

Statistical Natural Language Processing

Part V: NLP using Clustering

Henning Wachsmuth

<https://ai.uni-hannover.de>

Learning Objectives

Concepts

- Different types of clustering
- Pros and cons of the different types
- How to employ unsupervised learning within NLP
- Evaluation of clustering

Methods

- Partitioning of a set of texts into groups with flat clustering
- Modeling topics of texts with soft clustering
- Ordering of texts by similarity with hierarchical clustering

Tasks

- Authorship attribution
- Topic detection
- Sentiment analysis

Outline of the Course

- I. Overview
- II. Basics of Data Science
- III. Basics of Natural Language Processing
- IV. Representation Learning
- V. NLP using Clustering
 - Introduction
 - Flat Clustering
 - Soft Clustering
 - Hierarchical Clustering
 - Conclusion
- VI. NLP using Classification and Regression
- VII. NLP using Sequence Labeling
- VIII. NLP using Neural Networks
- IX. NLP using Transformers
- X. Practical Issues

Introduction

Clustering

Clustering (aka cluster analysis)

- The grouping of a set of instances into $k \geq 1$ classes, called *clusters* k is possibly, but not necessarily predefined.
- The meaning of clusters is usually unknown beforehand.
- The resulting model can assign arbitrary instances to clusters.

Similarity measures in clustering

- Clustering algorithms use similarities to find patterns in instances.
- To merge clusters, similarities are also computed between clusters.
Different ways to define cluster similarity exist (details below).

Clustering vs. cluster labeling

- Clustering does *not* assign labels to the created clusters.
- Cluster labeling requires to infer the hidden concept connecting the instances in a group.

Not in the scope of this course

Clustering

Types of Clustering Techniques

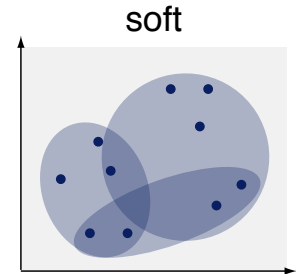
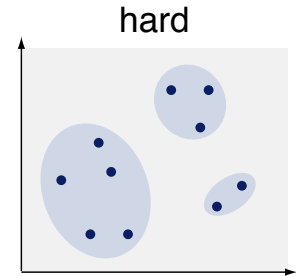
Hard vs. soft clustering

- **Hard.** Create a partition, such that each instance $\mathbf{x}^{(i)}$ belongs to a single cluster c_j .

$$\{1, 2, 3, 4\} \rightarrow c_1 = \{1, 3, 4\}, c_2 = \{2\}$$

- **Soft.** Create overlapping clusters, such that each $\mathbf{x}^{(i)}$ belongs to c_j with a weight $w_j^{(i)} \in [0, 1]$.

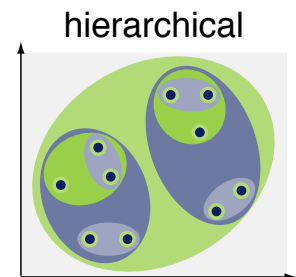
$$\{1, 2, 3, 4\} \rightarrow c_1 = (1.0, 0.6, 0.8, 0.0), c_2 = (0.0, 0.4, 0.2, 1.0)$$



Flat vs. hierarchical clustering

- **Flat.** Create a set of independent clusters. (both above)
- **Hierarchical.** Create a binary tree over all instances where each node represents a cluster of a certain size.

$$\{1, 2, 3, 4\} \rightarrow \{ \{ \{ \{1\}, \{3\} \}, \{4\} \}, \{2\} \}$$



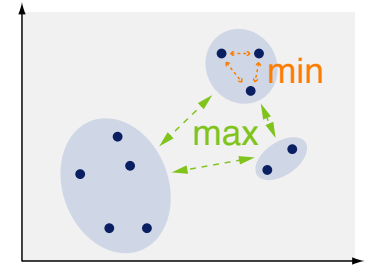
Clustering

Clustering in NLP

Clustering as unsupervised learning

- Clustering models y are mostly learned unsupervised.
- The goal is to minimize the distance within clusters, and to maximize it between the clusters.

Or: Maximize *similarity* within, minimize between



Why clustering in NLP?

- Clustering often targets situations where the set of classes is unknown.
- The main goal is to find out which instances belong to the same class.

Selected applications in NLP

- **Topic detection.** Identifying the topics covered in a text corpus
- **Text retrieval.** Finding texts that share similar properties

Similarity may capture content, structure, and/or style.

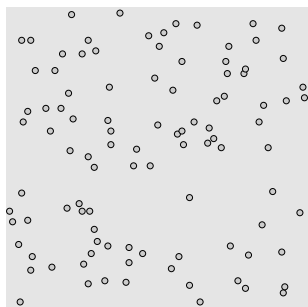
Evaluation of Clustering

Main evaluation goals

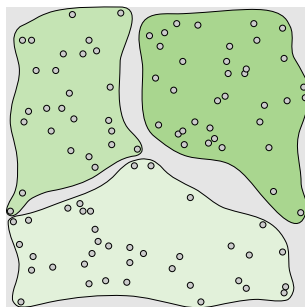
- Rank different clusterings by quality (in terms of class representation).
- Determine the ideal number of clusters k .

Problem

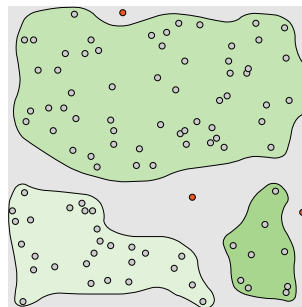
- Various possible ways to cluster a set of instances exist.



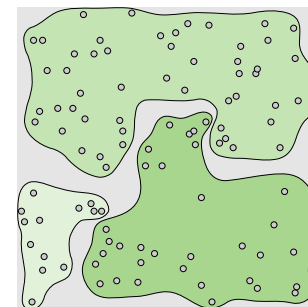
Set of instances



k -means



DBSCAN



Complete link

- Without a ground truth, deciding what is best is often hard.

Types of evaluation

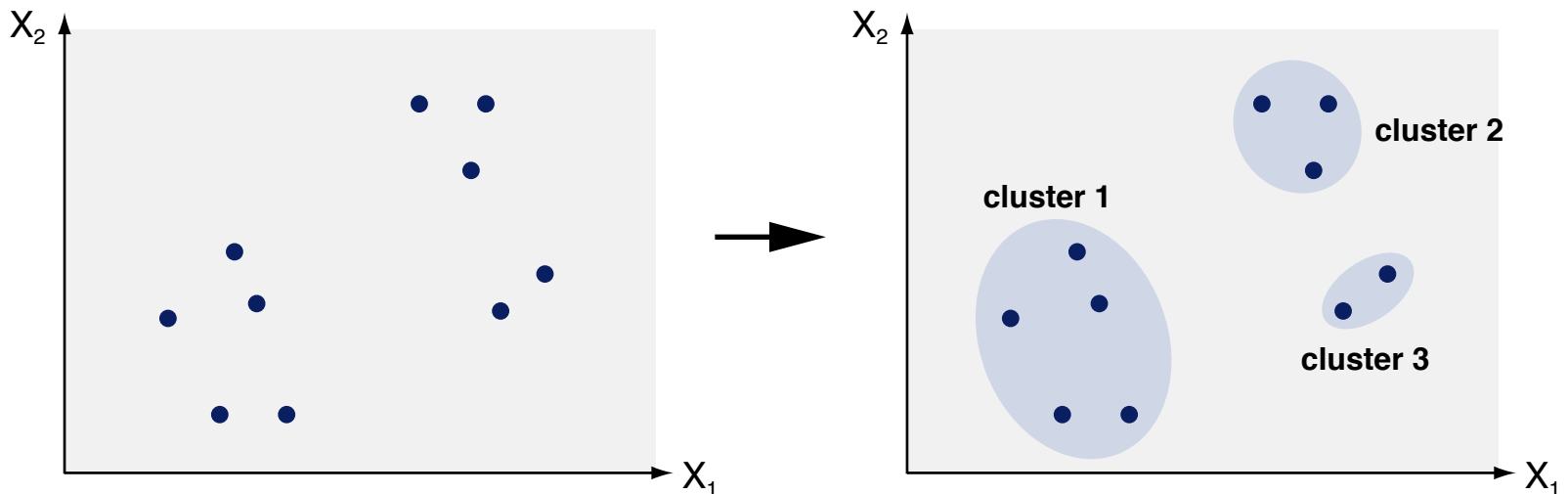
- **Intrinsic.** Quantify cluster quality based on distance, size, and/or shape.
- **Extrinsic.** Given a ground-truth test set, compare different clusterings.

Flat Clustering

Flat Clustering

Flat (hard) clustering

- A clustering technique that partitions instances into disjunct clusters
- **Input.** A set of instances $X = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\}$ without class labels
- **Output.** A set of clusters $C = \{c_1, \dots, c_k\}$ and a mapping $X \rightarrow C$



Number of clusters k

- Many clustering algorithms have k as a hyperparameter.
- Some determine k automatically.

Flat Clustering

Two Main Types of Algorithms

Iterative algorithms

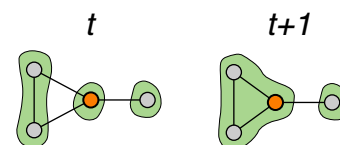
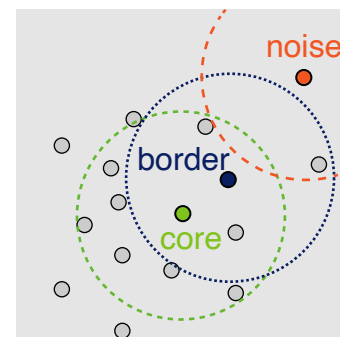
- Iterative clustering and re-assignment of instances to clusters
- Exemplar-based (e.g., *k-means*). Instances are considered in isolation when adding them to clusters.

We focus on this type here.

- Exchange-based (e.g., *Kerninghan-Lin*). Instances are exchanged between pairs of clusters.

Density-based algorithms

- Clustering of instances into regions of similar density
- Point density (e.g., *DBSCAN*). Distinction of instances in the *core* of a region, at the *border*, and *noise*
- Attraction (e.g., *MajorClust*). Instances in a cluster “join forces” to “attract” further instances.



Flat Clustering with k-means

Cluster centroid

- The mean of all instances in a cluster, i.e., the average of their vectors.

k -means clustering

- A simple flat clustering algorithm that creates $k \geq 1$ clusters
- Instances are assigned to the cluster whose centroid is closest to them.
- k is a hyperparameter to be optimized.

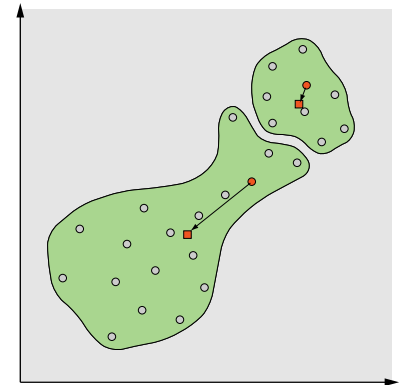
Chosen based on evaluation measures (see below) or based on domain knowledge

k -means in a nutshell

1. Compute centroids of candidate clusters.
2. Re-cluster based on similarity to centroids.
3. Repeat until convergence.

Variations

- Some versions of k -means include a maximum number of iterations.



Flat Clustering with k-means

Pseudocode

Signature

- **Input.** A set of instances X , a number of clusters k
- **Output.** A clustering C , i.e., a set of clusters

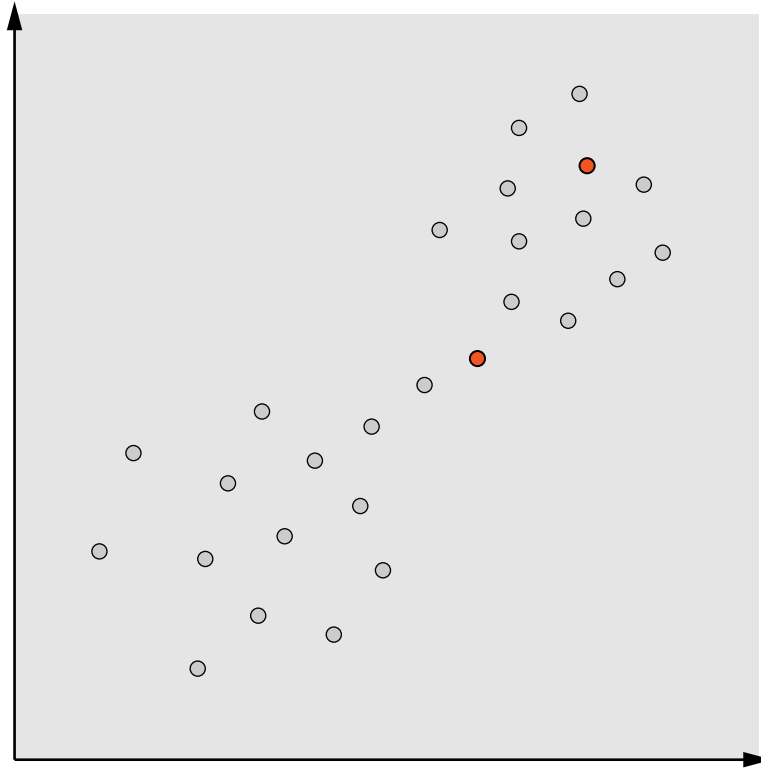
kMeansClustering(Set<Instance> X , int k)

```
1.  Set<Instance> [] clusters ← ∅
2.  Instance [] centroids ← chooseRandomInstances( $X$ ,  $k$ )
3.  repeat
4.      Instance [] prevCentroids ← centroids
5.      for int  $i$  ← 1 to  $k$  do clusters[ $i$ ] ← ∅
6.      for each  $x \in X$  do // create clusters
7.          int  $z$  ← 1
8.          for int  $j$  ← 2 to  $k$  do // find nearest centroid
9.              if  $\text{sim}(x, \text{centroids}[j]) > \text{sim}(x, \text{centroids}[z])$  then  $z \leftarrow j$ 
10.         clusters[ $z$ ] ← clusters[ $z$ ] ∪ { $x$ }
11.         for int  $i$  ← 1 to  $k$  do // update centroids
12.             centroids[ $i$ ] ← computeMean(clusters[ $i$ ])
13.     until prevCentroids = centroids // convergence
14.     return clusters
```

Flat Clustering with k-means

Example: 2-means with Euclidean Distance

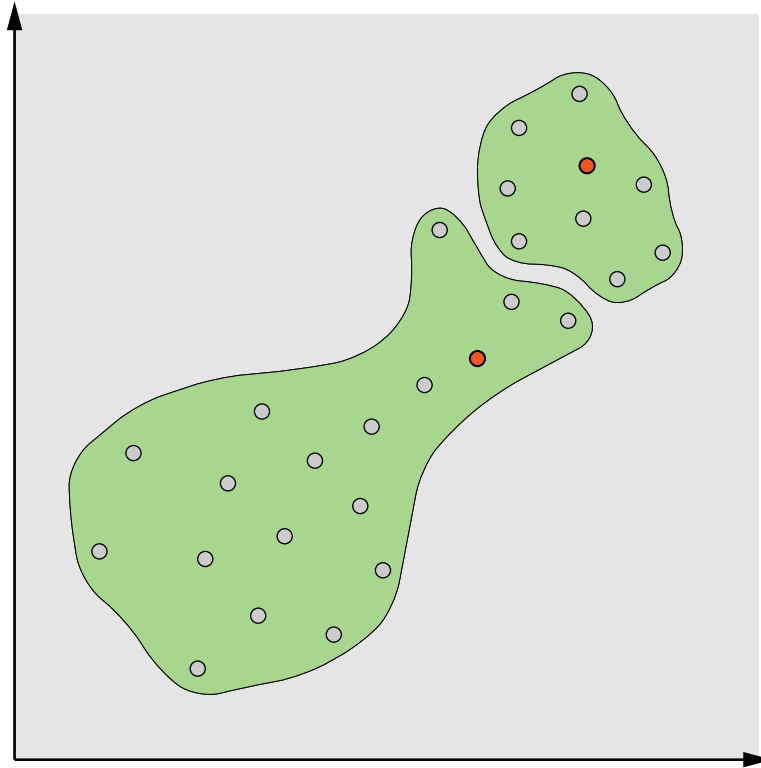
Pseudocode line 2: Choose k instances randomly as initial “centroids”



Flat Clustering with k-means

Example: 2-means with Euclidean Distance

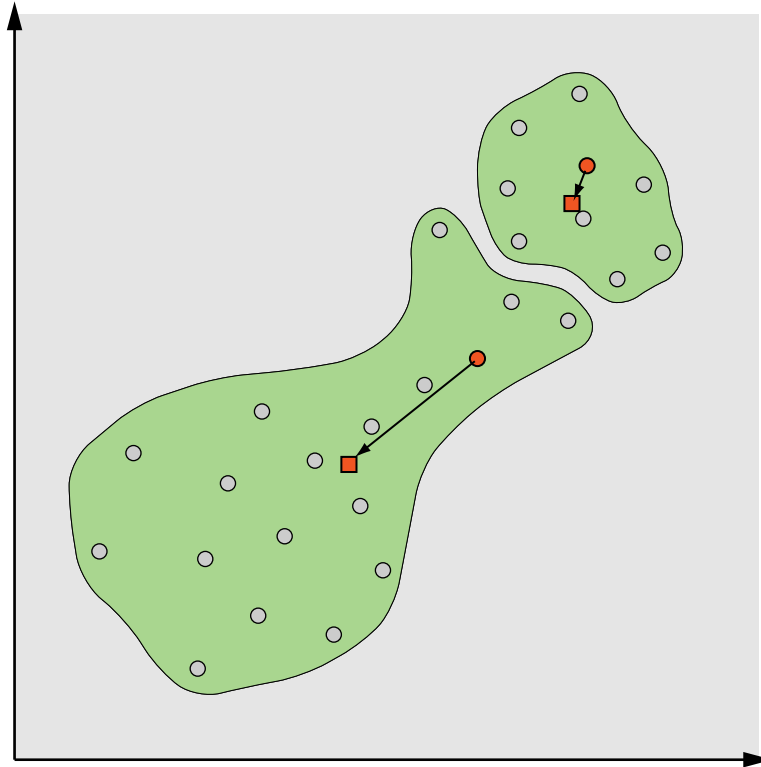
Pseudocode lines 5–10: Cluster by distance to the k centroids



Flat Clustering with k-means

Example: 2-means with Euclidean Distance

Pseudocode lines 11–12: Recompute centroids of the k clusters



Flat Clustering with k-means

Example: 2-means with Euclidean Distance

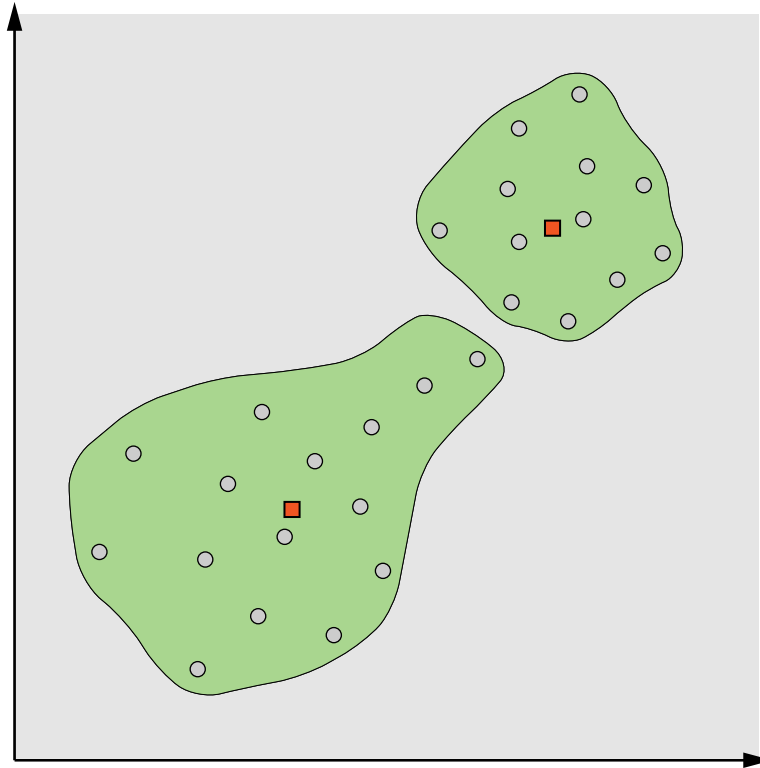
Repeat until convergence (lines 5–10 again)



Flat Clustering with k-means

Example: 2-means with Euclidean Distance

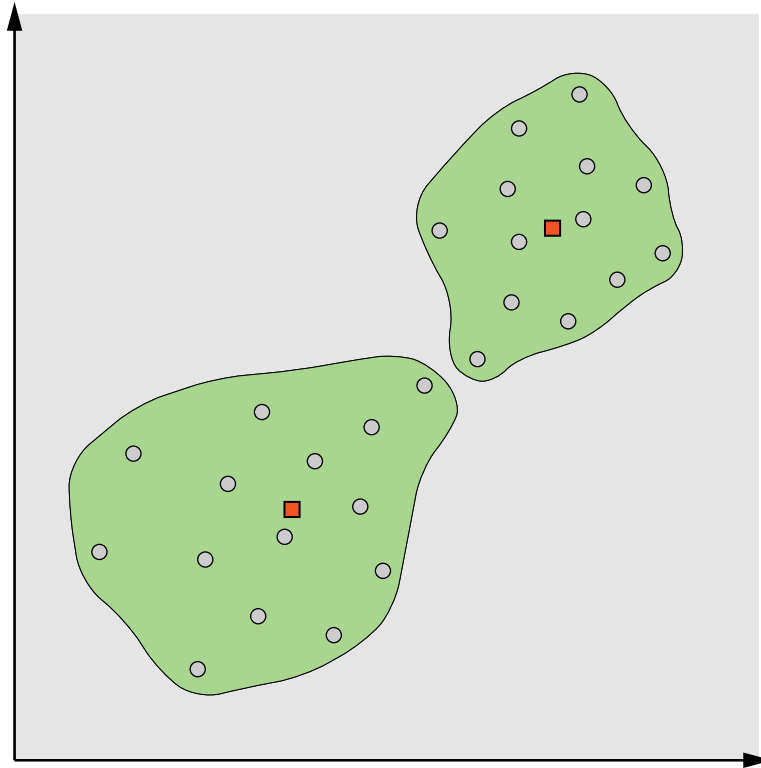
Pseudocode lines 11–12 again: Repeat until convergence



Flat Clustering with k-means

Example: 2-means with Euclidean Distance

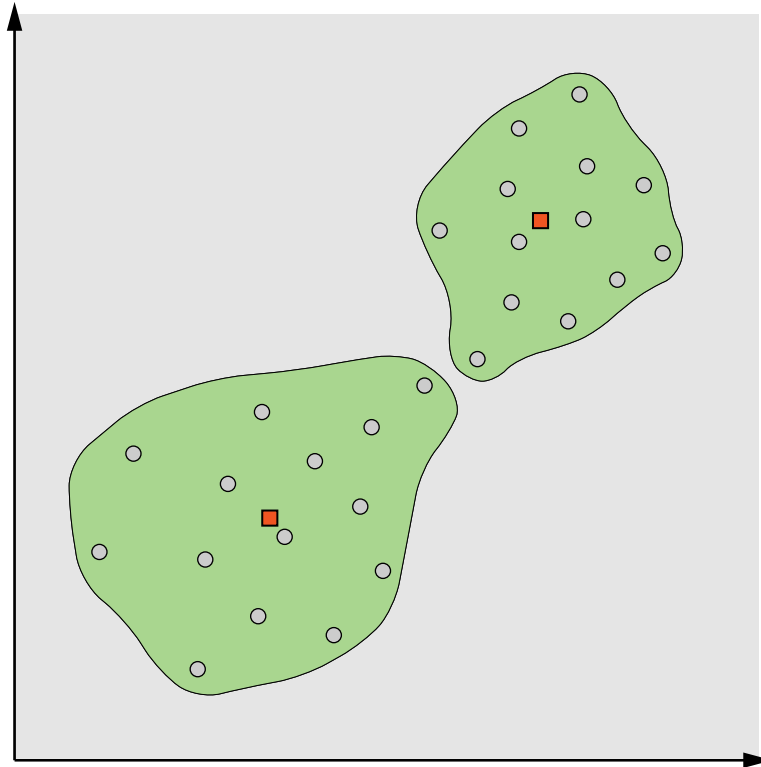
Pseudocode lines 11–12 again: Repeat until convergence (2)



Flat Clustering with k-means

Example: 2-means with Euclidean Distance

Pseudocode lines 11–12 again: Repeat until convergence (3) → done!



Evaluation of Clustering

Choice of the number of clusters

- Unless decided by expert knowledge, k needs to be evaluated against some intrinsic or extrinsic scoring function.
- However, most used functions grow (or fall) with the number of clusters.

Selected scoring functions

- **Intrinsic.** Maximum cluster size → highest for $k = 1$
- **Intrinsic.** Maximum cluster distance → highest for $k = |X|$
- **Intrinsic.** Mean centroid distance of instances → highest for $k = |X|$
- **Extrinsic.** B³ F₁-score → highest for $k = |X|$
- **Extrinsic.** Purity of clusters → highest for $k = |X|$

Common intrinsic evaluation measures

- **Elbow criterion.** Find the k that maximizes score reduction.
- **Silhouette analysis.** Optimize distances and sizes of clusters.

Both measures have a visual intuition, but work mathematically.

Evaluation of Clustering

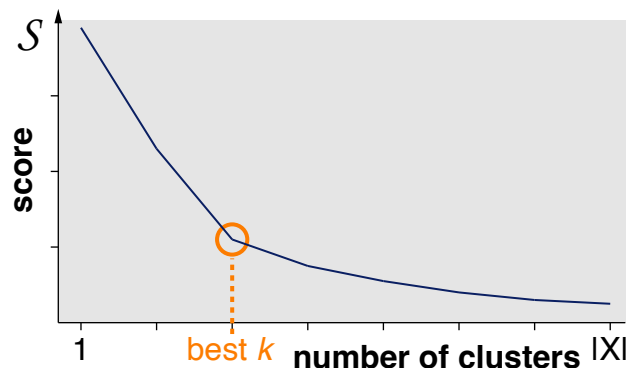
Elbow Criterion

Elbow criterion

- A method to find the best value of a hyperparameter (say, k in k -means)
- Requires some scoring function \mathcal{S} (e.g., the mean centroid distance)

Input

- A set of clusterings $\mathcal{C} = \{C_1, \dots, C_p\}$ for hyperparameter values $k_1 \leq \dots \leq k_p$
- A score $\mathcal{S}(C_i)$ for each clustering C_i



Approach

- **Visually.** Pick the k where the curve has an “elbow” — if visible!
- **Computationally.** Pick the k with the maximum second derivate:
This reflects the point where the score reduction changes strongest.

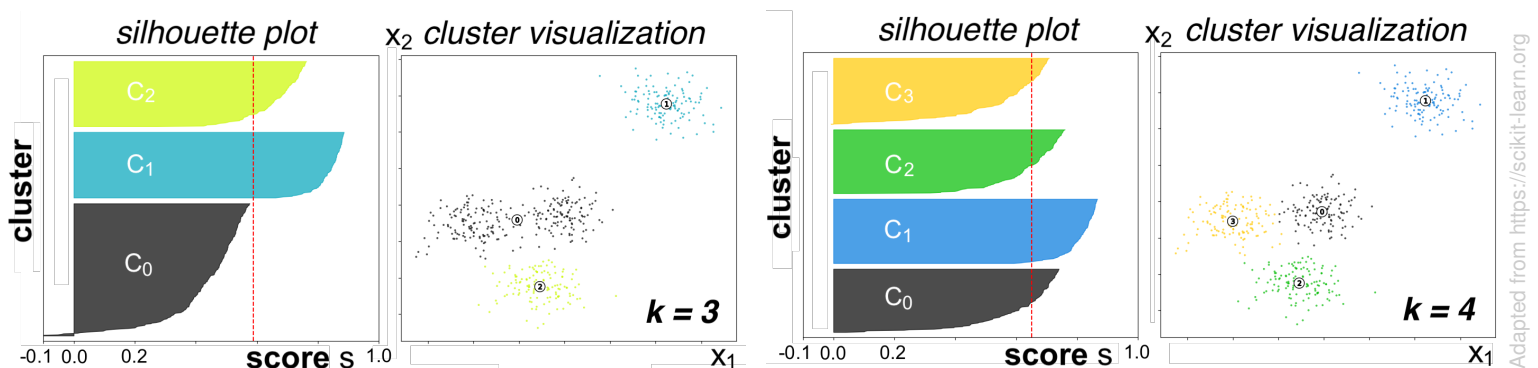
$$\begin{aligned} k &:= \operatorname{argmax}_i ((\mathcal{L}(C_{i-1}) - \mathcal{L}(C_i)) - (\mathcal{L}(C_i) - \mathcal{L}(C_{i+1}))) \\ &= \operatorname{argmax}_i (\mathcal{L}(C_{i-1}) - 2 \cdot \mathcal{L}(C_i) + \mathcal{L}(C_{i+1})) \end{aligned}$$

Evaluation of Clustering

Silhouette Analysis

Silhouette analysis

- A method to find the best number of clusters k in clustering
- Computes a score $s \in [-1, 1]$ for each cluster c_j of a clustering C_i that reflects how far each $x \in c_j$ is to instances from other clusters
 - » 0 : x likely in right cluster, 0 : boundary to other clusters; < 0 : likely in wrong cluster



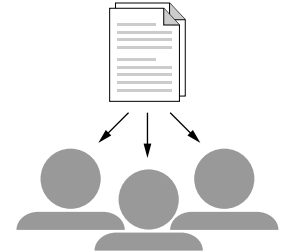
Approach

- **Visually.** Pick the k where most scores (x-axis) are above average, and where the cluster size (y-axis) is balanced.
- **Computationally.** Pick k with maximum average score (vertical red line).

Authorship Attribution

Authorship attribution

- The text analysis that determines the authors of texts
- Tackled in NLP as a downstream task
- May be both supervised or unsupervised



Observations

- Unlike in many tasks, computers tend to be better than humans here.
- Style features are often helpful, such as *stopword n-grams*.

“The happening of some of the cases given: the clearance of approval by the ...”

Case study: CLEF 2016 Shared Task

- Shared task: Teams compete with approaches on the same task/data
- **Task.** Given a corpus with ≤ 100 texts, identify the number k of authors and assign each text to one author.
- **Data.** Training sets are given; results are computed on unseen test sets
Opinion articles and reviews in Dutch, English, and Greek (400–800 words)

Authorship Attribution

Case Study: Approaches

Eight participating teams

- Two teams used k -means, including an estimation of the best k .
- The others determined authors based on different criteria first.

k -means approach #1 (Mansoorizadeh et al., 2016)

- **Features.** Word, POS, and punctuation n -grams, sentence lengths
- **Similarity.** Cosine
- **Choosing k .** Create a similarity graph using similarity threshold 0.5; use number of subgraphs as k .

k -means approach #2 (Sari and Stevenson, 2016)

- **Features.** TF-IDF on character n -grams, average word embeddings
- **Similarity.** Cosine
- **Choosing k .** Take the k that results in the highest silhouette score.

Authorship Attribution

Case Study: Results

B^3 measure for a text d

- B^3 precision. Proportion of texts in the cluster of d by the author of d .
- B^3 recall. Proportion of texts by the author of d found in the cluster of d .
- B^3 F_1 -score. Harmonic mean as usual

The values are averaged over all texts.

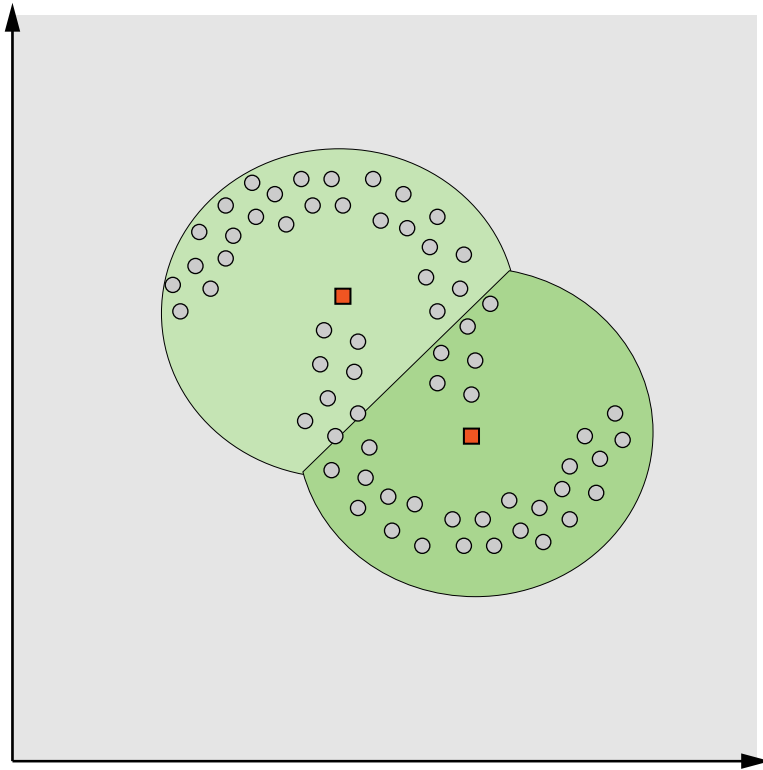
Effectiveness and efficiency results

Approach	B^3 precision	B^3 recall	B^3 F_1 -score	Run-time
Kocher	0.982	0.722	0.822	00:01:51
Bagnall	0.977	0.726	0.822	63:03:59
Sari and Stevenson	0.893	0.733	0.795	00:07:48
Zmiycharov et al.	0.852	0.716	0.768	01:22:56
Gobeill	0.737	0.767	0.706	00:00:39
Kuttichira	0.512	0.720	0.588	00:00:42
Mansoorizadeh et al.	0.280	0.822	0.401	00:00:17
Vartapetian and Gillam	0.195	0.935	0.234	03:03:13

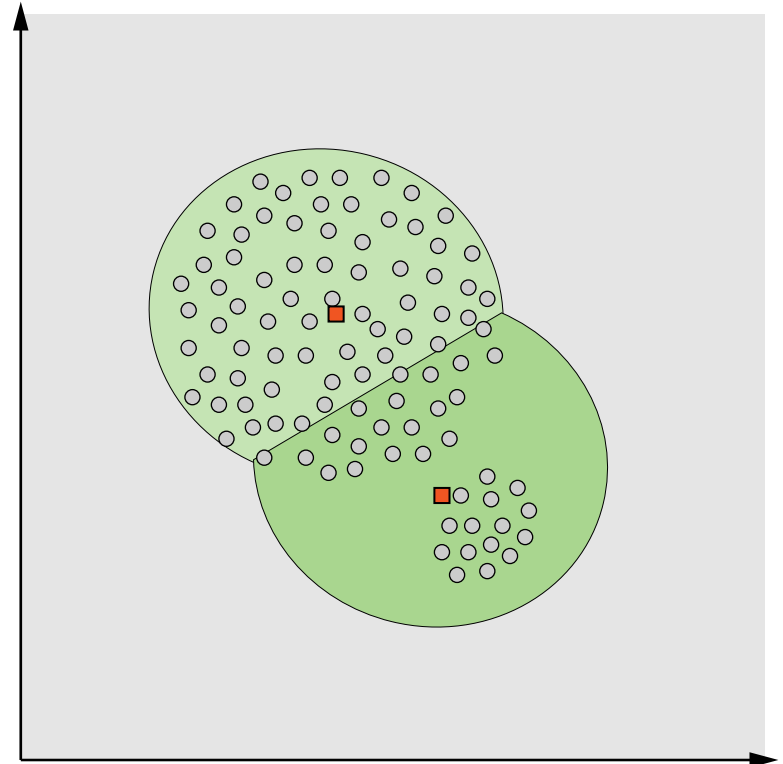
Flat Clustering

Issues with Iterative, Exemplar-based Clustering Algorithms

Algorithms such as k -means fail to detect nested clusters.



Similarly, they fail to detect clusters with large difference in size.

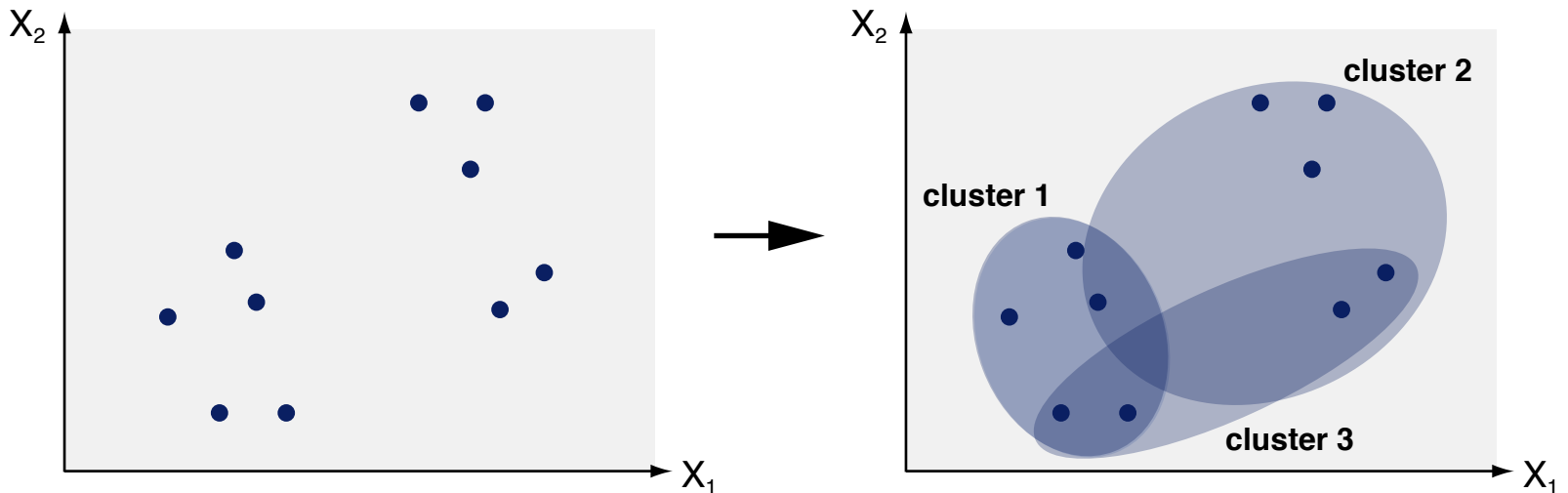


Soft Clustering

Soft Clustering

Soft clustering

- A flat clustering technique that maps instances to overlapping clusters
- **Input.** A set of instances $X = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\}$ without class labels
- **Output.** A set of clusters $C = \{c_1, \dots, c_k\}$ and a mapping $w_j : X \rightarrow [0, 1]$ for each $c_j \in C$, such that $\forall \mathbf{x}^{(i)} \in X : \sum_{j=1}^k w_j^{(i)} = 1$



Number of clusters k

- As for hard clustering, k may be a hyperparameter.

Soft Clustering

Idea and Algorithms

Idea of soft clustering in NLP

- Given the following five texts:

“Maja likes to eat broccoli and bananas.”	→ 1.0 food
“Max had a banana/spinach smoothie for breakfast.”	→ 1.0 food
“Dogs and cats are pets.”	→ 1.0 pets
“Eating food is important for everyone, including cats.”	→ 0.8 food, 0.2 pets
“The hamster munches on a piece of broccoli.”	→ 0.5 food, 0.5 pets

- A soft clustering algorithm might identify two soft clusters:

c_1 representing information on food

c_2 representing information on pets

- It also assigns each text $d^{(i)}$ a weight $w_j^{(i)}$ for each cluster c_j .

Selected soft clustering algorithms

- Fuzzy k -means clustering
- Latent Dirichlet Allocation (detailed here)

Topic Modeling

Topic modeling

- An analysis that identifies $k \geq 2$ topics t_1, \dots, t_k in a text corpus
- Each t is modeled as a list of words (o_1, \dots, o_m) that cooccur in a statistically meaningful way.

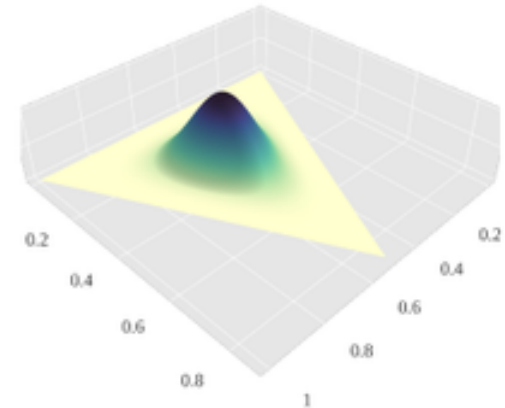


BIRDS NEST TREE
BRANCH LEAVES

Blei and Jordan (2003)

Latent Dirichlet Allocation (LDA)

- An unsupervised soft clustering algorithm for topic modeling
- It learns the latent (say, hidden) structure of topics in texts and of words in topics.



<https://commons.wikimedia.org>

LDA in a nutshell

- Assign each word in each text to the topic from it probably came.
- Repeat assignment until the topic probabilities of words hardly change.
- Return the topic distributions of texts and word distributions of topics.

Topic Modeling

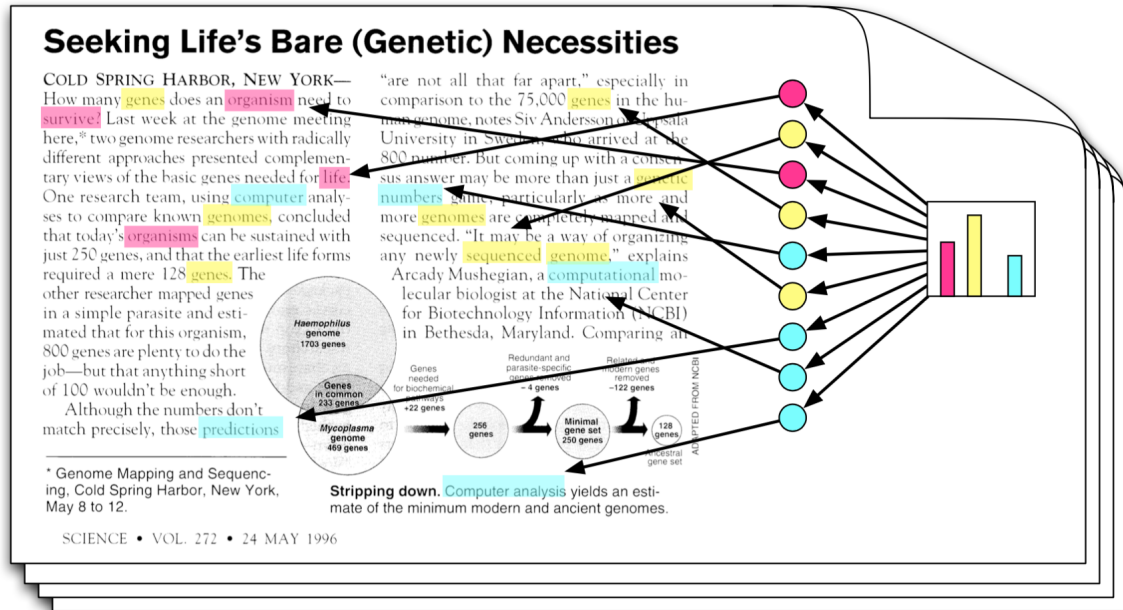
Output of LDA

gene 0.04
dna 0.02
genetic 0.01
...

life 0.02
evolve 0.01
organism 0.01
...

brain 0.04
neuron 0.02
nerve 0.01
...

data 0.02
number 0.02
computer 0.01
...



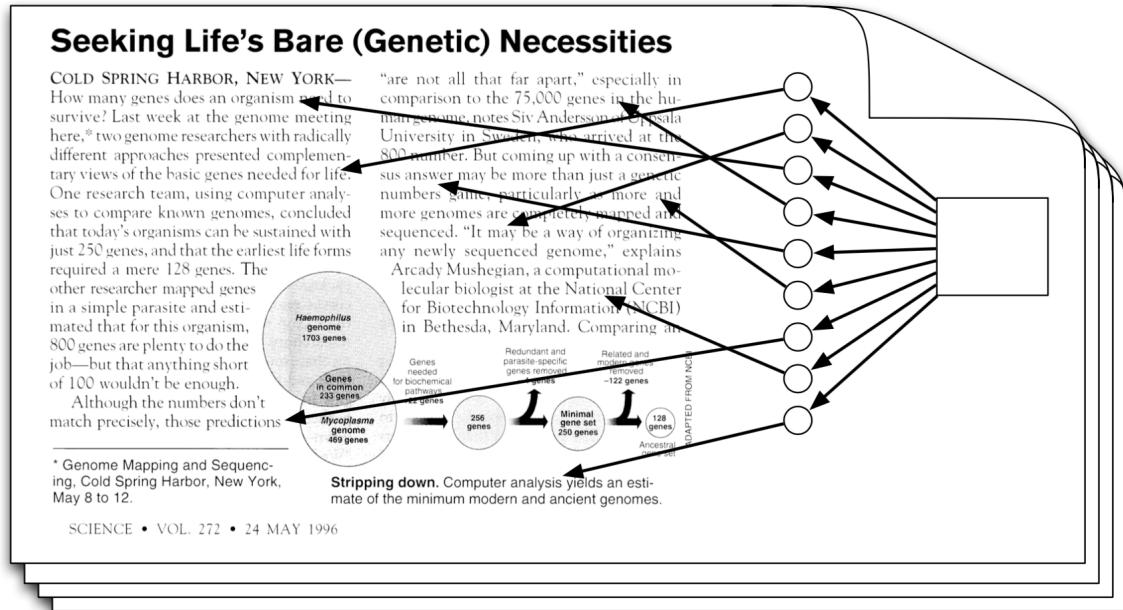
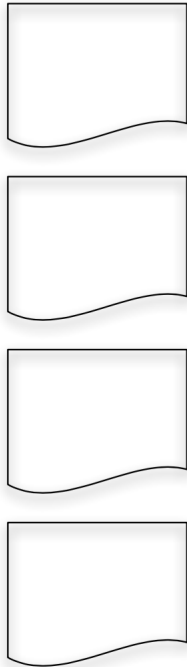
Blei and Jordan (2004)

Output

- Each text d is a weighted combination of corpus-wide topics t_1, \dots, t_k .
- Each topic t_j is represented by a list of words (o_1, \dots, o_m) .
- Each word $o^{(i)}$ in a text d is drawn from one of the topics t_j .

Topic Modeling

Input of LDA



Blei and Jordan (2004)

Input

- A text corpus $D = \{d_1, \dots, d_n\}$
- A number k of topics to be found
- A number m of words to represent each topic with

Topic Modeling

Pseudocode (sketch!)

LDA (**Set**<**String**> D , **int** m , **int** k)

1. Randomly assign each word o in each text $d^{(i)} \in D$ to one topic t_j
2. **repeat**
3. **for** $1 \leq i \leq n$, $1 \leq j \leq k$ **do**
4. double $p(t_j|d^{(i)}) \leftarrow$ Fraction of word in $d^{(i)}$ assigned to t_j
5. **for each** word $o \in \Omega$, $1 \leq j \leq k$ **do** // Vocabulary Ω
6. double $p(o|t_j) \leftarrow$ Fraction of assignments to t_j from o
7. **for** $1 \leq i \leq n$, **each** word o in $d^{(i)}$, $1 \leq j \leq k$ **do**
8. Reassign o to topic t_j with probability $p(t_j|d^{(i)}) \cdot p(o|t_j)$
9. **until** probabilities $p(o|t_j)$ stable
10. **for each** text $d^{(i)}$, $1 \leq j \leq k$ **do** // Get topic weighting
11. double $w_j^{(i)} \leftarrow$ Fraction of words in $d^{(i)}$ from topic t_j
12. **for each** topic t_j , $1 \leq l \leq m$ **do** // Get word lists
13. String $o_j^{(l)} \leftarrow$ The word l -th most often assigned to t_j
14. **return** $(w_1^{(i)}, \dots, w_k^{(i)})$ **for each** $d^{(i)}$, $(o_j^{(1)}, \dots, o_j^{(m)})$ **for each** t_j

Notice (details beyond the scope here)

- For efficiency, LDA is often implemented using *Gibbs sampling*.

Topic Modeling

LDA: Example computation

Example

- Assume we look for $k = 2$ topics (t_1 and t_2) for the corpus from above:

$d^{(1)}$: “Maja likes to eat **broccoli** and bananas.”

$d^{(2)}$: “Max had a banana/**broccoli** smoothie for breakfast.”

$d^{(3)}$: “Dogs and cats are pets.”

$d^{(4)}$: “Eating food is important for everyone, including cats.”

$d^{(5)}$: “The hamster munches on a piece of **broccoli**.”

Initialization (lines 1–6)

- Let 3 of 7 words in $d^{(1)}$ be assigned to t_1 . Then $p(t_1|d^{(1)}) := 3/7 \approx 0.429$.
- Let **broccoli** be assigned 2 times to t_1 , and 20 words in total to t_1 . Then $p(\text{broccoli}|t_1) := 2/20 = 0.1$.

First update (lines 7–8)

- For text $d^{(1)}$, reassign **broccoli** to t_1 with probability $0.429 \cdot 0.1 \approx 0.043$.

Topic Modeling

Case Study

Case study

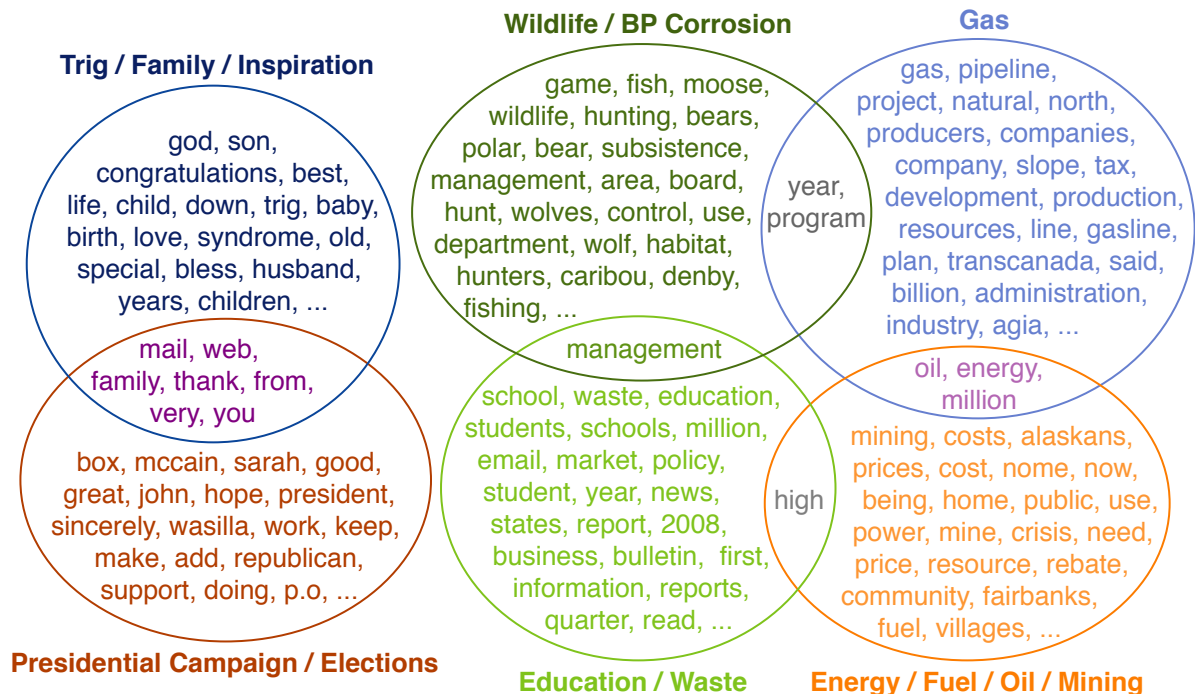
Based on <https://www.r-bloggers.com/2011/06/topic-modeling-the-sarah-palin-emails/>

- **Data.** Thousands of e-mails from Sarah Palin's inbox that were "published" in 2011
- **Goal.** Find main topics covered in the e-mails



LDA topics

(labeled manually)



Topic Modeling

Example Texts with Highlighted Topic Words

0.99 Trig / Family / Inspiration

Hello Governor Palin, Our family wanted to congratulate you and your family on the birth of your son, Trig. Our fourth child, Daniel, was born with Down Syndrome, and we can't imagine our family without him. Recently, I met a mom with a 34-year-old daughter with DS and she said it best: "Don't you feel like you've been chosen to be a member of a very special club?" God bless your family, what a beautiful example of love you are to all who see you! the Paul & Tricia Pietig family, Des Moines, Iowa

0.9 Wildlife / BP Corrosion, 0.1 Presidential Campaign / Election

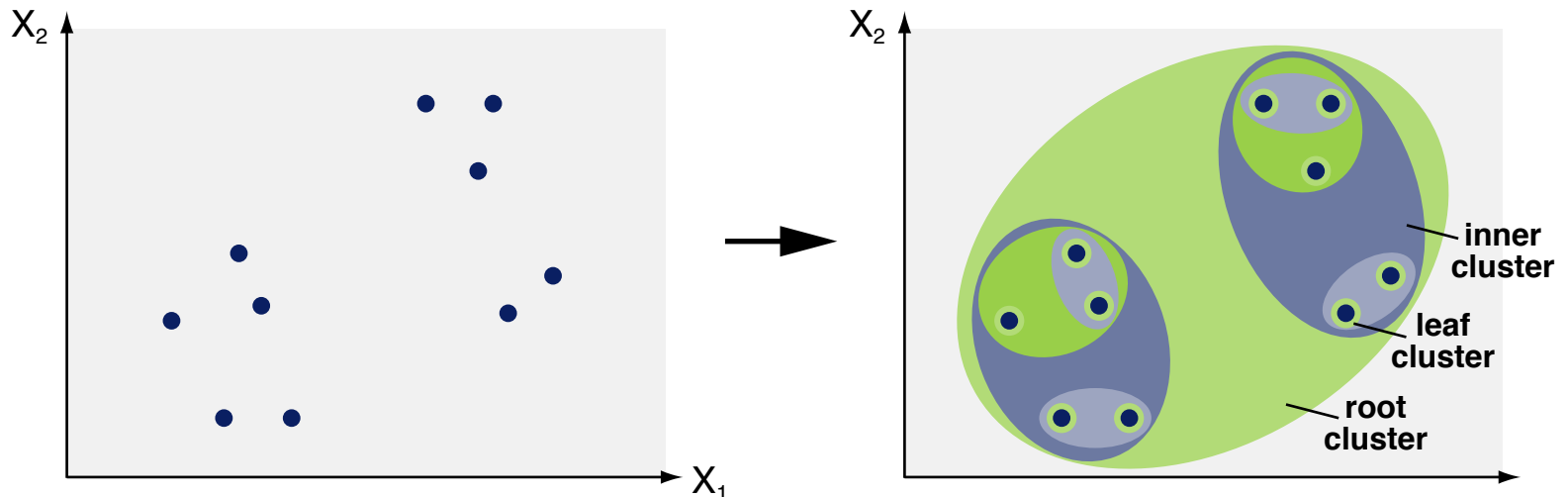
We understand that you have been discussed as a possible choice for the Vice Presidency. As support the democratic process and care about protecting the wildlife for future generations we want you to know that we don't believe people in our states would vote for you for any office if they knew your record on these issues. It is troubling that you are now working more than 50,000 Alaskans a vote on aerial killing of wolves and bears with legislation now being considered in the Alaska legislature.

Hierarchical Clustering

Hierarchical Clustering

Hierarchical clustering

- A technique that creates a binary tree over instances, which represents the sequential merging of instances into clusters
- **Input.** A set of instances $X = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\}$ without class labels
- **Output.** A tree $\langle V, E \rangle$ where each $v \in V$ denotes a cluster of some size, and each $(v_1, v_2) \in E$ that v_2 has been merged into v_1



Notice

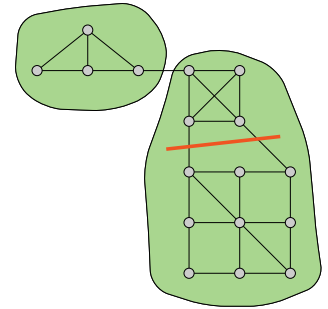
- A flat clustering can be obtained via cuts in the hierarchy tree.

Hierarchical Clustering

Two Main Techniques

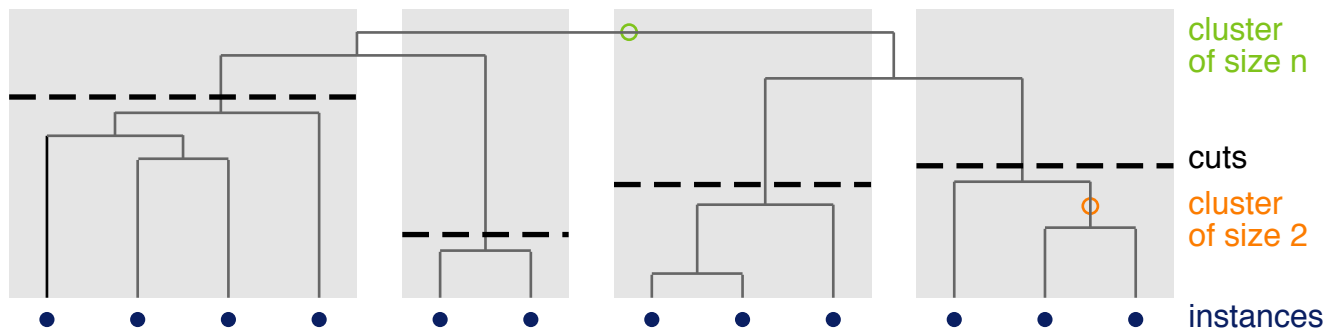
Divisive hierarchical clustering

- Incrementally split clusters into smaller ones (top-down).
- **MinCut**. Model all instances as a weighted graph; split clusters by finding the minimum cut in subgraphs.



Agglomerative hierarchical clustering (in the focus here)

- Incrementally create tree bottom-up, beginning with single instances.
- Merge closest pair of clusters based on the distances of their instances.
- Repeat until only one cluster remains.
- Clusters and their merging may be represented as a *dendrogram*.



Agglomerative Hierarchical Clustering

Signature

- **Input.** A set of instances X
- **Output.** A binary tree $\langle V, E \rangle$ containing all clusters

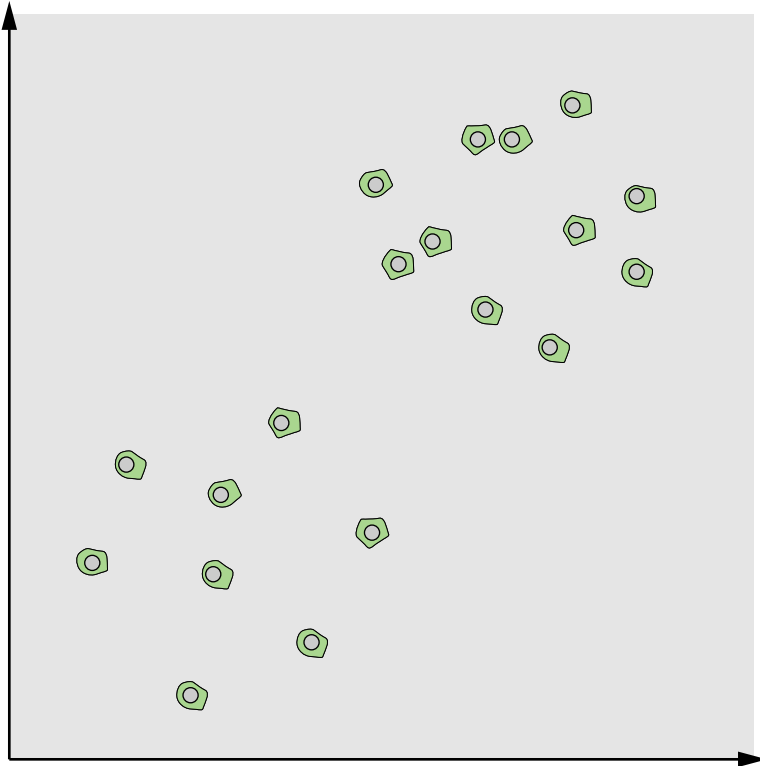
`agglomerativeHierarchicalClustering (Set<Instance> X)`

```
1.  Set<Set<Instance>> clusters  $\leftarrow$   $\{\{\mathbf{x}^{(i)}\} \mid \mathbf{x}^{(i)} \in X\}$  // cur. clusters
2.  Set<Set<Instance>> V  $\leftarrow$  clusters // tree nodes
3.  Set<Set<Instance>[]> E  $\leftarrow$   $\emptyset$  // tree edges
4.  while |clusters| > 1 do
5.      double [][] similarities  $\leftarrow$  updateSimilarities(clusters)
6.      Set<Instance> [] pair  $\leftarrow$  getClosest(clusters, similarities)
7.      Set<Instance> merged  $\leftarrow$  pair[0]  $\cup$  pair[1]
8.      clusters  $\leftarrow$  (clusters  $\setminus$  {pair[0], pair[1]})  $\cup$  {merged}
9.      V  $\leftarrow$  V  $\cup$  {merged}
10.     E  $\leftarrow$  E  $\cup$  {(merged, pair[0]), (merged, pair[1])}
11. return  $\langle V, E \rangle$ 
```

Agglomerative Hierarchical Clustering

Example

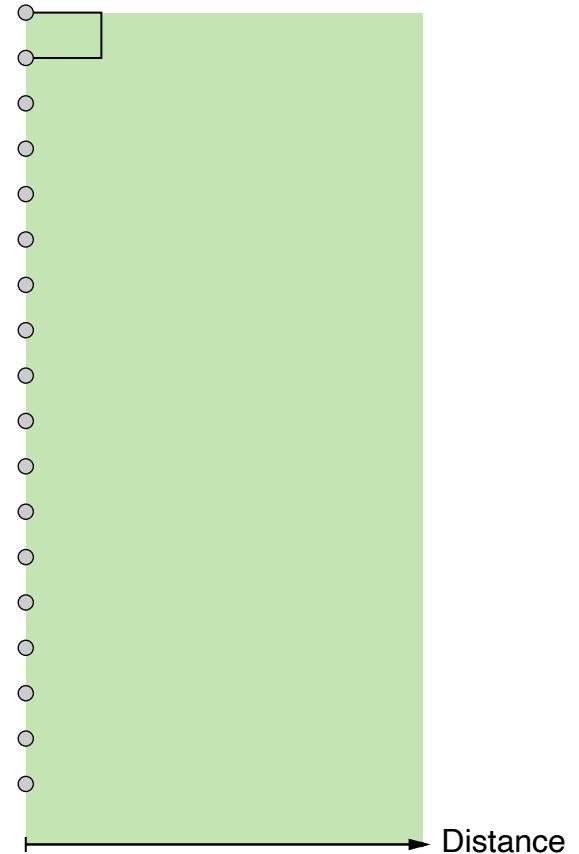
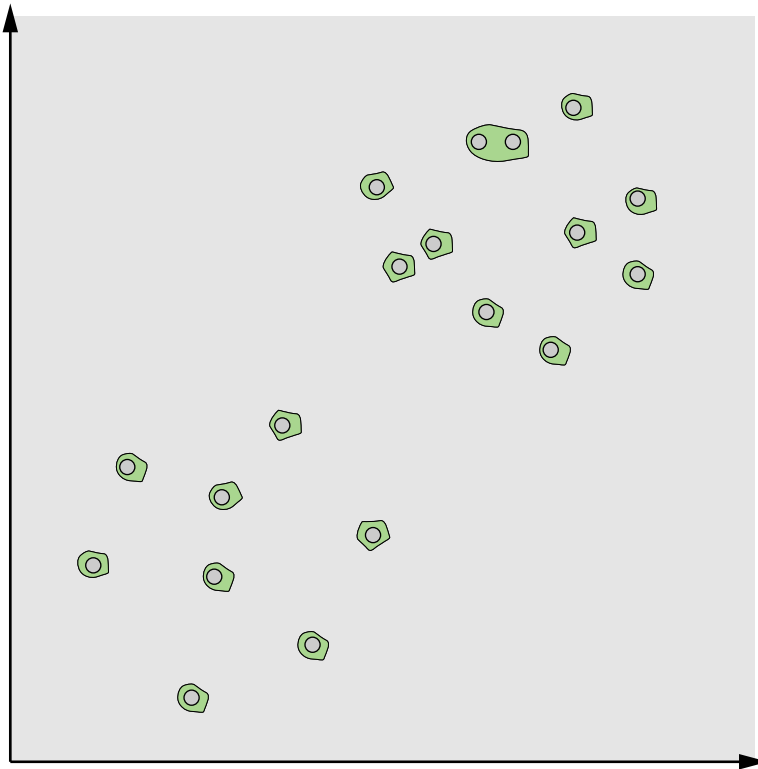
Pseudocode line 1: Assign each instance to an individual cluster.



Agglomerative Hierarchical Clustering

Example

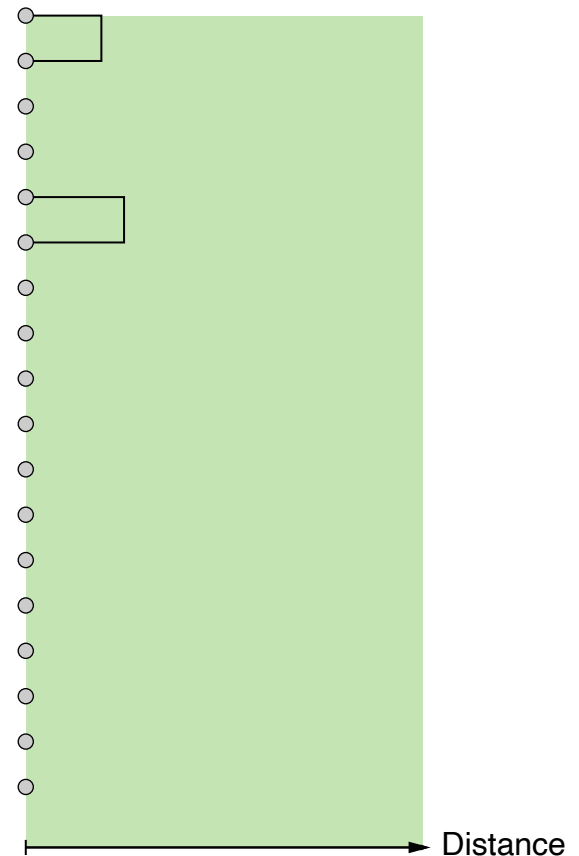
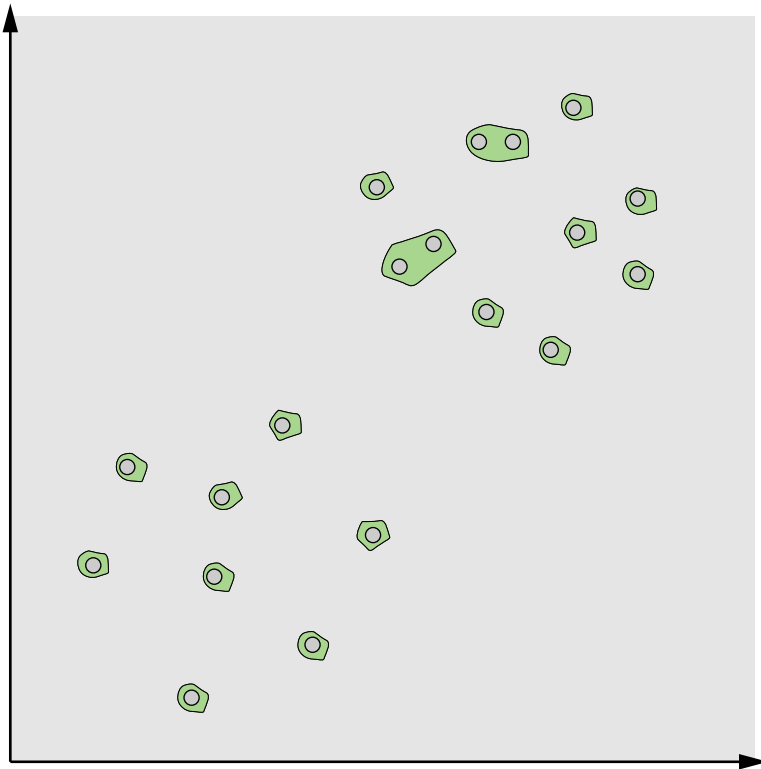
Pseudocode lines 5–10: Combine closest pair of clusters into one cluster.



Agglomerative Hierarchical Clustering

Example

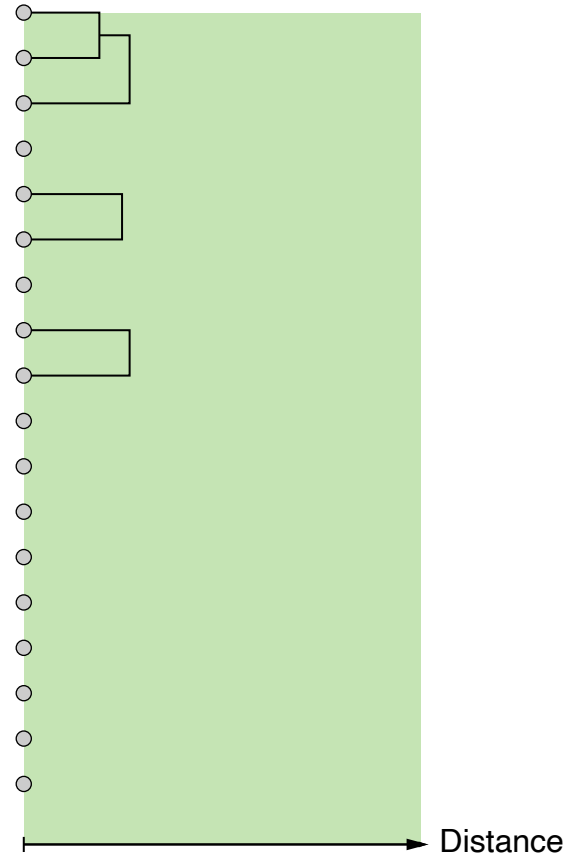
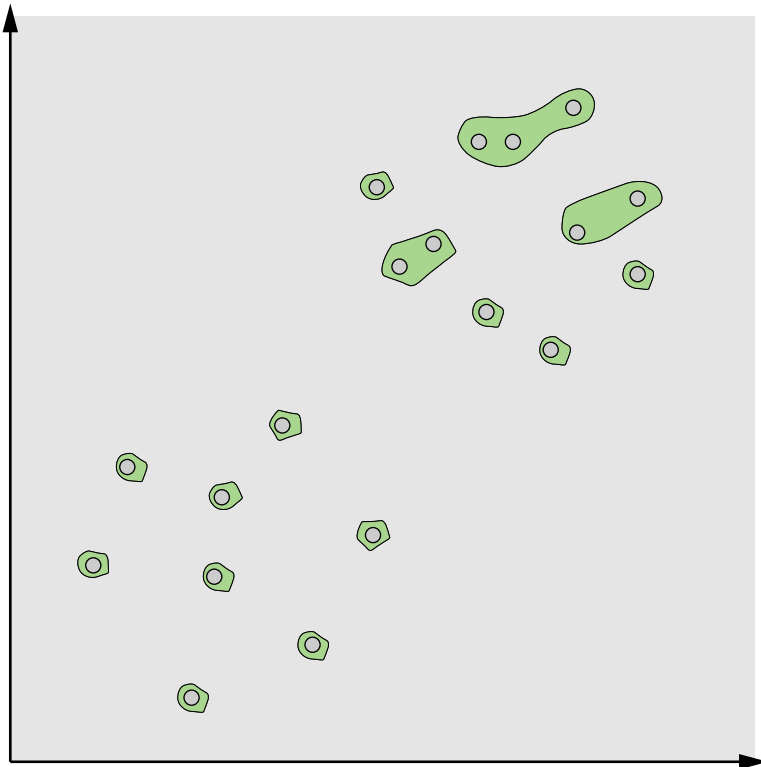
Pseudocode lines 5–10: Repeat until only one cluster remains.



Agglomerative Hierarchical Clustering

Example

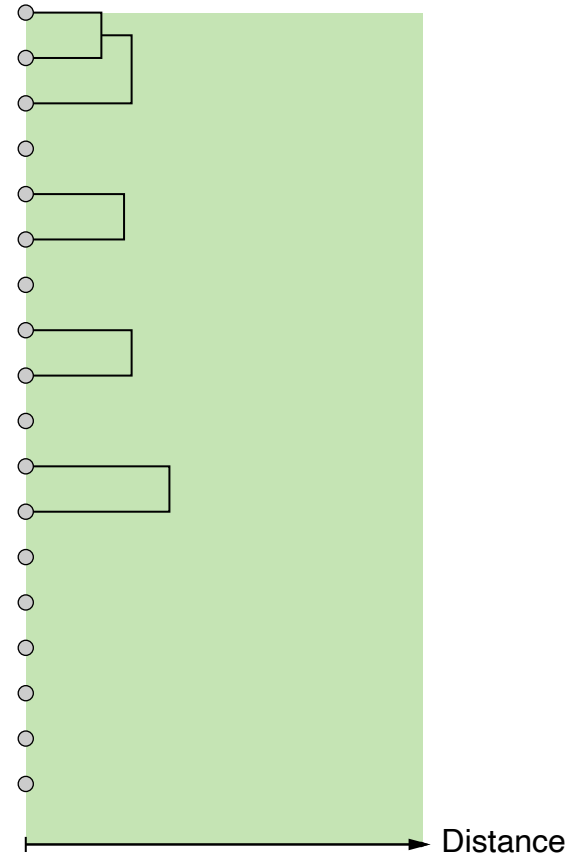
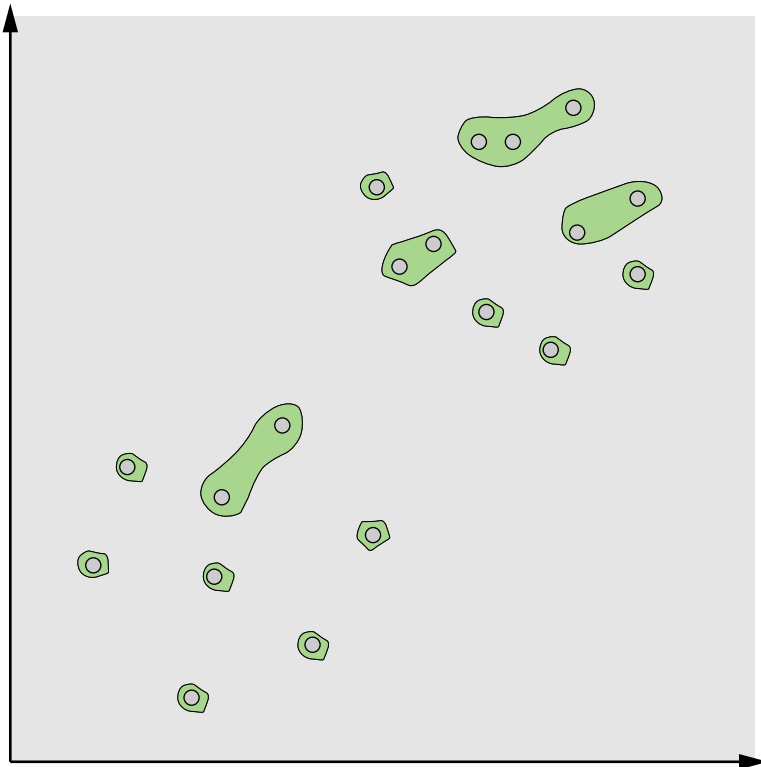
Pseudocode lines 5–10: Repeat until only one cluster remains.



Agglomerative Hierarchical Clustering

Example

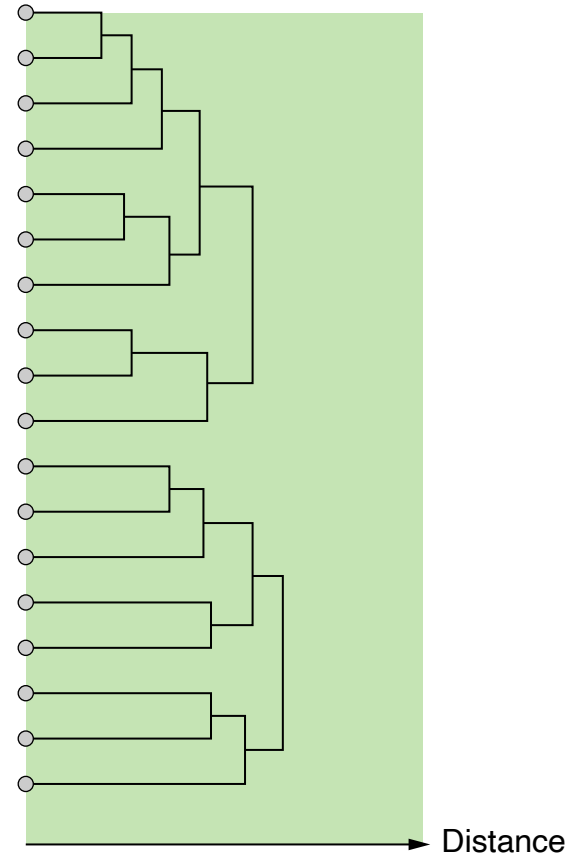
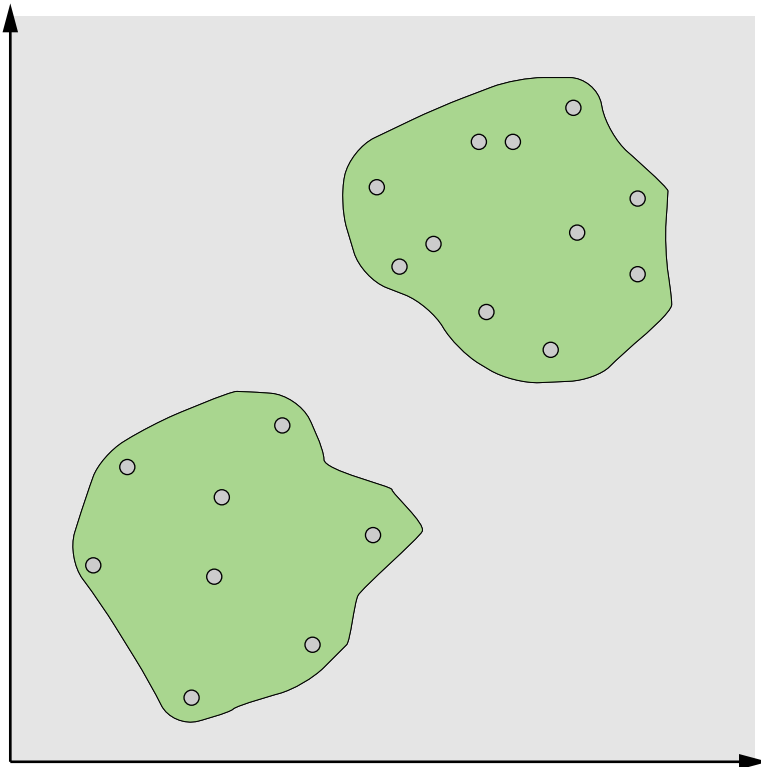
Pseudocode lines 5–10: Repeat until only one cluster remains.



Agglomerative Hierarchical Clustering

Example

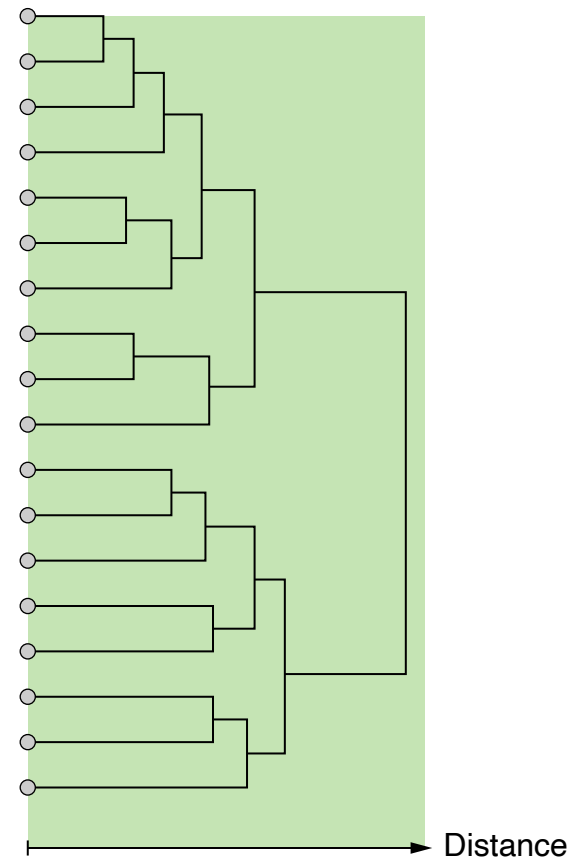
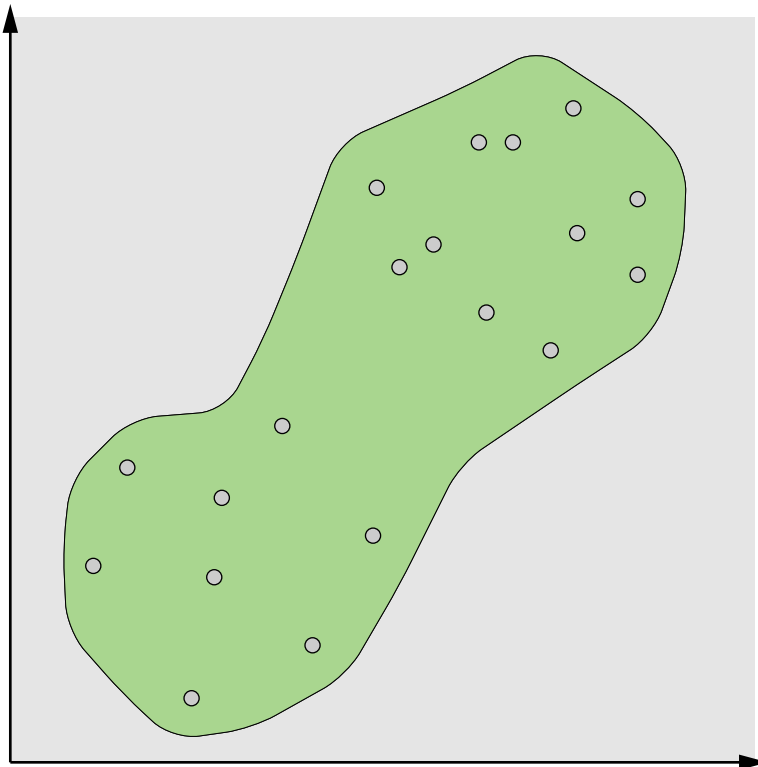
Pseudocode lines 5–10: Repeat until only one cluster remains.



Agglomerative Hierarchical Clustering

Example

Pseudocode line 11: The dendrogram illustrates the final graph. → done!



Agglomerative Hierarchical Clustering

Cluster Similarity

Two levels of similarity

- **Instance-levels.** Similarity of instances (or cluster representatives)
Measures as Lecture Part IV: Cosine, Euclidean, ...
- **Cluster-level.** Aggregation of instance similarity into cluster similarity

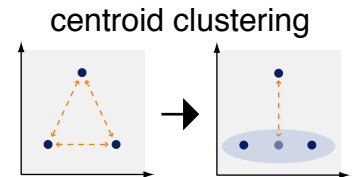
How to aggregate similarity?

- Different measures for the similarity of two clusters exist.
- They may result in fully different clusterings.
- **Examples.** *Single link, complete link, group-average link*

Why not centroid clustering?

- Centroid similarity is *non-monotonous*, i.e., larger clusters may be more similar to other clusters than their sub-clusters.

Other non-monotonous measures exist, e.g., *median distance*.



Agglomerative Hierarchical Clustering

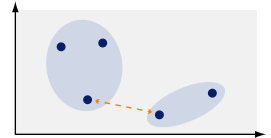
Cluster Similarity Aggregation Methods

Single link clustering

- Use the nearest neighbors across two clusters c, c' .

$$sim(c, c') = \max_{\mathbf{x} \in c, \mathbf{x}' \in c'} sim(\mathbf{x}, \mathbf{x}')$$

single link

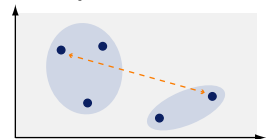


Complete link clustering

- Use the furthest neighbors across two clusters c, c' .

$$sim(c, c') = \min_{\mathbf{x} \in c, \mathbf{x}' \in c'} sim(\mathbf{x}, \mathbf{x}')$$

complete link

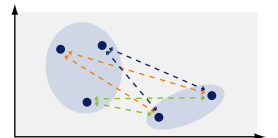


Group-average link clustering

- Average over all similarities of two clusters c, c' .

$$sim(c, c') = \frac{1}{|c| \cdot |c'|} \sum_{\mathbf{x} \in c, \mathbf{x}' \in c'} sim(\mathbf{x}, \mathbf{x}')$$

group-average link



Review Sentiment Analysis

Sentiment analysis

- The text analysis that predicts whether a text (span) conveys sentiment
- An extensively studied downstream task in NLP, industrially important
- Usually tackled with supervised classification

Sentiment polarity vs. scores

- **Polarity.** *Positive* or *negative*, possibly also *neutral* etc.
- **Scores.** Numeric scale, e.g., $\{1, \dots, 5\}$ or $[0, 1]$



<https://publicdomainpictures.net>

Reviews

- Written consumer judgments of products, services, and works of arts.
For example, reviews of books, movies, hotels, devices, etc.
- Reviews often comprise several “local” sentiments on different aspects.

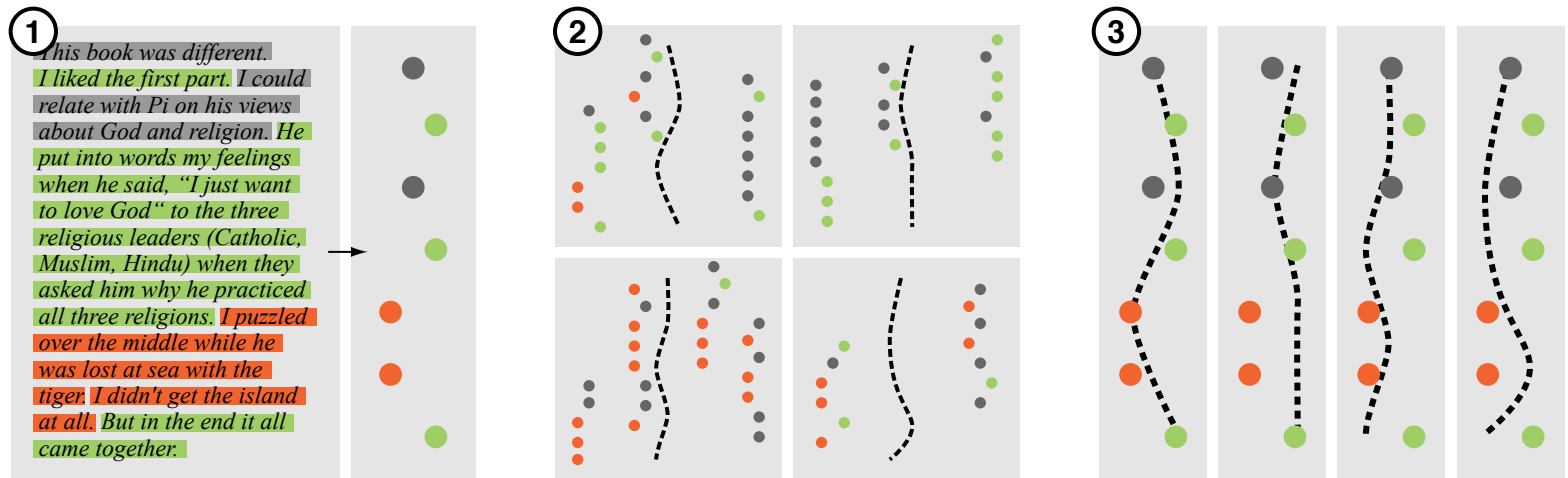
This book was different. I liked the first part. I could relate with Pi on his views about God and religion. He put into words my feelings when he said, “I just want to love God” to the three religious leaders (Catholic, Muslim, Hindu) when they asked him why he practiced all three religions. I puzzled over the middle while he was lost at sea with the tiger. I didn't get the island at all. But in the end it all came together.

Review Sentiment Analysis

Sentimen Flow Patterns (Wachsmuth et al., 2017)

Sentiment flow patterns as features

1. Represent a review by its sequential flow of local sentiment.
2. Cluster known training flows to identify a set of *flow patterns*.
3. Analyze unknown flow based on its similarity to each pattern.



Hypotheses (both evaluated in later lecture parts)

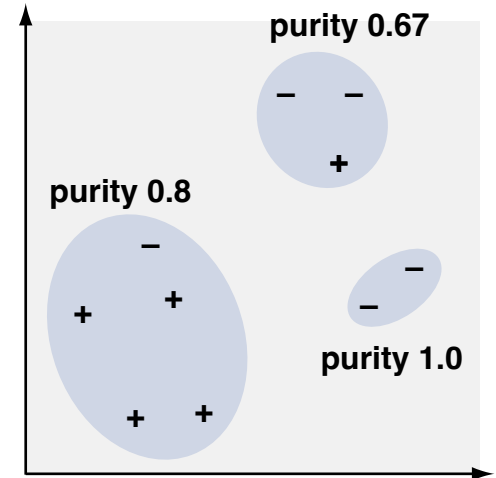
- Similar flows indicate similar global sentiment.
- Similar flow patterns occur across review domains.

Review Sentiment Analysis

How to Obtain Flow Patterns?

Supervised clustering

- Cluster instances with known classes.
- Measure *purity* of clusters, i.e., the fraction of instances whose class is the majority class.
- Ensure all clusters have a minimum purity τ .



Clustering flows

1. Length-normalize all local sentiment flows from a training set.
2. Hierarchically cluster the normalized flows to obtain a binary tree.
3. Obtain flat clusters by finding the cuts closest to the tree's root that create clusters with purity $\geq \tau$.

This maximizes the mean cluster size and, hence, commonness of the patterns.

Obtaining flow patterns

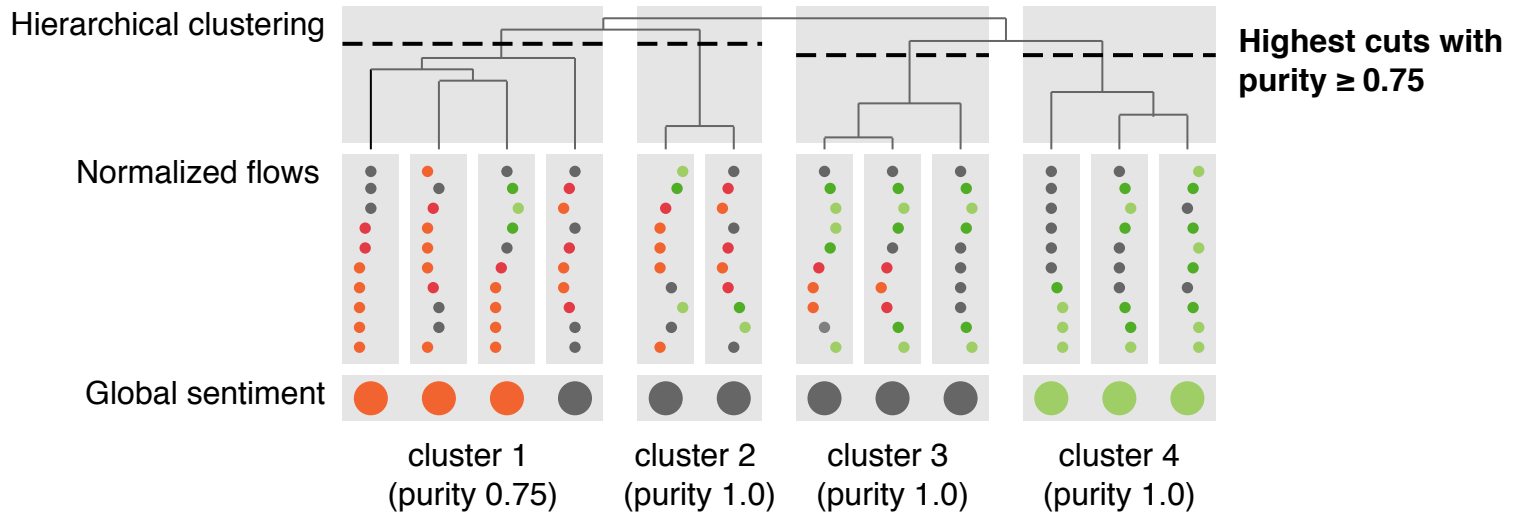
- The centroid of each cluster adequately serves as a flow pattern.

Small clusters might be discarded before, e.g., those of size 1.

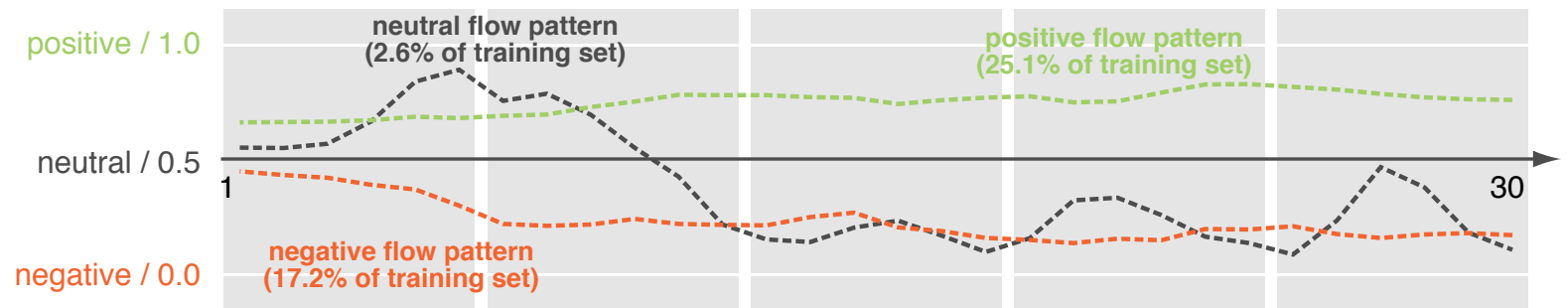
Review Sentiment Analysis

Example: Sentiment Flow Patterns using Clustering

Flow clustering for $\tau = 0.75$



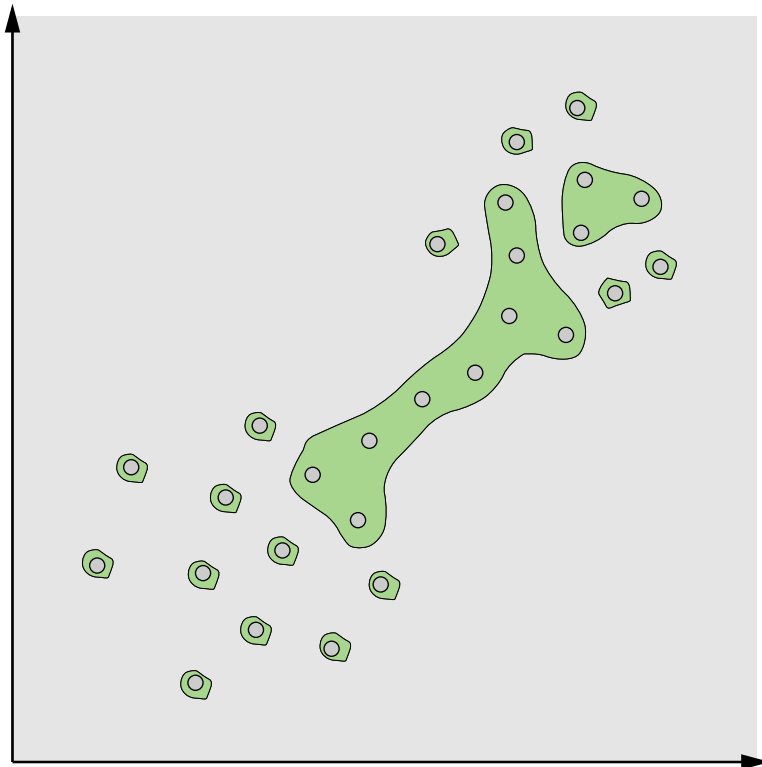
Most common flow patterns in 900 TripAdvisor reviews



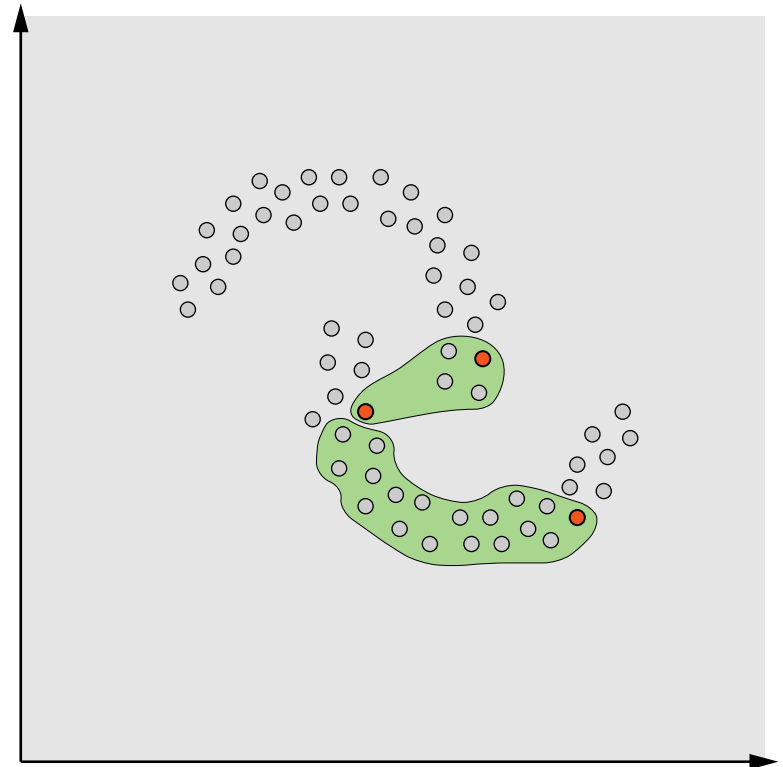
Hierarchical Clustering

Issues with Hierarchical Clustering Algorithms

Chaining problem of clustering using single-link similarity



Nesting problem of clustering using complete-link similarity

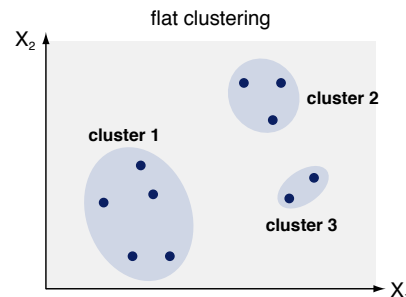


Conclusion

Conclusion

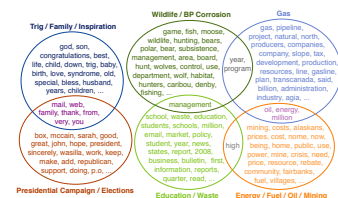
NLP using clustering

- Mostly unsupervised learning of text properties
- Targets situations where no ground truth is available
- Always based on similarity/distance measures



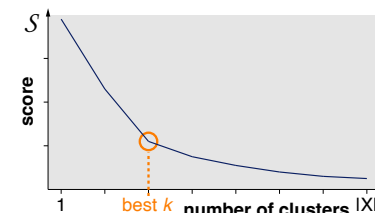
Clustering techniques

- Flat clustering models disjunct classes of instances.
- Soft clustering models weighted class overlaps.
- Hierarchical clustering stepwise organizes instances.



Evaluation of clustering

- Often hard to assess which clustering is optimal
- Elbow and silhouette analysis for intrinsic evaluation
- Ground truth enables extrinsic measures like purity



References

Some content and examples taken from

- **Blei (2012)**. David J. Blei. Probabilistic Topic Models. Tutorial at the 29th International Conference on Machine Learning, 2012.
https://www.khoury.northeastern.edu/home/vip/teach/DMcourse/5_topicmodel_summ/UIntrotoTopicModelsBlei2011-5.pdf
- **Blei and Jordan (2012)**. David J. Blei and Michael I. Jordan. Modeling Annotated Data. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pages 127–234, 2003.
- **Mansoorizadeh et al. (2016)**. Muharram Mansoorizadeh, Mohammad Aminian, Taher Rahgooy, and Mehdy Eskandari. Multi Feature Space Combination for Authorship Clustering. In Working Notes of the CLEF 2016 Evaluation Labs, pages 932–938, 2016.
- **Sari and Stevenson (2016)**. Yunita Sari and Mark Stevenson. Exploring Word Embeddings and Character N-Grams for Author Clustering. In Working Notes of the CLEF 2016 Evaluation Labs, pages 989–991, 2016.
- **Stein and Lettmann (2025)**. Benno Stein and Theodor Lettmann. Data Mining. Lecture Slides, 2025. <https://webis.de/lecturenotes.html>
- **Wachsmuth (2015)**. Henning Wachsmuth. Text Analysis Pipelines — Towards Ad-hoc Large-scale Text Mining. LNCS 9383, Springer, 2015.

References

Some content and examples taken from

- **Wachsmuth and Stein (2017)**. Henning Wachsmuth and Benno Stein (2017). A Universal Model of Discourse-Level Argumentation Analysis. Special Section of the ACM Transactions on Internet Technology: Argumentation in Social Media, 17(3):28:1–28:24.