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

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

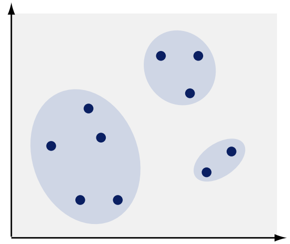
Clustering

Types of Clustering Techniques

Hard vs. soft clustering

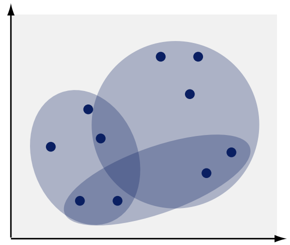
- Hard. Create a partition, such that each instance $\mathbf{x}^{(i)}$ belongs to a single cluster c_j.

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



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

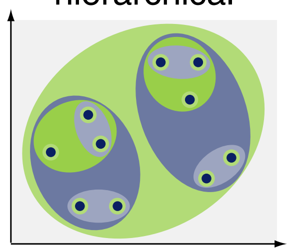
$$\{1,2,3,4\} \quad \rightarrow \quad c_1 = (1, 0.6, 0.8, 0), \;\; c_2 = (0, 0.4, 0.2, 1)$$



Flat vs. hierarchical clustering

- Flat. Both types above simply group a set of instances into a set of clusters.
- Hierarchical. Create a binary tree over all instances where each node represents a cluster of a certain size.

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



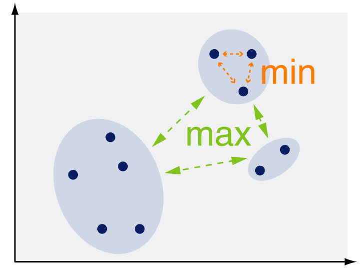
Clustering

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, mininimize between



Why clustering in NLP?

- Particularly targets situations where the set of classes is unknown
- The main goal is often to find out what classes exist.

Selected applications in NLP

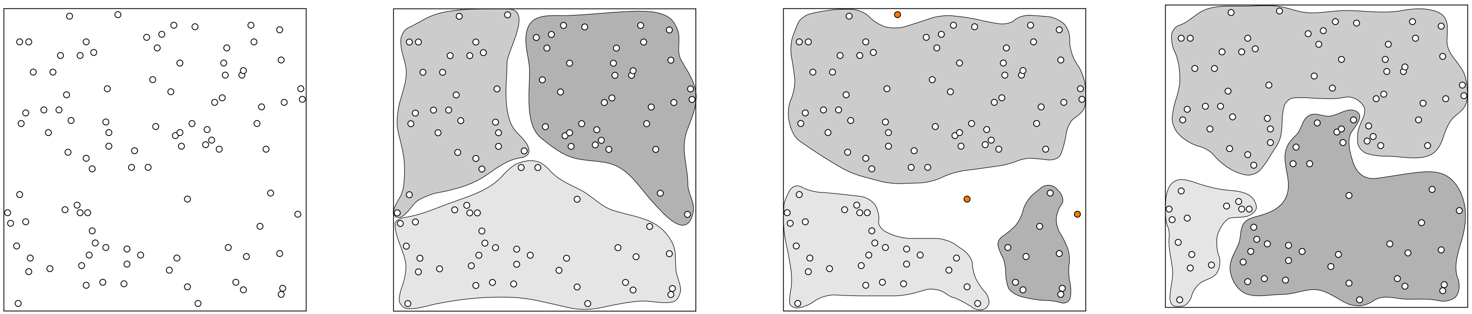
- Topic detection. Identifying the topics covered in a text corpus
- Text retrieval. Finding texts that share similar properties

  For example, similar in terms of author, structure, genre, or similar

Evaluation of Clustering

The clustering evaluation problem

- Various possible ways to cluster a set of instances exist.
- Without a ground truth, the evaluation of cluster quality is often hard.



Set of instances | k-means | DBSCAN | Complete link

Main evaluation goals

- Rank alternative clusterings by quality.
- Determine the ideal number of clusters k.

Types of evaluation

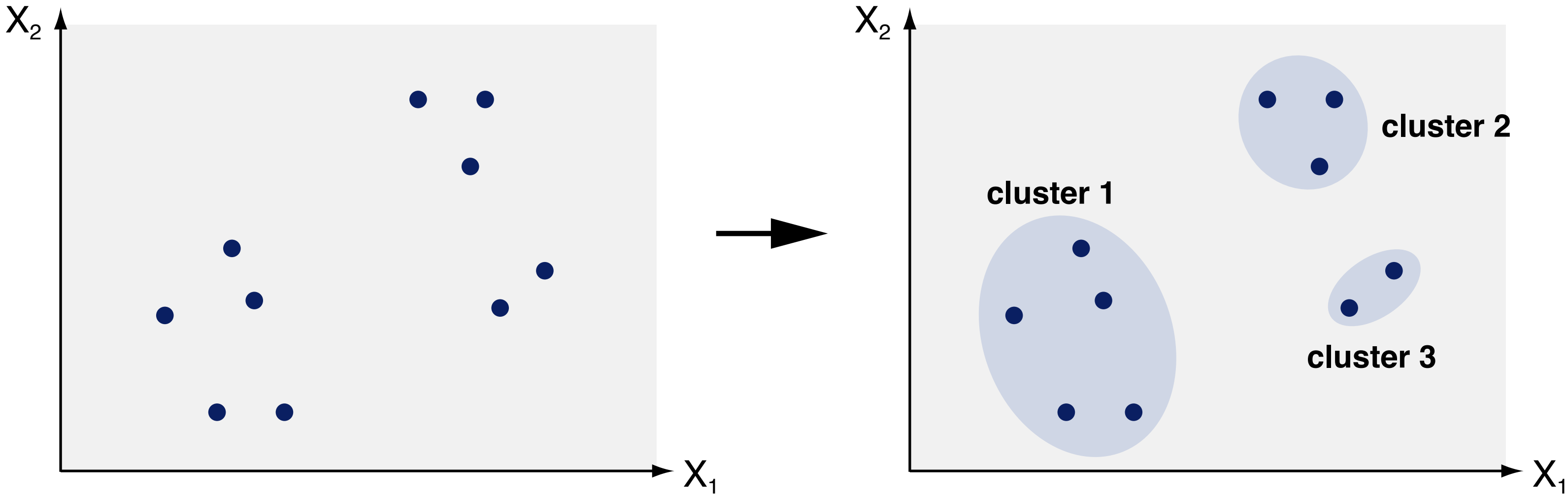
- Intrinsic. Quantify cluster quality based on size, shape, and/or distance
- Extrinsic. Given a test set, compare different clusterings

Flat Clustering

Flat Clustering

Flat (hard) clustering

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



Number of clusters k

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

Flat Clustering

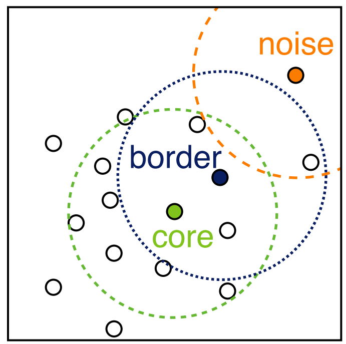
Two Main Types of Algorithms

Iterative algorithms

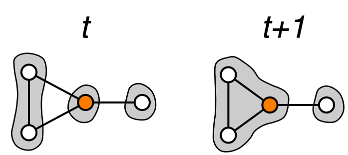
- Iterative clustering and re-assignment of instances to clusters
- Exemplar-based (e.g., k*-means*). Instances are considered in isolation when adding them to clusters.

  We focus on this type here.
- Exchange-based (e.g., *Kernighan-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

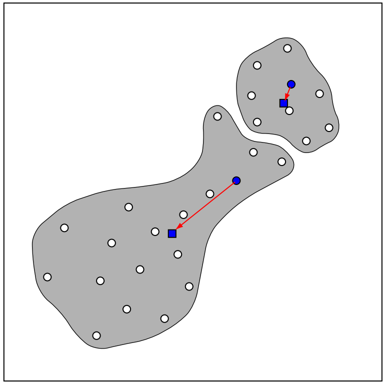
- The mean of all instances in a cluster, i.e., the average of their vectors.

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 chosen based on domain knowledge or based on evaluation measures (see below).

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

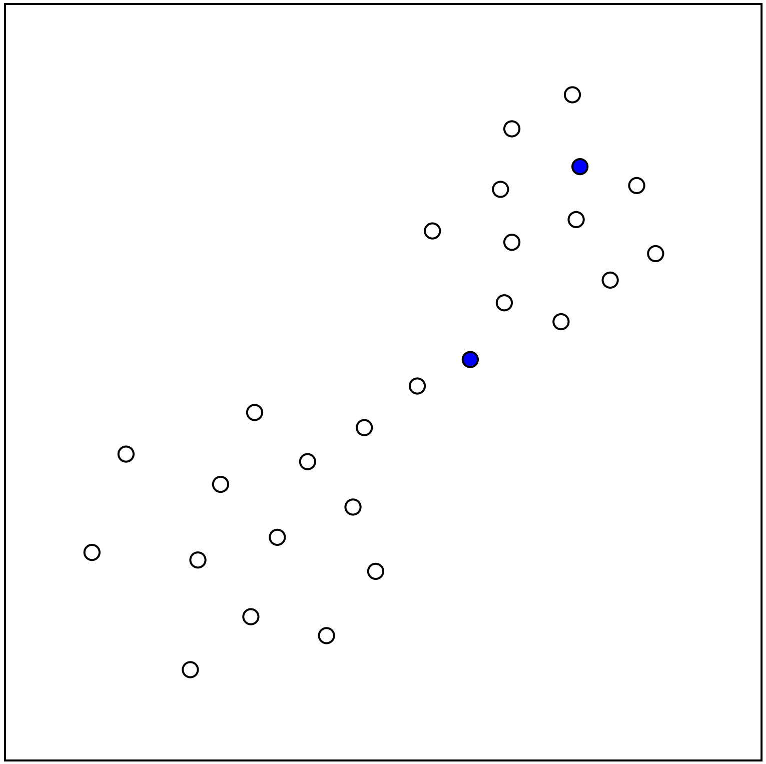
kMeansClustering(Set<Instance> X, int k)

```
Set<Instance> [] clusters ← ∅
Instance [] centroids ← chooseRandomInstances(X, k)
repeat
    Instance [] prevCentroids ← centroids
    for int i ← 1 to k do clusters[i] ← ∅
    for each x ∈ X do // create clusters
        int z ← 1
        for int j ← 2 to k do // find nearest centroid
            if sim(x,centroids[j]) > sim(x,centroids[z]) then z←j
        clusters[z] ← clusters[z] ∪ {x}
    for int i ← 1 to k do // update centroids
        centroids[i] ← computeMean(clusters[i])
until prevCentroids = centroids // convergence
return clusters
```

Flat Clustering with k-means

Example: 2-means

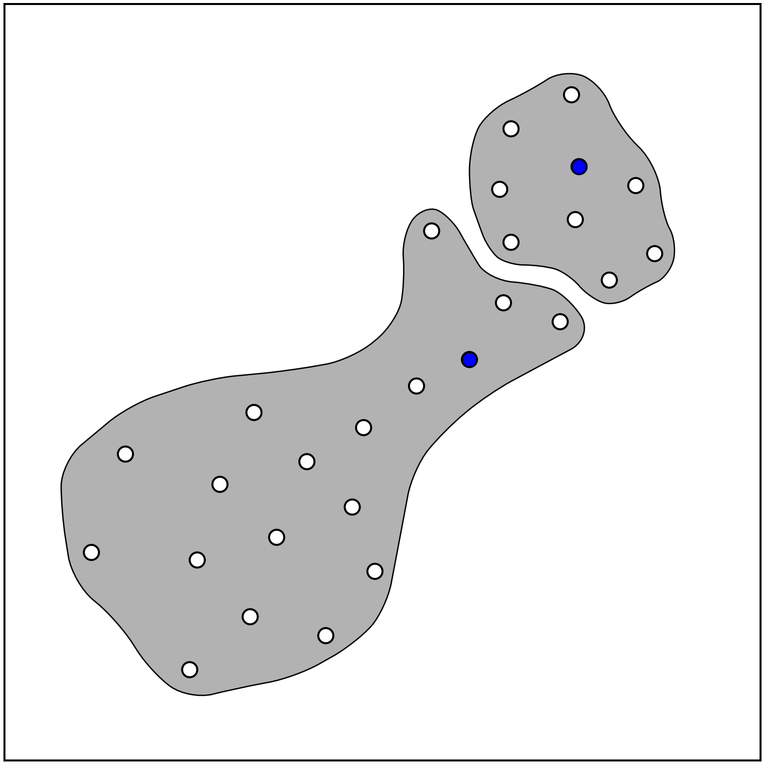
Line 2: Choose k instances randomly as centroids



Flat Clustering with k-means

Example: 2-means

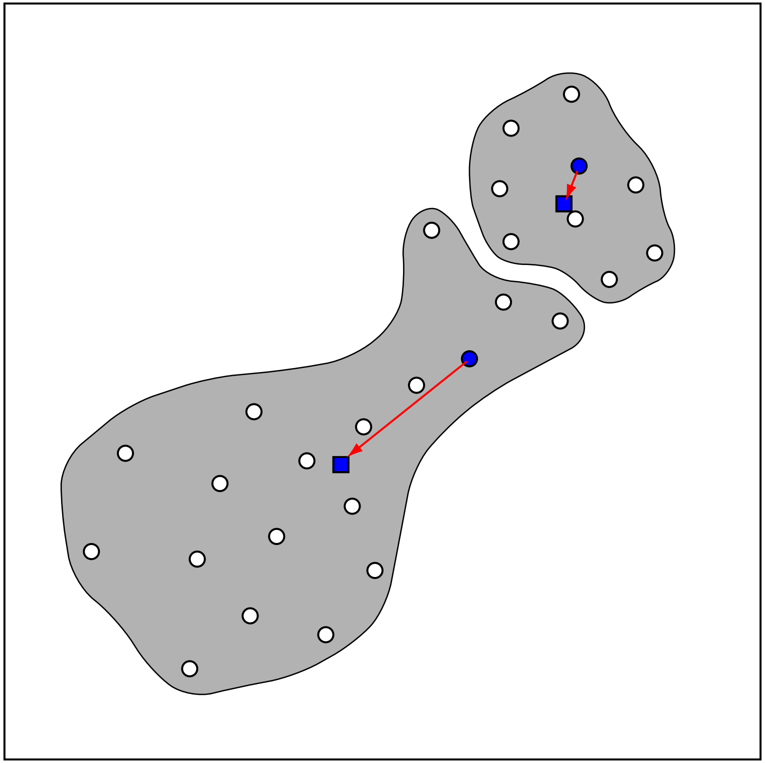
Lines 5–10: Cluster by distance to the k centroids



Flat Clustering with k-means

Example: 2-means

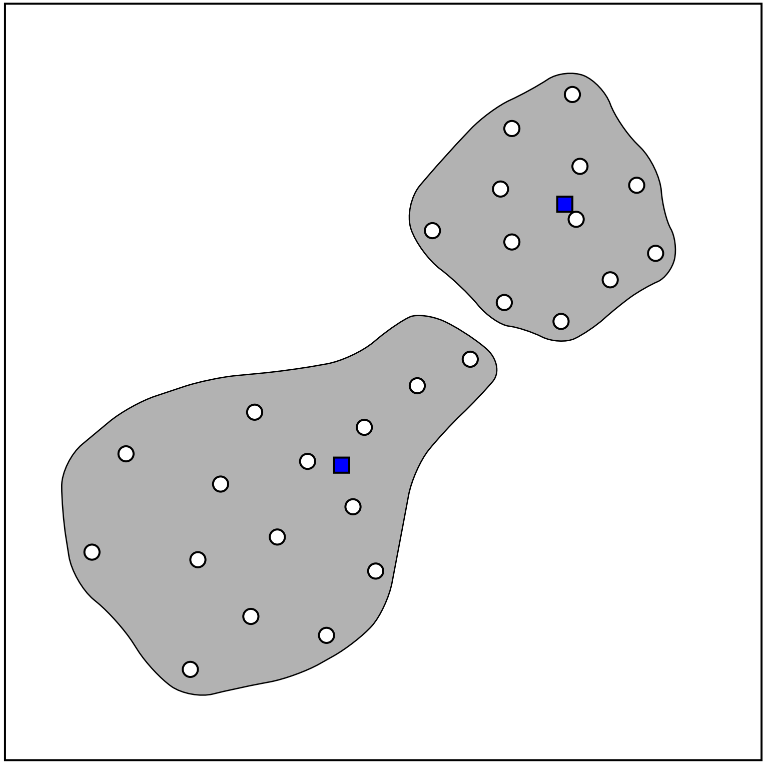
Lines 11–12: Recompute centroids of the k clusters



Flat Clustering with k-means

Example: 2-means

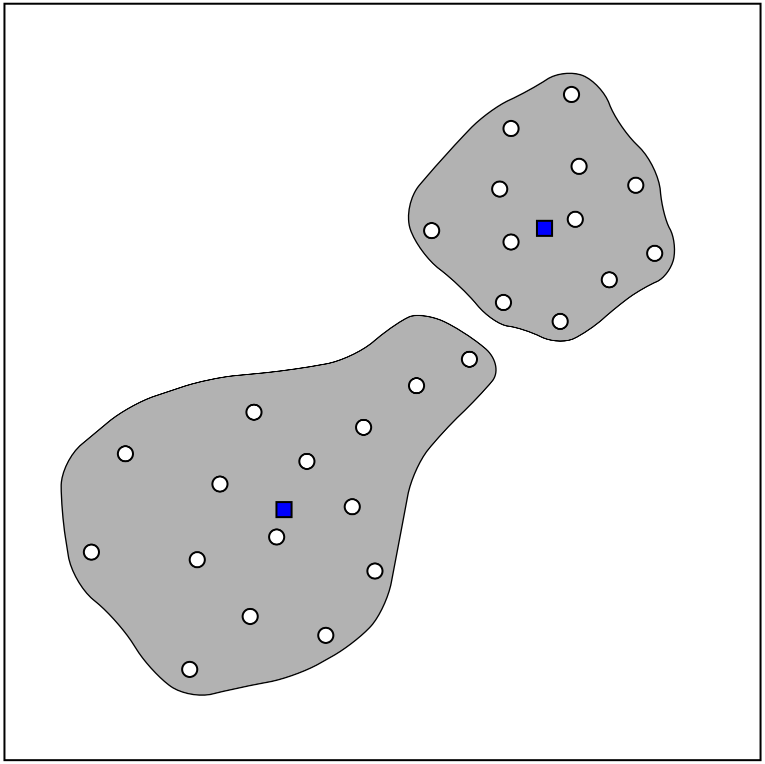
Repeat until convergence (lines 5–10 again)



Flat Clustering with k-means

Example: 2-means

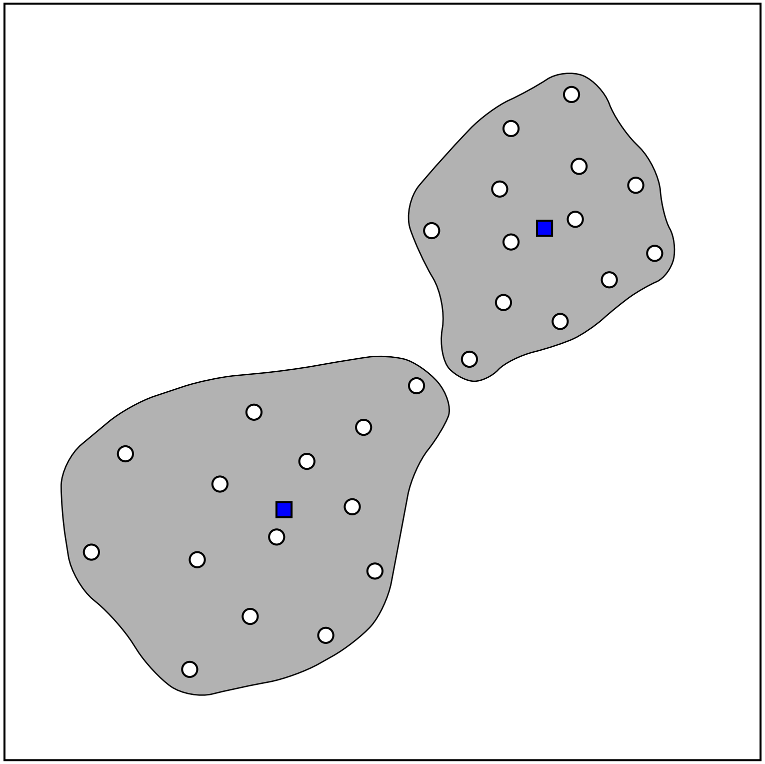
Repeat until convergence (lines 11–12 again)



Flat Clustering with k-means

Example: 2-means

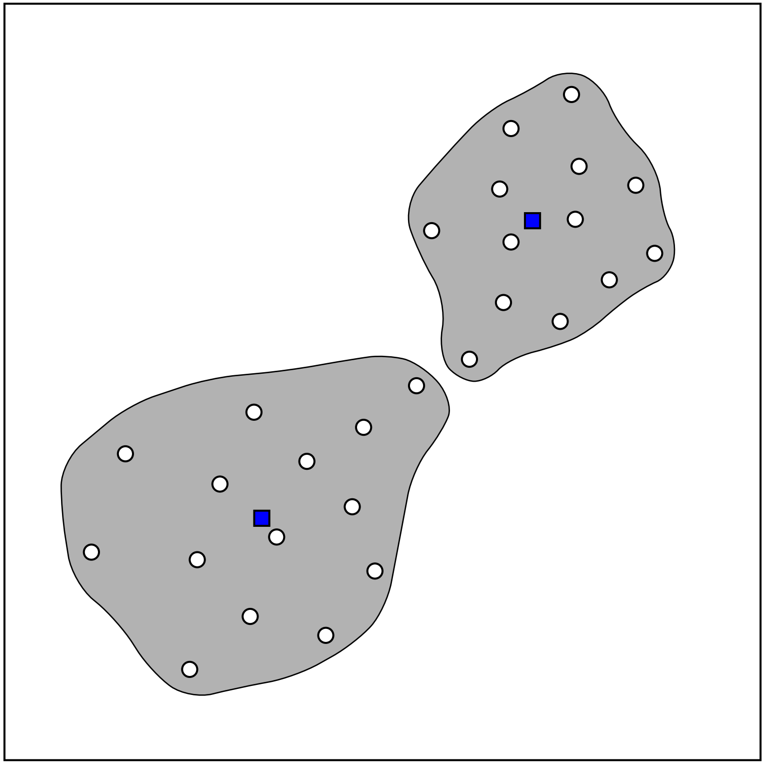
Repeat until convergence (lines 5–10 again)



Flat Clustering with k-means

Example: 2-means

Convergence!



Evaluation of Clustering

Choice of the number of clusters

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

Example cost functions

- Intrinsic. Squared distances of instances to centroid $\rightarrow$ 0.0 for $k = |X|$
- Intrinsic. Maximum cluster size $\rightarrow$ highest for $k = 1$ (cost 0.0)
- Intrinsic. Maximum cluster distance $\rightarrow$ highest for $k = |X|$ (cost 0.0)
- Extrinsic. B^3 F_1-score $\rightarrow$ 1.0 for $k = |X|$
- Extrinsic. Purity of clusters $\rightarrow$ 1.0 for $k = |X|$

Common intrinsic evaluation measures

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

  Both measure have a visual intuition, but work mathematically.

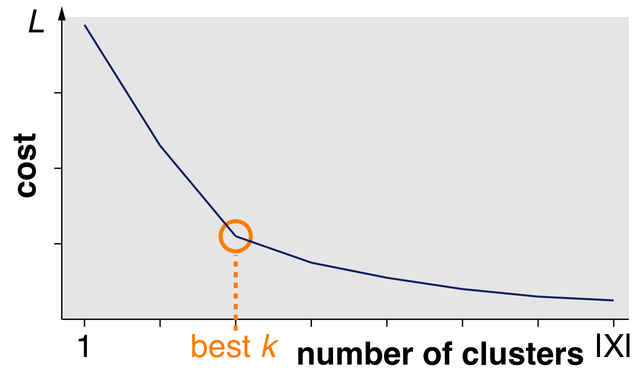
Evaluation of Clustering

Elbow Criterion

Elbow criterion

- A method to find the best value of a hyperparameter, e.g., k in k-means
- Requires some cost function $\mathcal{L}$

  For example, the average cluster similarity



Input

- A *set* of clusterings $\mathbf{C} = \{C_1, \ldots, C_p\}$ for hyperparameter values $k_1, \ldots, k_p$
- A cost $\mathcal{L}(C_i)$ for each clustering C_i

Approach

- Visually. Pick the k where the curve has an "elbow".

  Problem: Not all curves have a clear elbow.

- Computationally. Pick the k with the maximum second derivate:

  This reflects the point where the cost reduction changes strongest.

$$k \quad := \quad \text{argmax}_i(\ \mathcal{L}(C_{i-1})\ -\ 2 \cdot \mathcal{L}(C_i)\ +\ \mathcal{L}(C_{i+1})\ )$$

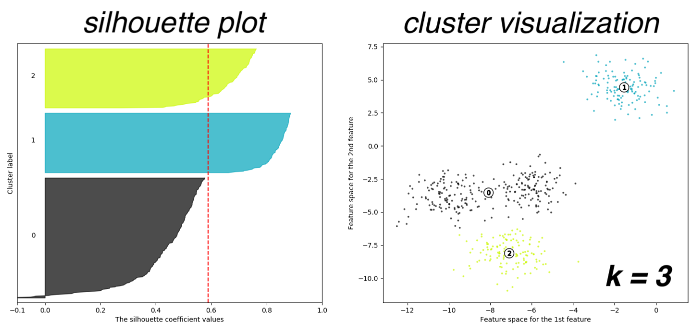
Evaluation of Clustering

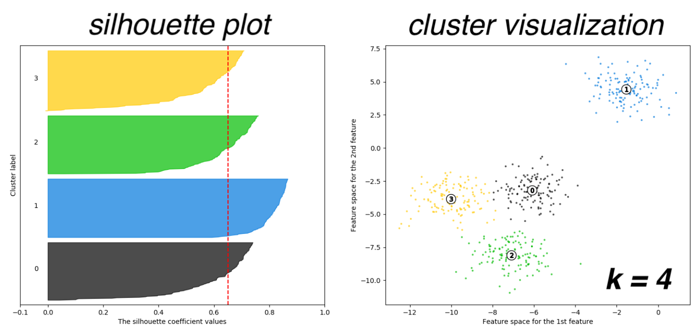
Silhouette Analysis

Silhouette analysis

- A method to find the best number of clusters k in clustering
- Computes a score in $[-1, 1]$ for each cluster of a clustering that reflects how close each instance is to instances from other clusters

  $\sim$*1:* Far away $\sim$*0:* At the boundary to other clusters $<$*0:* Possibly in wrong cluster





Approach

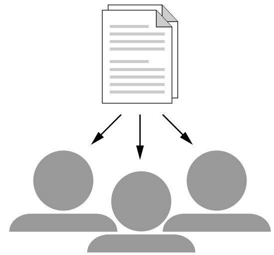
- Visually. Pick the k where most scores (x-axis) are above average, and where the cluster size (y-axis) is balanced.

  Multiple similar candidates might be hard to choose between.

- Computationally. Pick k with maximum average score (vertical red line).

Authorship Attribution

Authorship attribution



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

Observations

- Unlike in most 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

- Concept. Teams compete with approaches on the same task and 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 participants used k-means, including an estimation of the best k.
- The others identified 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.

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**
Vartapetiance and Gillam	0.195	**0.935**	0.234	03:03:13

B^3 precision and recall of a text d

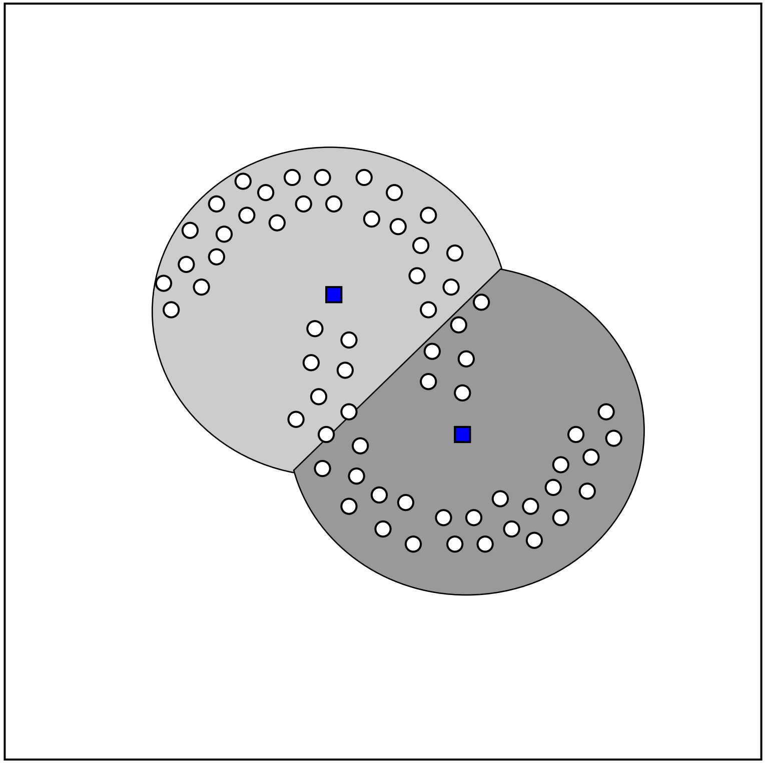
- B^3 precision. Proportion of texts in the cluster of d by the author of d.
- B^3 recall. Proportion of texts by the author of d found in the cluster of d.

The values are averaged over all texts. F_1-score as usual.

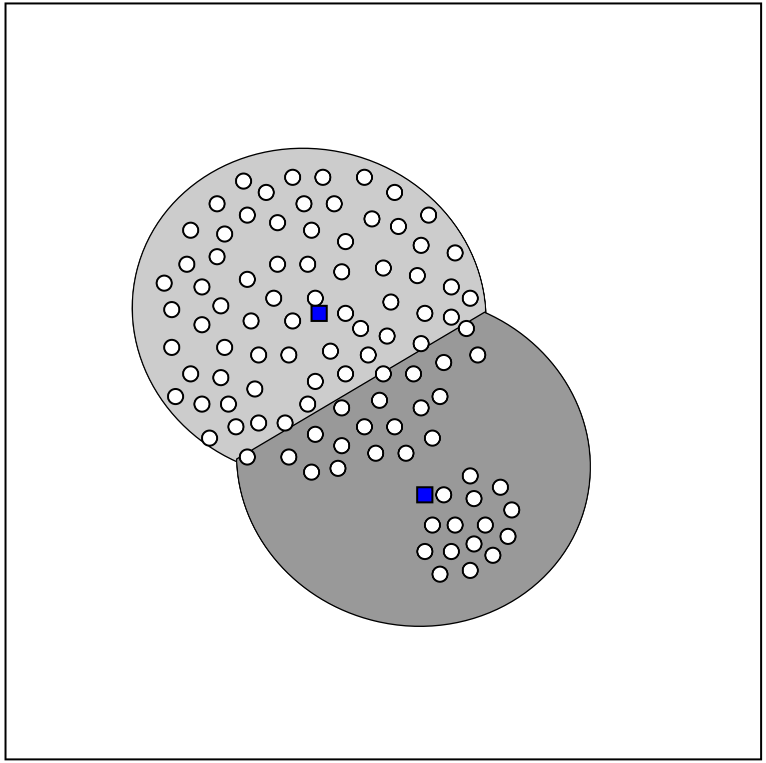
Flat Clustering

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.

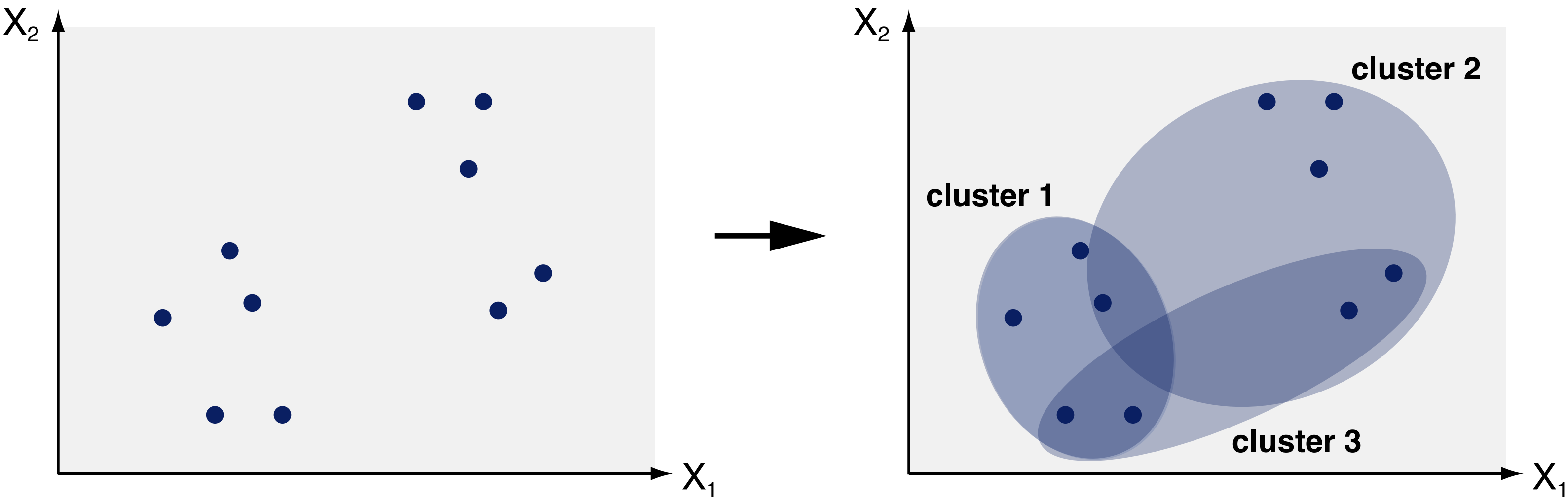


Soft Clustering

Soft Clustering

Soft clustering

- A flat clustering technique that maps instances to overlapping clusters
- Input. A set of instances $X = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}$ without class labels
- Output. A set of clusters $C = \{c_1, \ldots, c_k\}$ and a mapping $w_j : X \to [0, 1]$ 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 sentences:

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

- A soft clustering algorithm might identify two soft clusters:

Topic A representing food

Topic B representing pets

- It also assigns each sentence a weight for each cluster.

Selected algorithms used for soft clustering

- Fuzzy k-means clustering
- Latent Dirichlet Allocation

Topic Modeling

Topic modeling

- An analysis that extracts topics from a text corpus based on patterns in the use of words
- A topic is modeled as a list of words that cooccur in a statistically meaningful way.

BIRDS NEST TREE
BRANCH LEAVES

Why topic modeling?

- Find low-dimensional representations of high-dimensional text.
- Infer some kind of meaning from a vocabulary.
- Summarize texts concisely and capture their similarity.

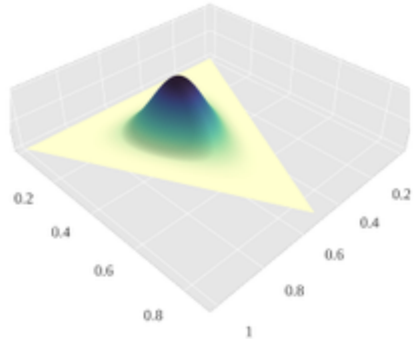
How to do topic modeling?

- Different techniques have been proposed for topic modeling.
- The most popular one is *Latent Dirichlet Allocation (LDA)*.
- The terms *topic modeling* and *LDA* are often used synonymously.

Topic Modeling

Latent Dirichlet Allocation (LDA)

LDA



- A probabilistic technique to automatically discover topics in a corpus

  In principle, LDA can also be used for data other than text.
- Learns the relative importance of topics in texts and of words in topics
- Based on the "bag-of-words" idea

LDA in a nutshell

- Model a text as a composition of words from word lists called *topics*.
- Decompose a text into the topics from which the words probably came.
- Repeat decomposition multiple times to obtain the most likely distribution of words over topics.

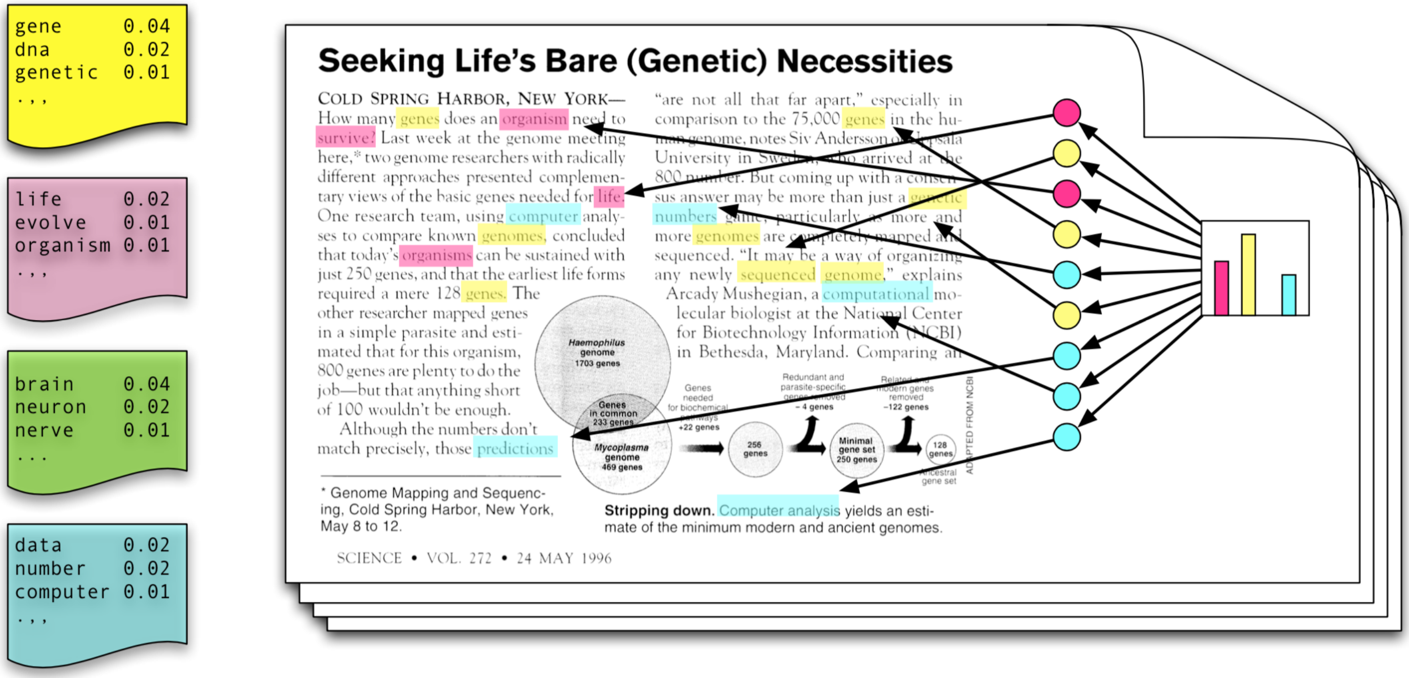
Notice

- Technically, LDA is often implemented using *Gibbs sampling*.

  The mathematical details are beyond the scope of this course.

Topic Modeling

Assumptions behind LDA

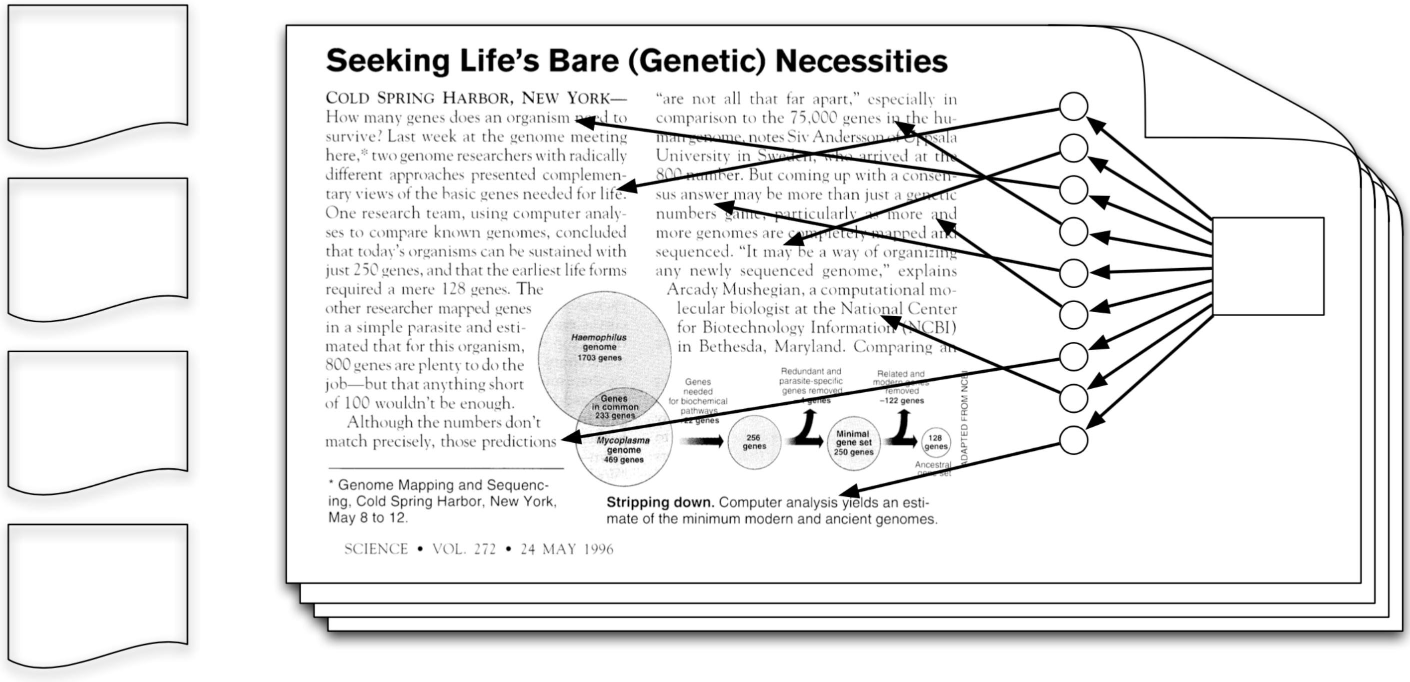


Assumptions

- Each text is a weighted combination of corpus-wide topics.
- Each topic is a distribution over words.
- Each word in a text is drawn from one of those topics.

Topic Modeling

Setting of LDA



Setting

- Given a text, we observe only words, not topics.
- The aim of LDA is to infer the latent (say, hidden) topic structure.

Topic Modeling

LDA Pseudocode Sketch

Signature

- Input. A set of n texts, a number of topics k to be found, and a number m of words to represent each topic with
- Output. Topic weighting of each text, word list for each topic

Pseudocode sketch

```
1.  repeat
2.      Randomly assign each word x in each text d to one topic t
3.      for each text d, word x in d, topic t do
4.          Reassign x to topic t with probability p(t|d) · p(x|t)
            // p(t|d): fraction of words in d currently assigned to t
            // p(x|t): overall fraction of assignments to t from x
5.  until probabilities stable (or until some max iterations)
6.  for each text d do Get topic weighting (w_1, ..., w_k)_d
                       // w_i: Fraction of words in d from topic i
7.  for each topic t do Get words (x_1, ..., x_m)_t
                       // x_i: The word i-th most often assigned to t
8.  return all (w_1, ..., w_k)_d and (x_1, ..., x_m)_t
```

Topic Modeling

Example Topic Models

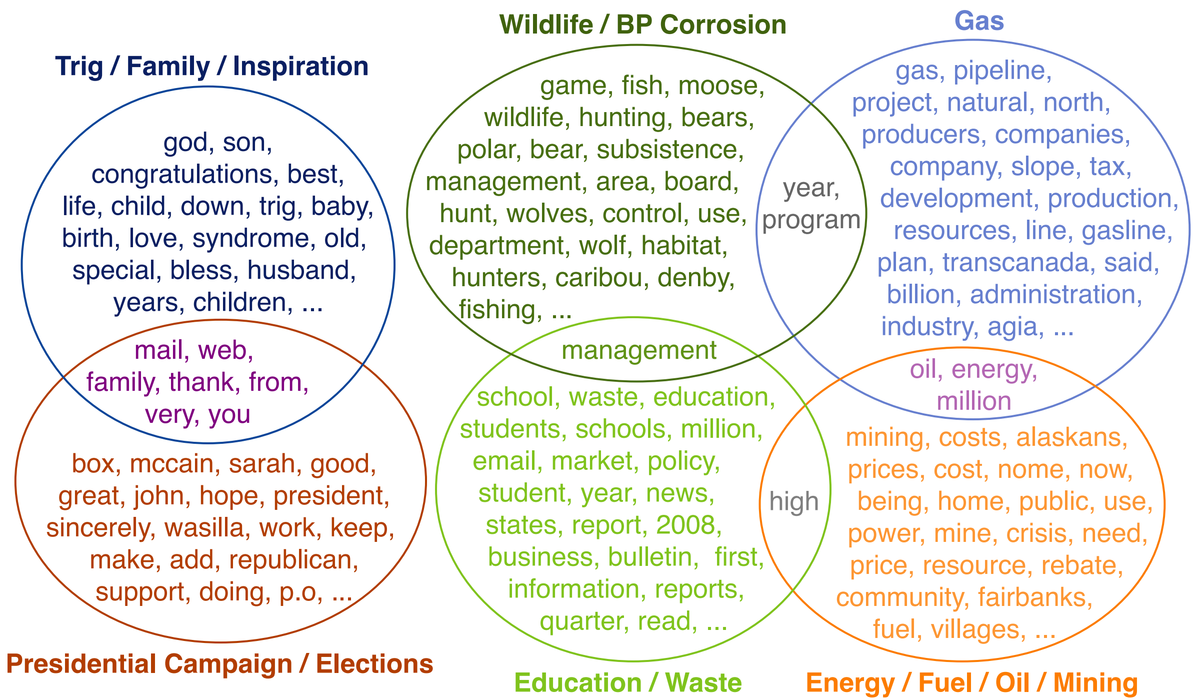
Case study

Taken from `http://blog.echen.me/2011/06/27/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

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

90% Wildlife / BP Corrosion, 10% Presidential Campaign / Election

We understand that you have been discussed as a possible choice for the **Vice Presidency**.

As **people** who **support** the democratic process and care about protecting our **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.

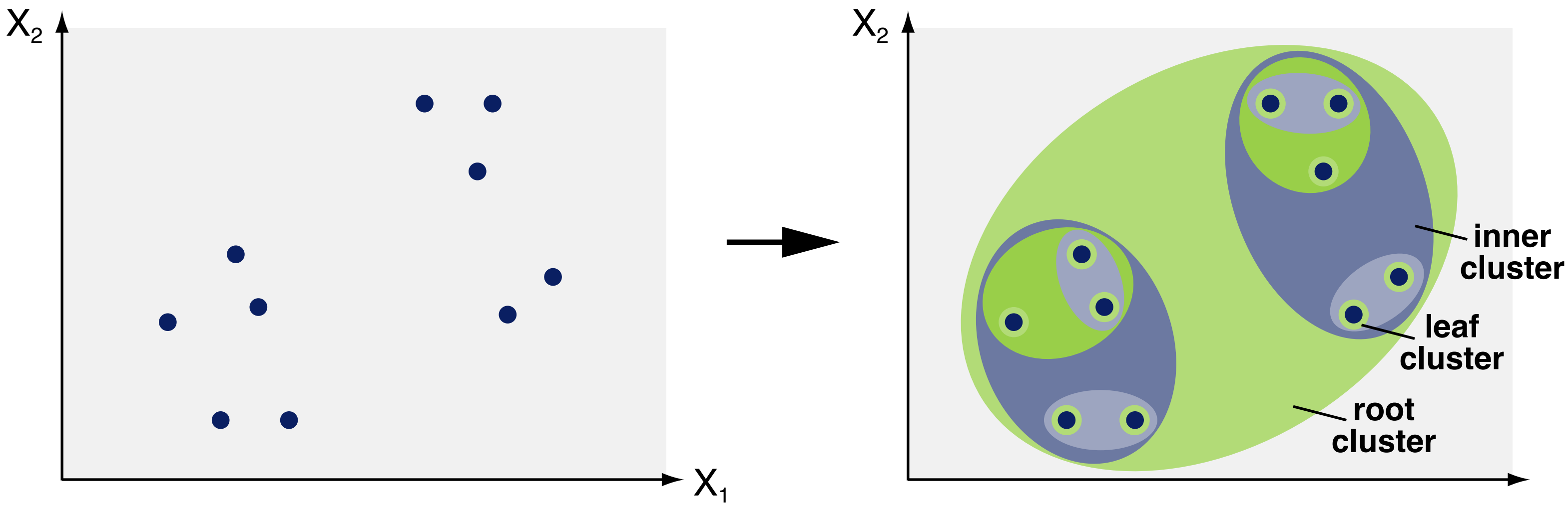
It is troubling that **you** are **now** working to deny more than 50,000 Alaskans a vote on **aerial** killing of **wolves** and **bears** with legislation now **being** considered in the Alaska legislature.

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 $X = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}$ 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

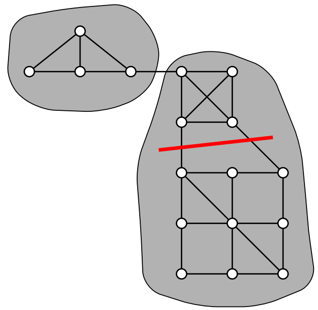


Notice

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

Hierarchical Clustering

Two Main Techniques

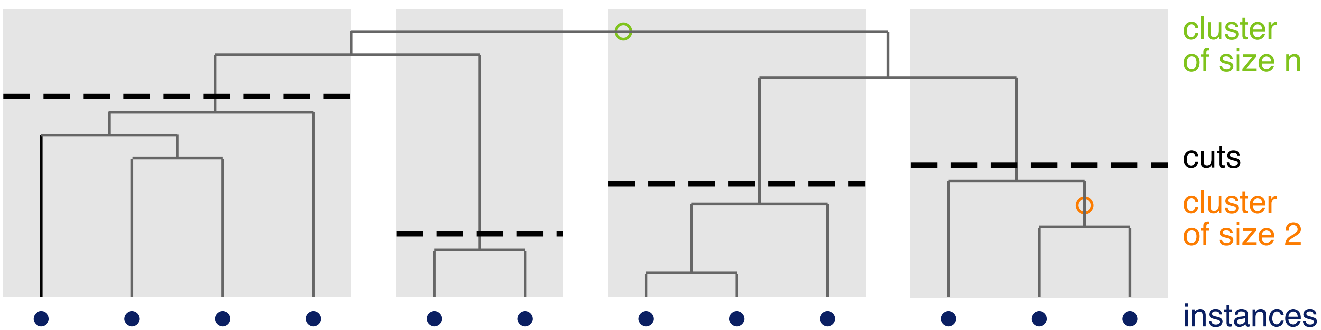


Divisive hierarchical clustering

- 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 closed 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

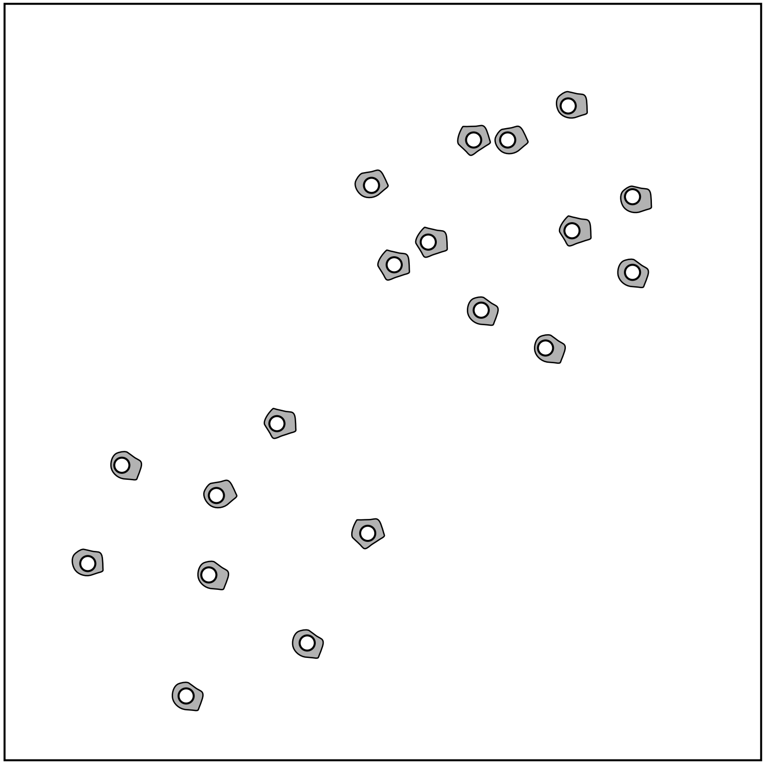
agglomerativeHierarchicalClustering(Set<Instance> X)

```
Set<Set<Instance>> clusters ← {{x^(i)} | x^(i) ∈ X} // cur. clusters
Set<Set<Instance>> V ← clusters // tree nodes
Set<Set<Instance>[]> E ← ∅ // tree edges
while |clusters| > 1 do
    double [][] similarities ← updateSimilarities(clusters)
    Set<Instance> [] pair ← getClosest(clusters, similarities)
    Set<Instance> merged ← pair[0] ∪ pair[1]
    clusters ← (clusters \ {pair[0], pair[1]}) ∪ {merged}
    V ← V ∪ {merged}
    E ← E ∪ {(merged, pair[0]), (merged, pair[1])}
return ⟨V, E⟩
```

Agglomerative Hierarchical Clustering

Example

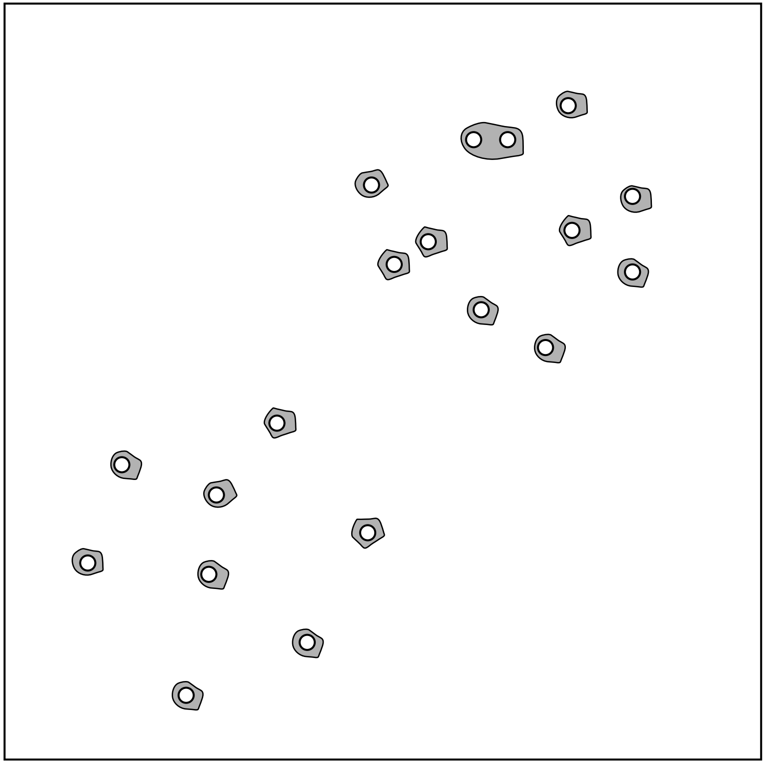
Line 1: Assign each instance to individual cluster

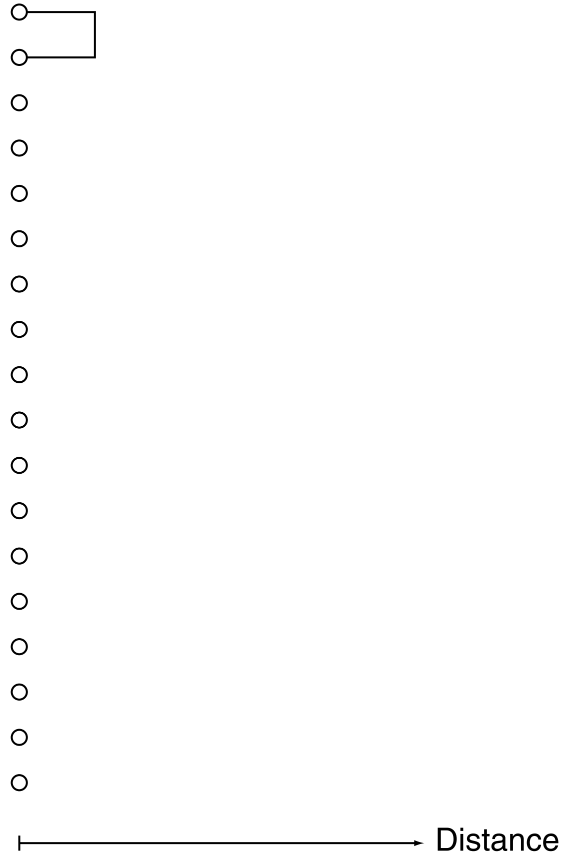


Agglomerative Hierarchical Clustering

Example

Lines 5–10: Combine closest pair of clusters into one cluster

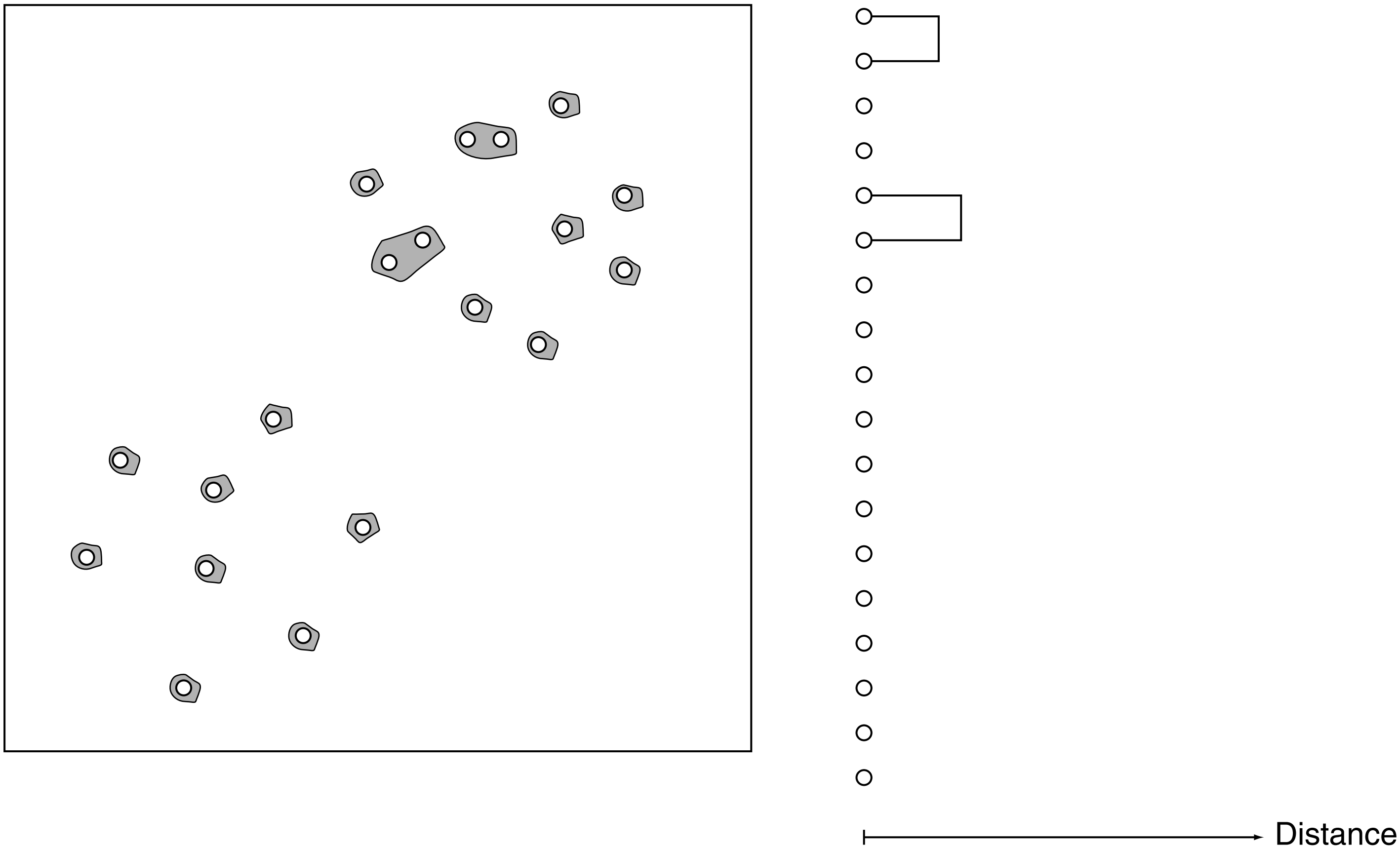




Agglomerative Hierarchical Clustering

Example

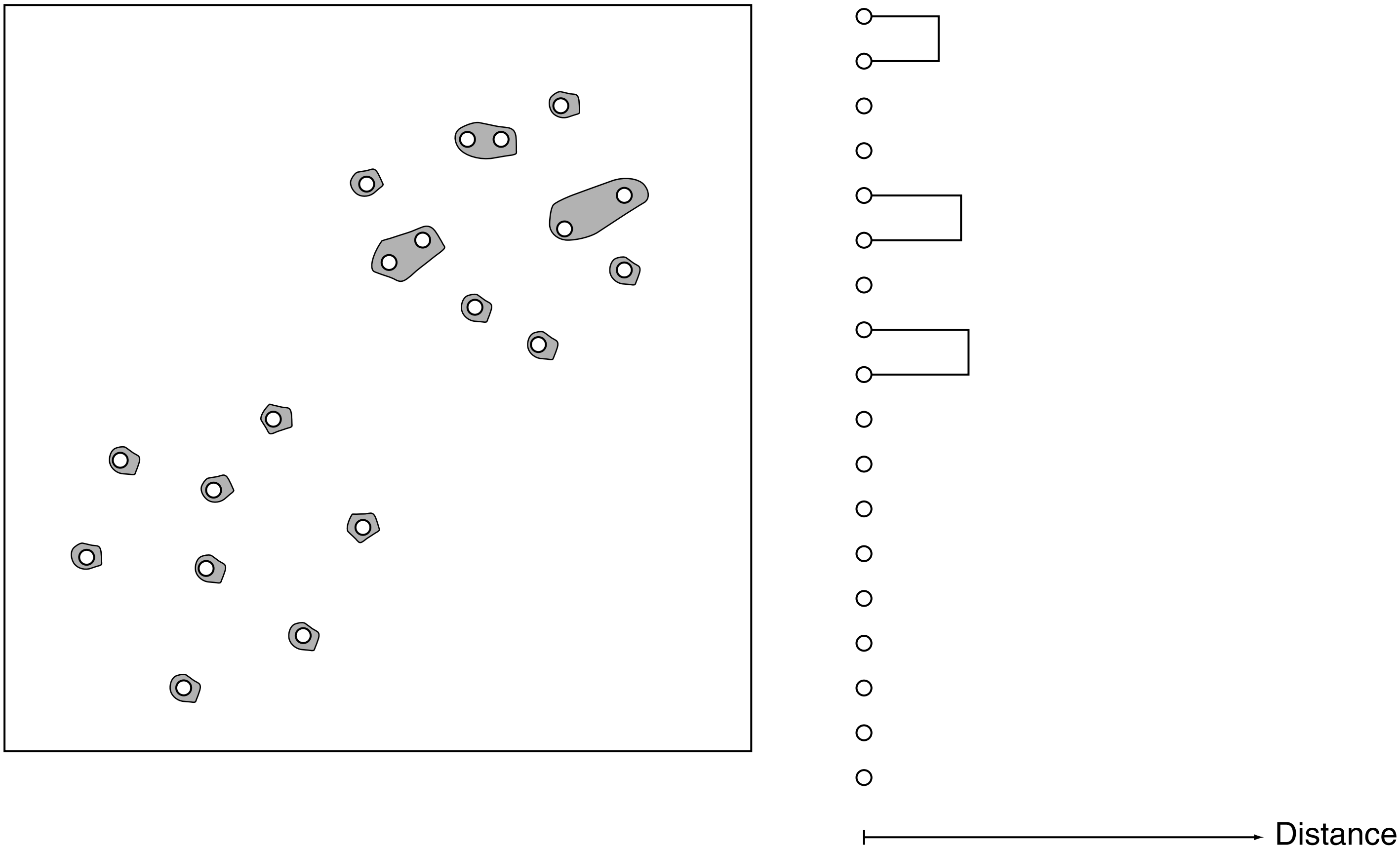
Lines 5–10: Repeat until only one cluster remains



Agglomerative Hierarchical Clustering

Example

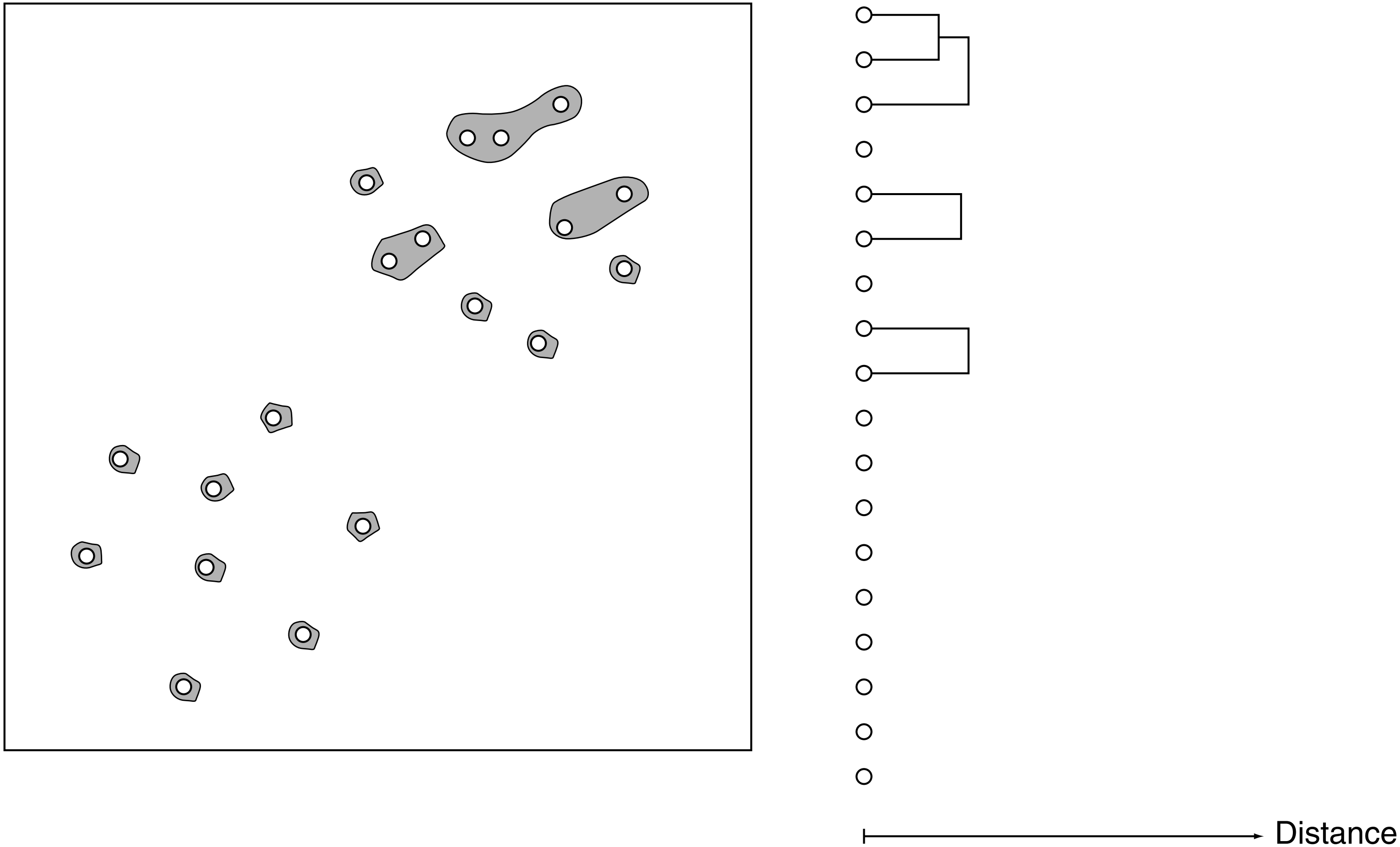
Lines 5–10: Repeat until only one cluster remains



Agglomerative Hierarchical Clustering

Example

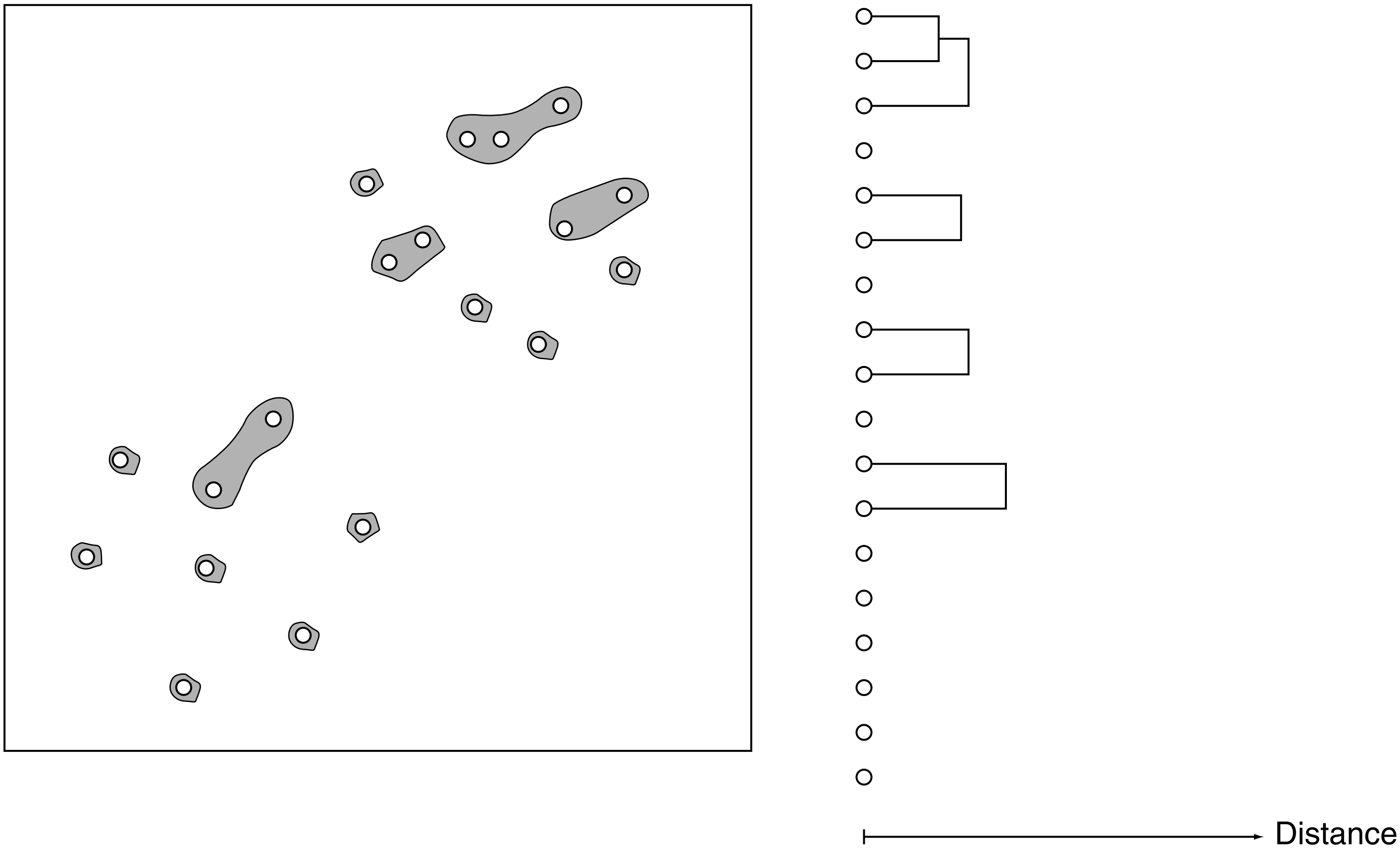
Lines 5–10: Repeat until only one cluster remains



Agglomerative Hierarchical Clustering

Example

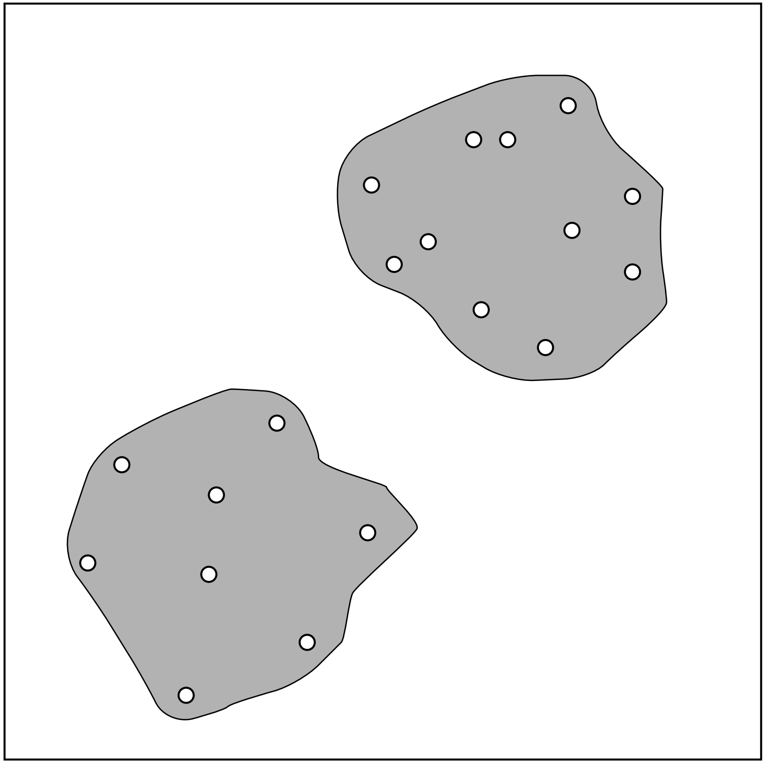
Lines 5–10: Repeat until only one cluster remains

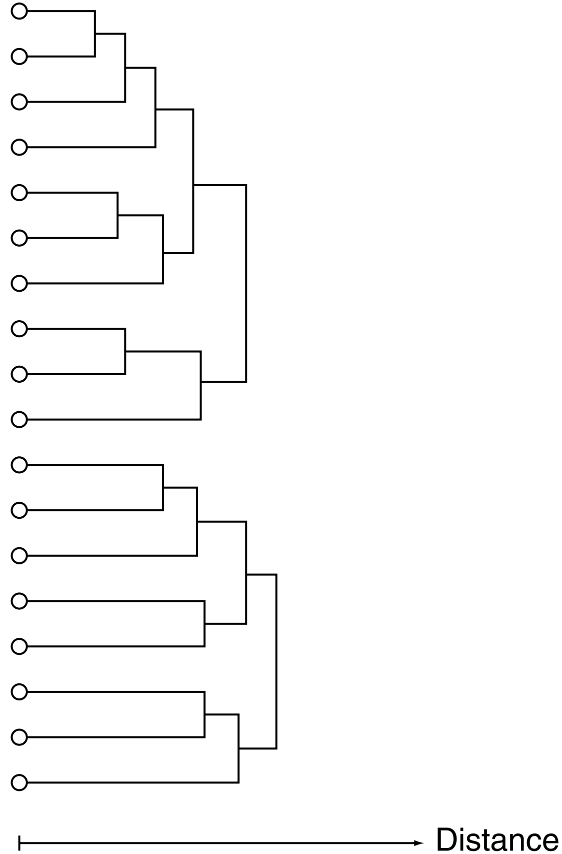


Agglomerative Hierarchical Clustering

Example

Lines 5–10: Repeat until only one cluster remains

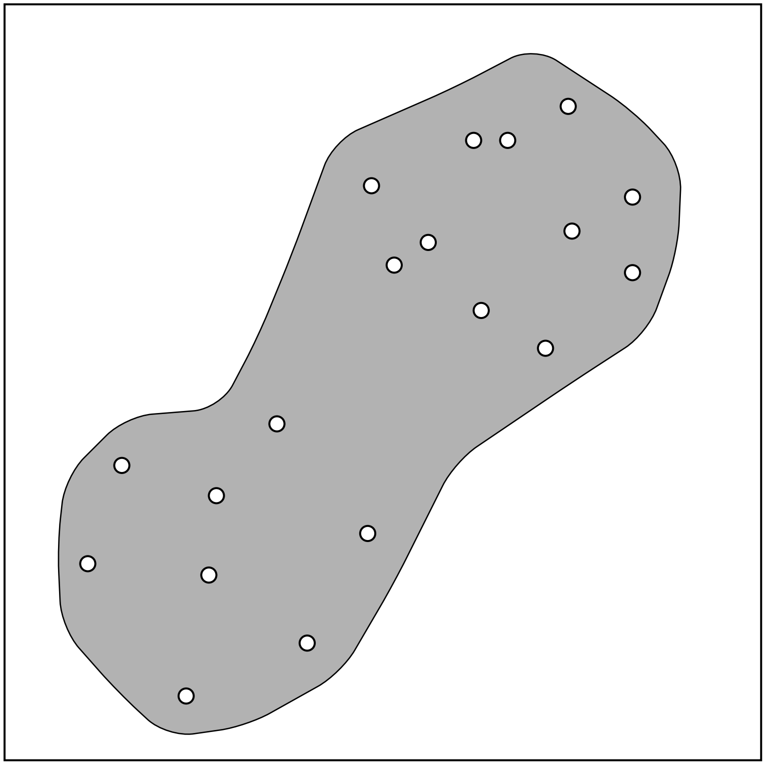


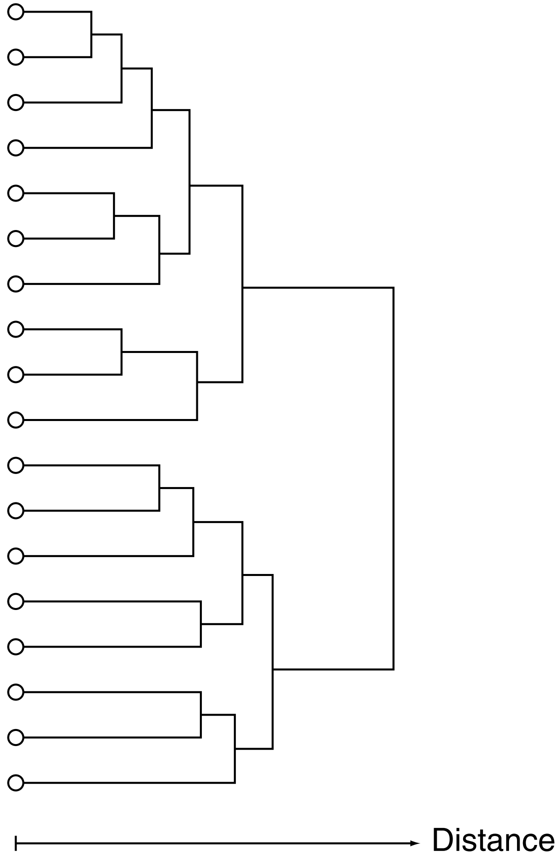


Agglomerative Hierarchical Clustering

Example

The dendrogram shows the final hierarchical clustering





Agglomerative Hierarchical Clustering

Cluster Similarity

Two components of cluster similarity

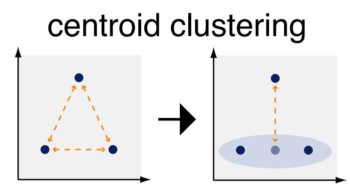
- Measure. Captures similarity of instances (or cluster representatives)
  Measures as in the previous lecture part: Cosine, Euclidean, ...
- Aggregation. Captures cluster similarity based on instance similarity

How to aggregate similarity?

- Several 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*.



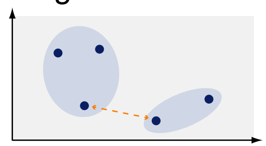
Agglomerative Hierarchical Clustering

Cluster Similarity Aggregation Methods

Single link clustering

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

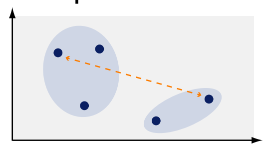
$$sim(c, c') = \max_{\mathbf{x} \in c, \mathbf{x}' \in c'} sim(\mathbf{x}, \mathbf{x}')$$



Complete link clustering

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

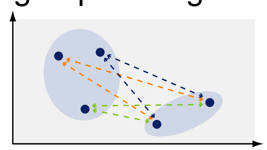
$$sim(c, c') = \min_{\mathbf{x} \in c, \mathbf{x}' \in c'} sim(\mathbf{x}, \mathbf{x}')$$



Group-average link clustering

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

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

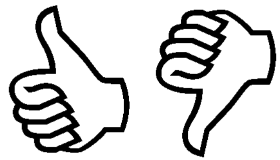


Review Sentiment Analysis

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., $\{1, \ldots, 5\}$ or $[0, 1]$

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.

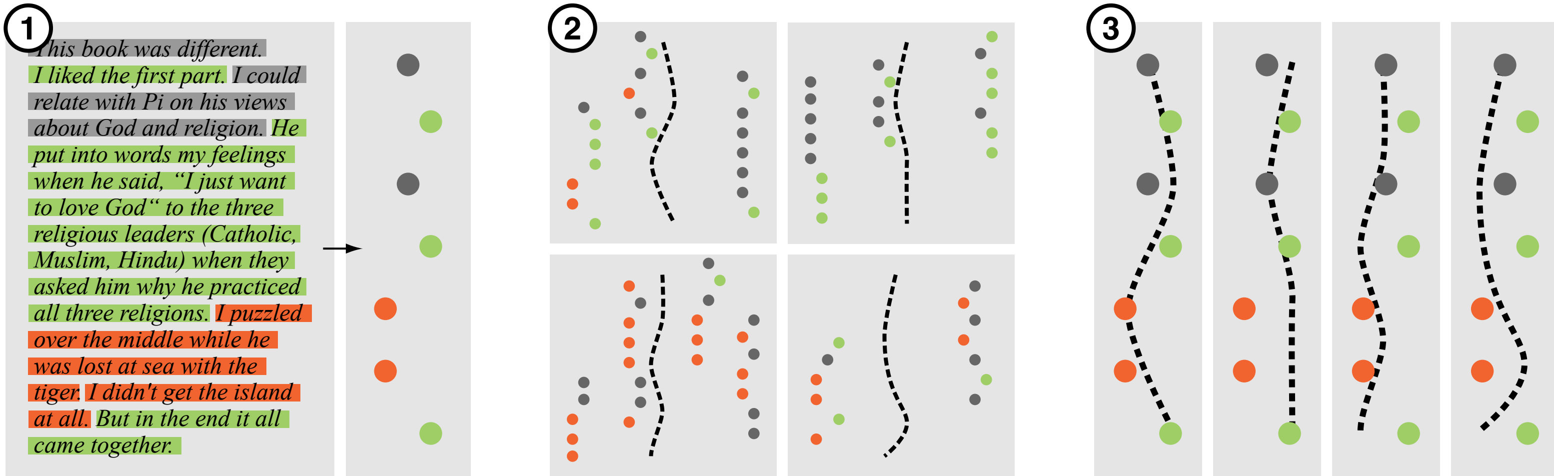
This book was different. I liked the first part. I could relate with Pi on his views about God and religion. He put into words my feelings when he said, "I just want to love God" to the three religious leaders (Catholic, Muslim, Hindu) when they asked him why he practiced all three religions. I puzzled 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.

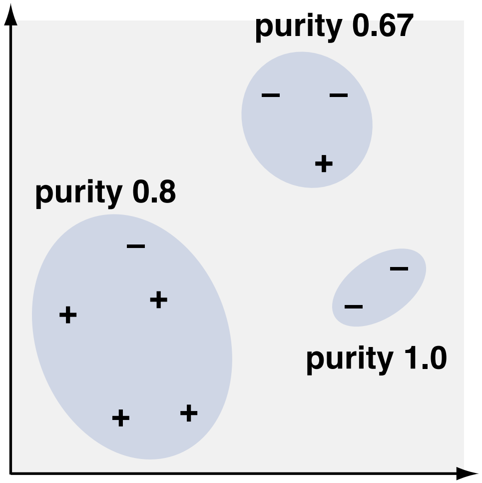


Hypotheses (both evaluated in later lecture parts)

- 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, the commonness of the patterns.

Obtaining flow patterns

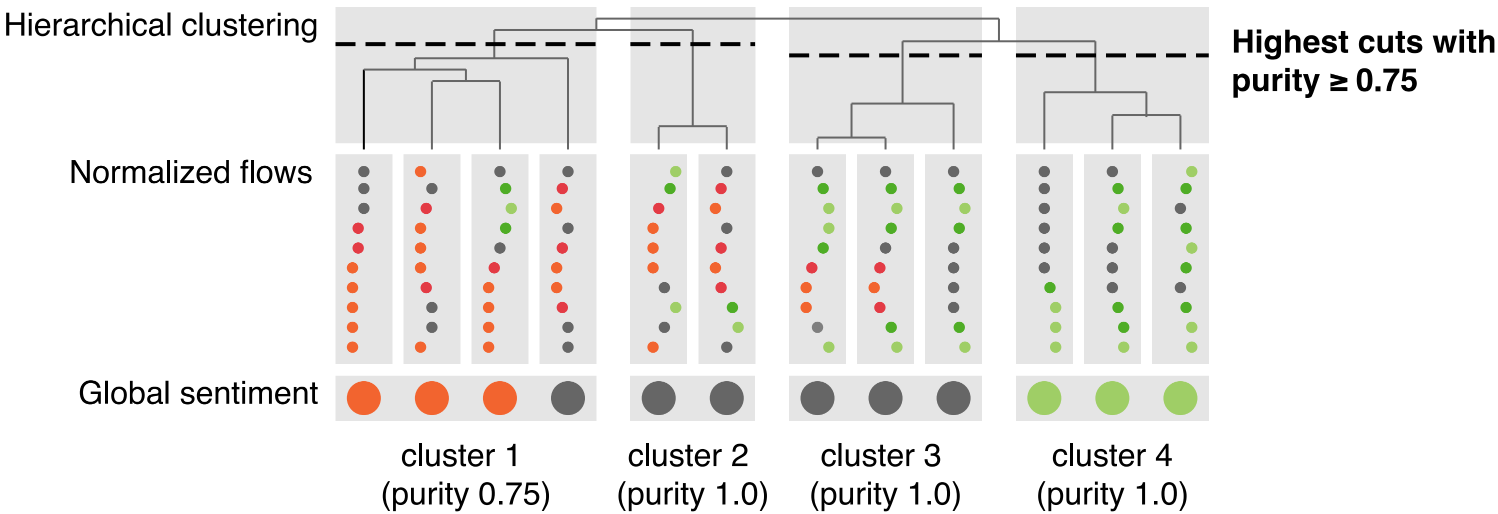
- The centroid of each cluster adequately serves as a flow pattern.

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

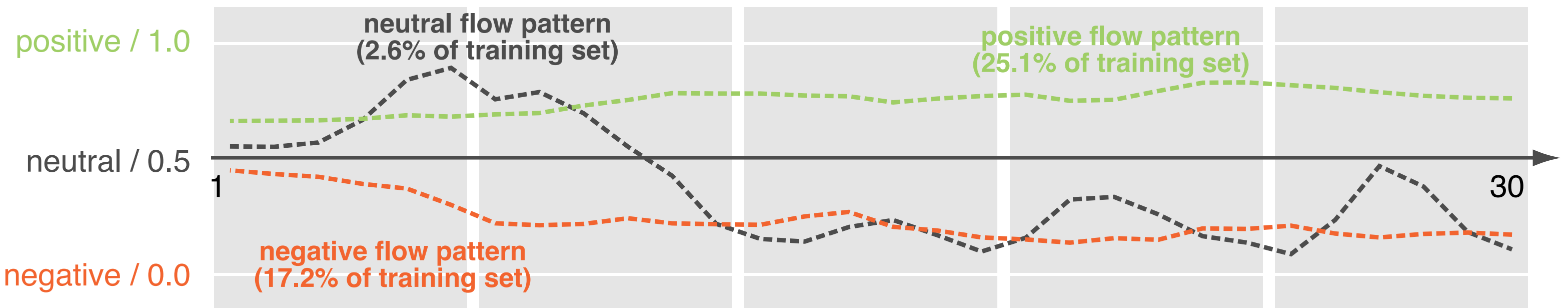
Review Sentiment Analysis

Example: Sentiment Flow Patterns using Clustering

Flow clustering for τ = 0.75



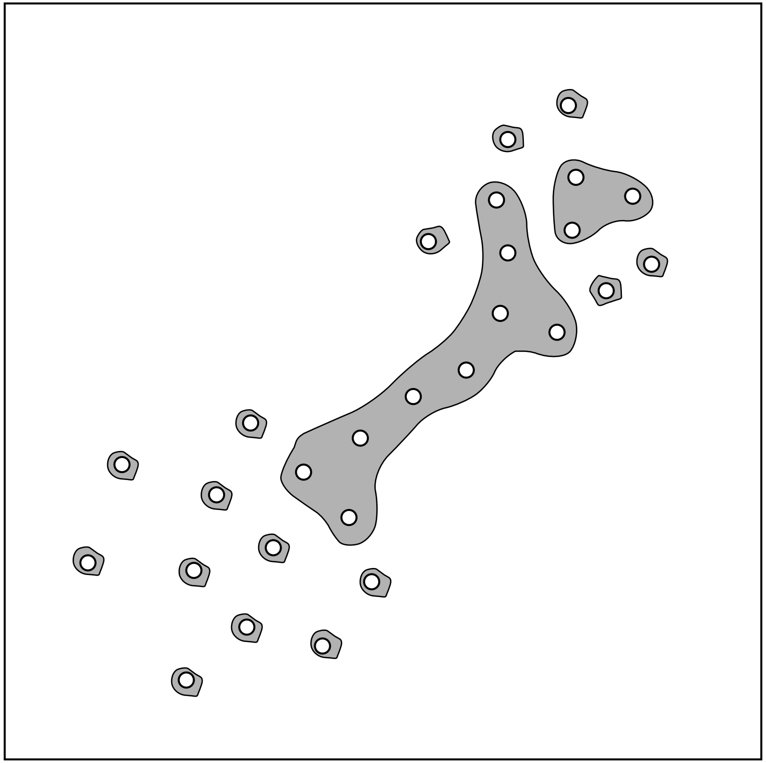
Most common flow patterns in 900 TripAdvisor reviews



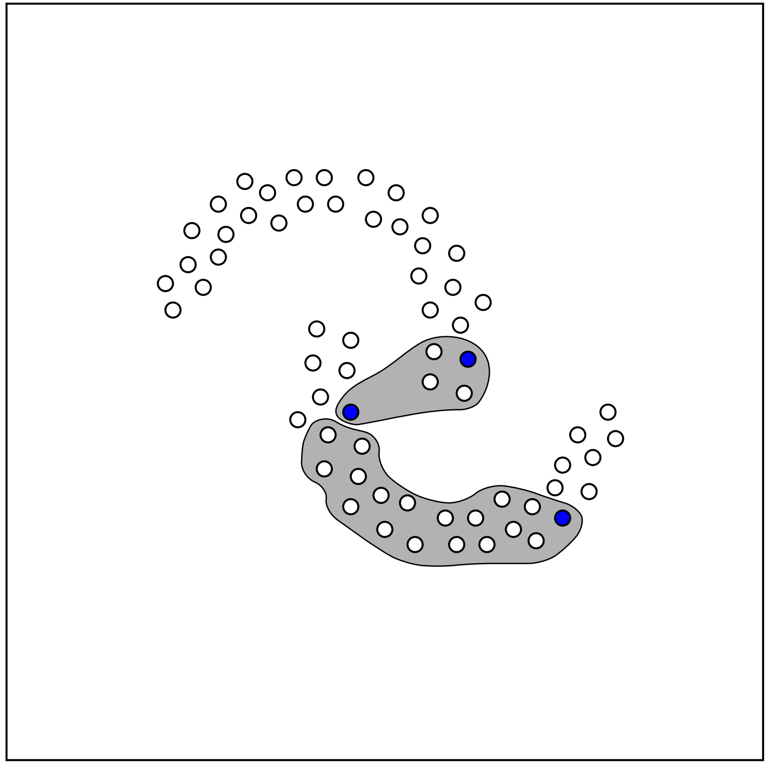
Hierarchical Clustering

Issues with Hierarchical Clustering Algorithms

Chaining problem of clustering using single-link similarity



Nesting problem of clustering using complete-link similarity

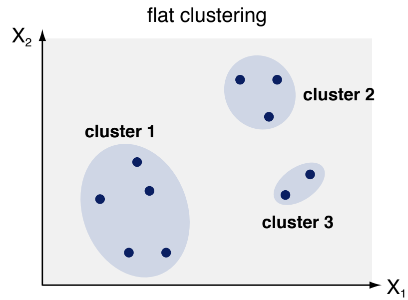


Conclusion

Conclusion

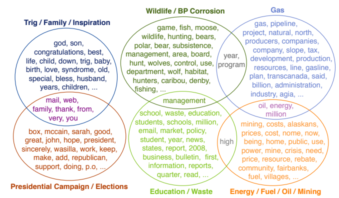
NLP using clustering

- Mostly unsupervised learning of text properties
- Targets situations where no ground truth is available
- Clustering is always based on similarity measures



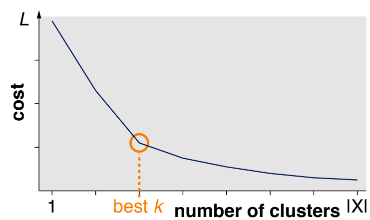
Clustering techniques

- Hard 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. http://www.cs.columbia.edu/~blei/talks/Blei_ICML_2012.pdf
- **Mansoorizadeh et al. (2016).** Muharram Mansoorizadeh, Mohammad Aminian, Taher Rahgooy, and Mehdy Eskandari (2016). Multi Feature Space Combination for Authorship Clustering. In Working Notes of the CLEF 2016 Evaluation Labs, pages 932–938.
- **Sari and Stevenson (2016).** Yunita Sari and Mark Stevenson (2016). Exploring Word Embeddings and Character N-Grams for Author Clustering. In Working Notes of the CLEF 2016 Evaluation Labs, pages 989–991.
- **Stein and Lettmann (2010).** Benno Stein and Theodor Lettmann. Data Mining. Lecture Slides, 2010. https://webis.de/lecturenotes/slides.html
- **Wachsmuth (2015).** Henning Wachsmuth (2015): Text Analysis Pipelines — Towards Ad-hoc Large-scale Text Mining. LNCS 9383, Springer.
- **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.