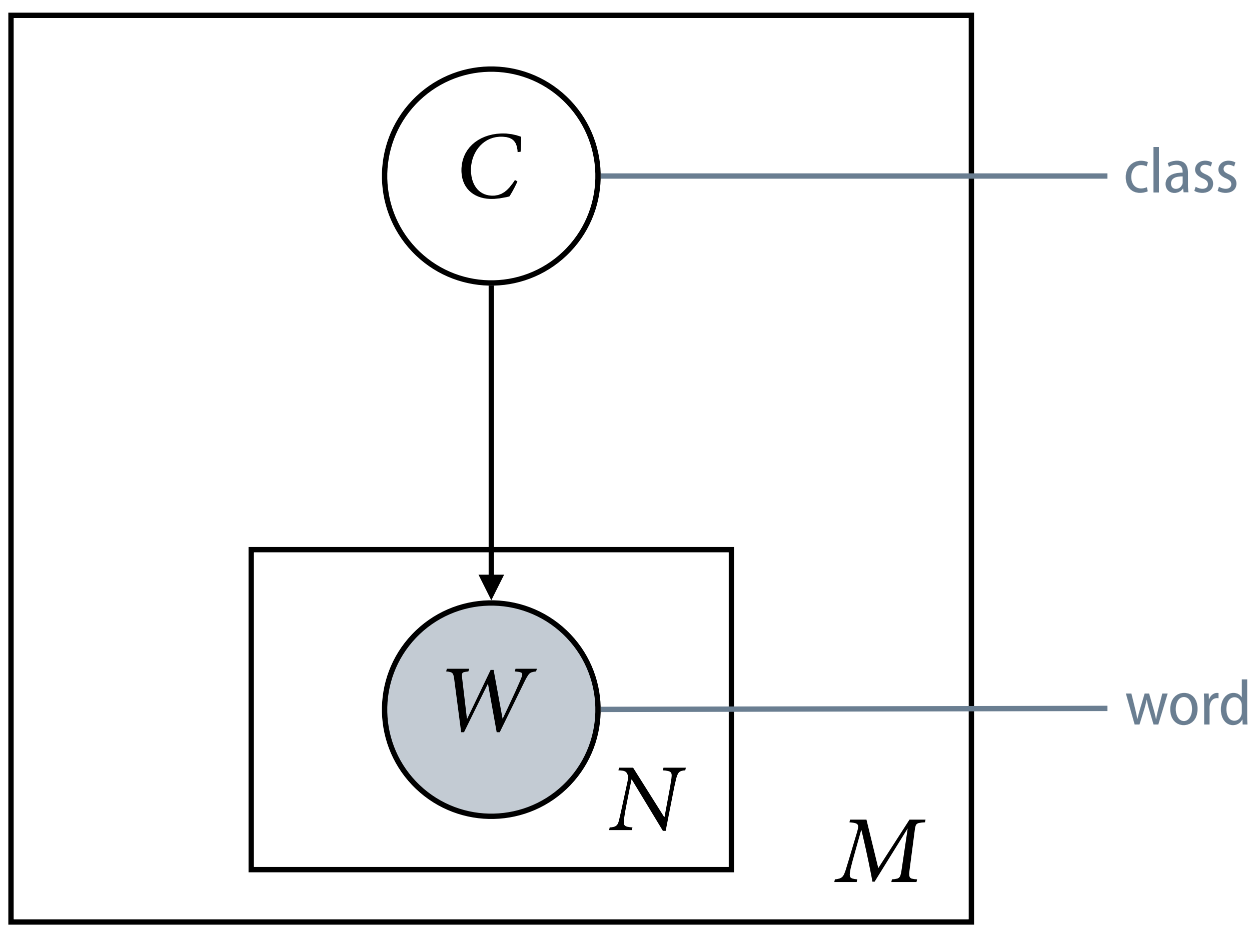
For each document $d$:

Draw a class. $c \sim \mathrm{Cat}(\gamma)$

For each position $i$ in $d$:

Draw a word from the vocabulary. $w_{d,i} \sim \mathrm{Cat}(\beta_c)$

# Graphical model for Naive Bayes




M – number of documents in the corpus, N – number of words per document

# Topic-specific unigram models


$z = 1$ $\quad$ $z = 2$ $\quad$ $z = 3$

$P(\blacksquare) = 90\%$ $\quad$ $P(\blacksquare) = 50\%$ $\quad$ $P(\blacksquare) = 10\%$

# Generating a corpus from a topic model

For each topic $k$:

Draw a unigram model. $\beta_k \sim \mathrm{Dir}(\eta)$

For each document $d$:

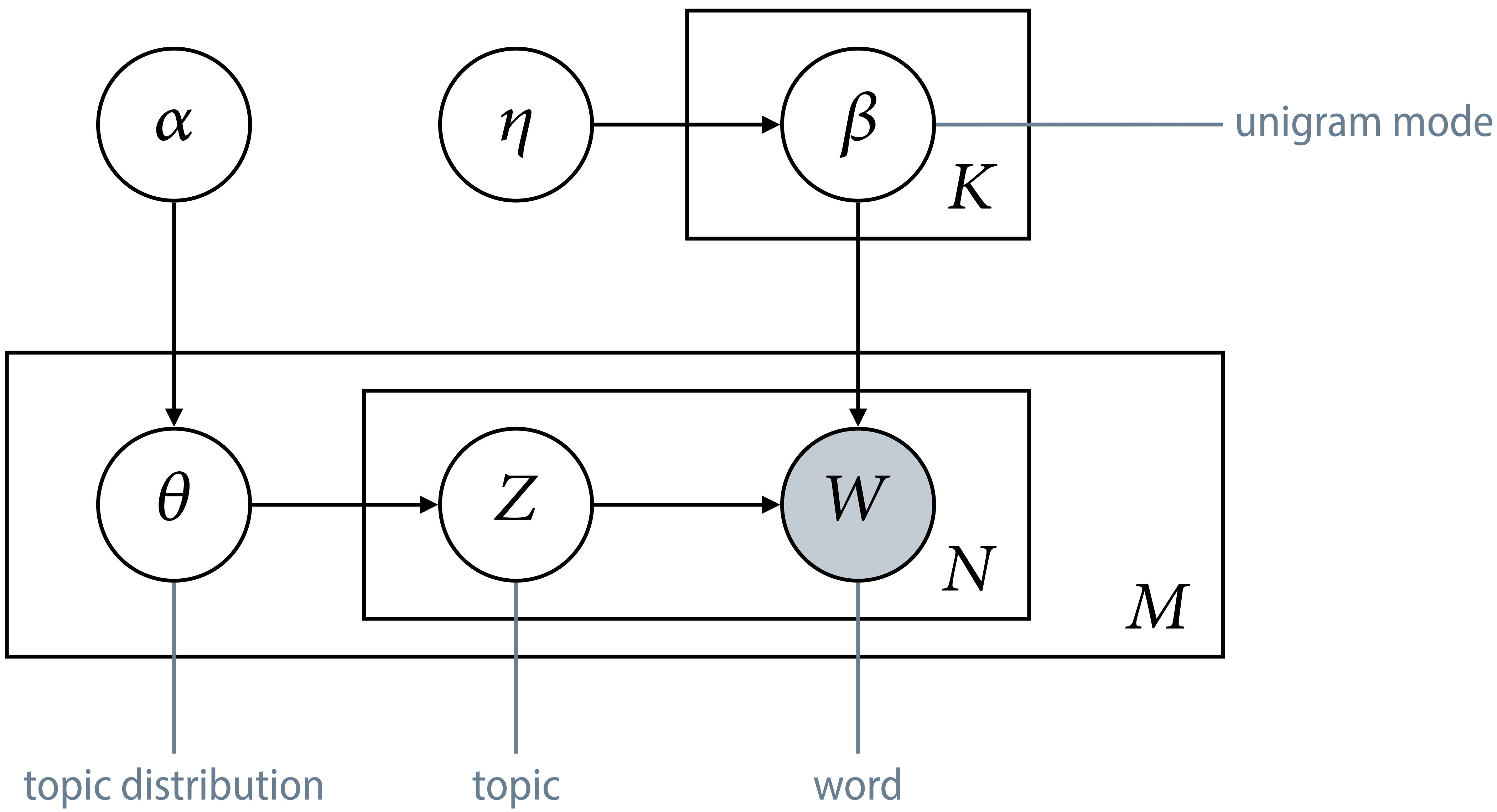
Draw a topic distribution. $\theta_d \sim \mathrm{Dir}(\alpha)$

For each position $i$ in $d$:

Draw a topic assignment. $z_{d,i} \sim \mathrm{Cat}(\theta_d)$

Draw a word from the vocabulary. $w_{d,i} \sim \mathrm{Cat}(\beta_{z_{d,i}})$

# Graphical model for Latent Dirichlet Allocation

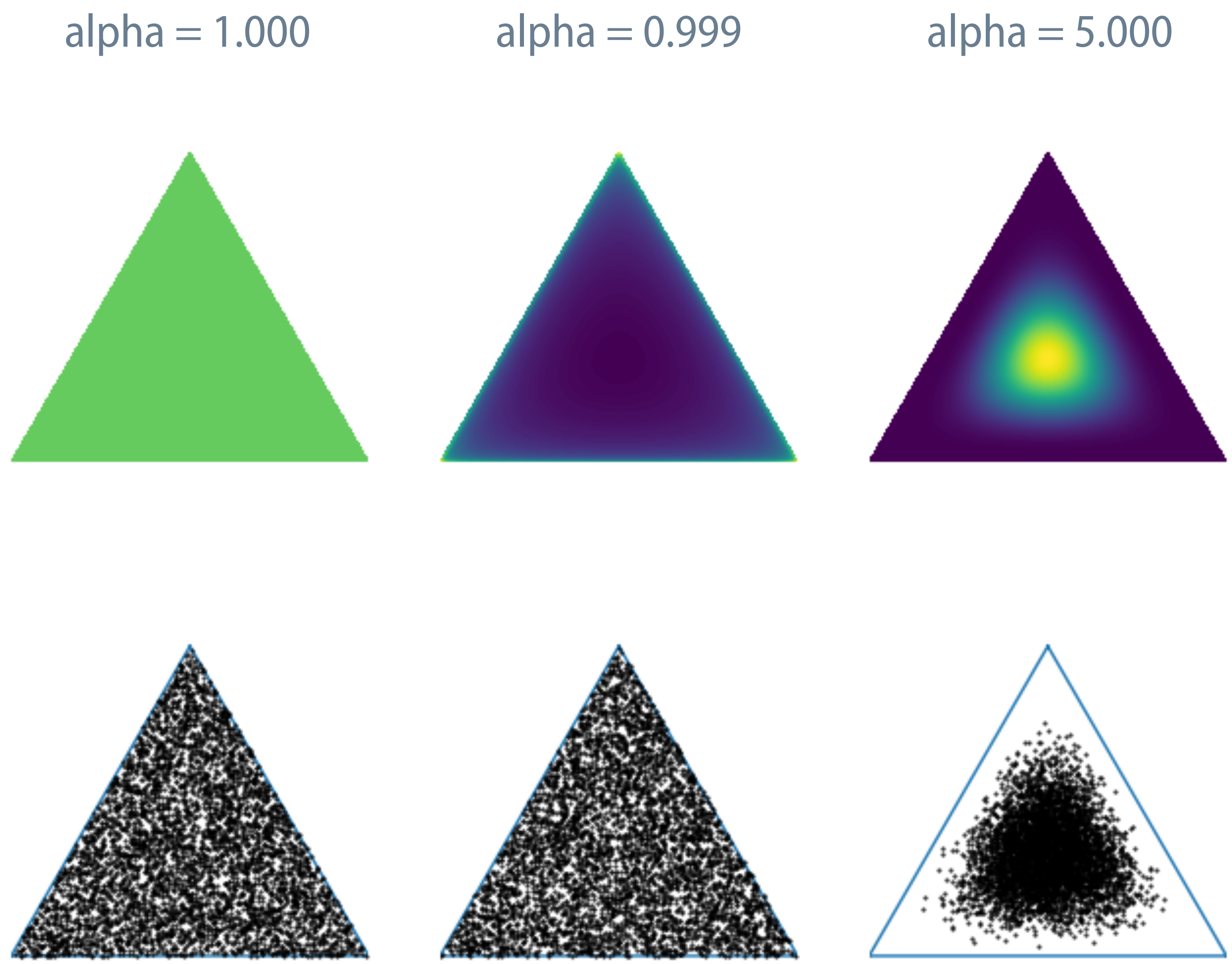



K – number of topics, M – number of documents in the corpus, N – number of words per document

## Hyperparameters

- The hyperparameter $K$ specifies the number of topics.
- The hyperparameter $\alpha$ controls the sparsity in the document-specific topic distributions.
- The hyperparameter $\eta$ controls the sparsity in the topic-specific word distribution (unigram model).

# Probability density of the Dirichlet distribution




Plot by Thomas Boggs

# Learning of topic models

- At learning time, we only know the words, but neither the topic distribution, topic assignments, nor the unigram models.

- We can apply Bayes' rule to get the posterior distribution of all these variables given the words.

- Direct computation of this posterior distribution is intractable. However, we can use Gibbs sampling.

  alternative: Variational Bayes Expectation–Maximization

# Evaluation of topic models

- During training, after each pass through the corpus we can log the marginal posterior $p(\boldsymbol{w} \mid \boldsymbol{z})$; this quantity should converge.

  similar role as loss in neural networks

- After training, we can compute the marginal likelihood on held-out data in order to compare different models.

  Wallach et al. (2009)

- We can inspect the generated topic models in order to assess their coherence and overall quality.

# Issues in evaluation

- **Stop words**

  Before assessment, topics can be filtered for stop words – if these were not already removed before training.

- **Junk topics**

  Often, one or a few topics simply contain generally common words. The recommendation is to ignore these junk topics.

# Qualitative evaluation

- In the absence of relevant measures for the evaluation of topic models, one alternative is to do a qualitative evaluation.

- In a first step, one generates a set of hypotheses or expectations about the topics, based on knowledge about a domain.

  Example: In *Harry Potter*, we would expect a *Life at Hogwarts* topic.

- Then, one inspects the topics and checks whether they actually exhibit the expected properties.

  Important to do this *after* generating the hypotheses!

# This lecture

- Introduction to text clustering
- Similarity measures
- An overview of hard clustering methods
- Evaluation of hard clustering
- Soft clustering: Topic models