Introduction to Text Mining

Part VII: Text Mining using Similarities and Clustering

Henning Wachsmuth

https://cs.upb.de/css

Similarities and Clustering: Learning Objectives

Concepts

- How to employ clustering within text mining
- The role of similarity measures in clustering
- The pros and cons of different clustering types

Text analysis techniques

- Computation of the similarity of text spans in various ways
- Partitioning of a set of texts into groups with flat clustering
- Creation of soft clusters using topic modeling
- Ordering of texts by similarity with hierarchical clustering

Covered text analyses

- Authorship attribution
- Topic detection
- Discourse pattern recognition

Outline of the Course

I. Overview

II. Basics of Linguistics

III. Text Mining using Rules

IV. Basics of Empirical Methods

V. Text Mining using Grammars

VI. Basics of Machine Learning

VII. Text Mining using Similarities and Clustering

- What Is Text Mining using Clustering?
- Similarity Measures
- Hard and Soft Flat Clustering
- Hierarchical Clustering

VIII. Text Mining using Classification and Regression

IX. Text Mining using Sequence Labeling

X. Practical Issues

What Is Text Mining using Clustering?

Clustering

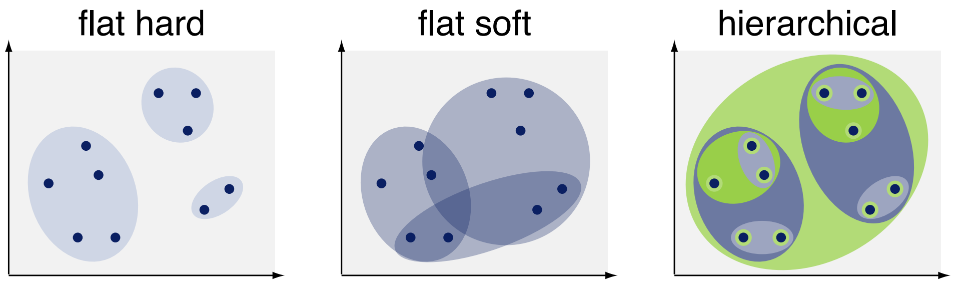
What is clustering (aka cluster analysis)?

- The grouping of a set of instances into some number $k \geq 1$ of classes.

  k is possibly, but not necessarily predefined.
- Each resulting group is called a *cluster*.
- The meaning of the clusters is usually unknown beforehand.

Types of clusterings

- Flat vs. hierarchical
- Hard vs. soft



flat hard

flat soft

hierarchical

Clustering vs. cluster labeling

- Clustering does not assign labels to the created clusters.
- *Cluster labeling* is all but trivial; it requires to infer the hidden concept connecting the instances in a group.

  The labeling is beyond the scope of this course.

Clustering

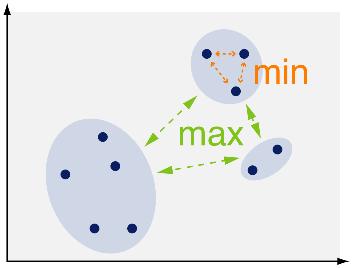
Clustering using Unsupervised Learning

Clustering as unsupervised learning

- Clustering models y are usually learned unsupervised, i.e., based on a set of instances $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}$ only (without class information).

  We will see a supervised variant of clustering below, though.
- Clustering is the most common unsupervised learning technique.

Objective of unsupervised clustering



- Minimize the distance within all clusters.
- Maximize the distance between the clusters.

  Analog: Maximize *similarity* within, mininimize across.

What does clustering do?

- Patterns in the instances are found using *similarity measures*.
- The resulting instance clusters correspond to classes.
- The resulting model can assign arbitrary instances to the clusters.

Clustering

Similarity Measures

Similarity measure

- A measure that quantifies how similar two instances of a concept are.

  Different types of similarity measures exist (details below).

Similarity measures in clustering

- Clustering algorithms compute similarities to decide what instances to merge in a cluster.
- To merge clusters, similarities are also computed between clusters.

  Different ways to define cluster similarity exist (details below).

Similarity vs. relatedness

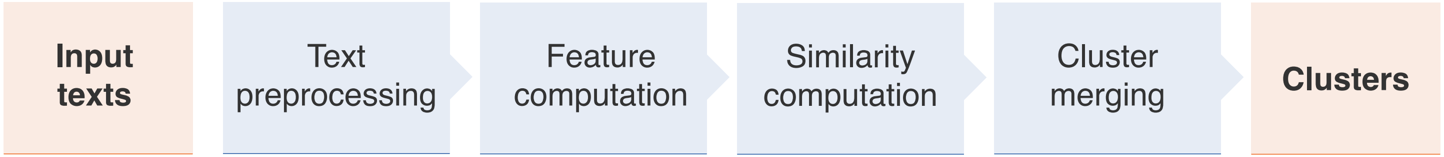
- Similar. Concepts with similar meaning, e.g., “car” and “bike”.
- Related. Dissimilar concepts may still be related, e.g., “car” and “gas”.

  Sometimes, related concepts are accepted as being similar, though.

Text Mining using Clustering

Clustering in text mining

- Input. Usually plain texts or text spans.
- Output. A set of clusters, and a model that maps from texts to clusters.



(similarity computation and cluster merging are mostly done iteratively)

Why clustering in text mining?

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

  The inference of class *labels* is done manually in many cases, though (see above).

Selected applications in text mining

- Topic detection. What are the topics covered by a corpus of texts?
- Text retrieval. Detection of texts with similar properties.

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

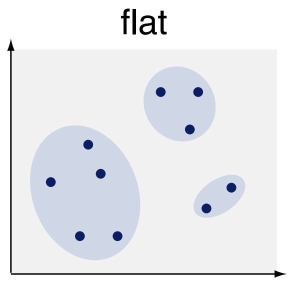
Text Mining using Clustering

Flat vs. Hierarchical Clustering

Flat and hierarchical clustering

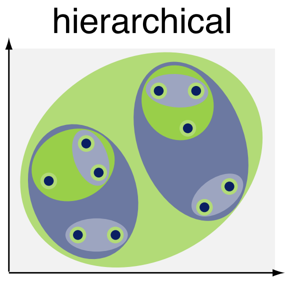
- Flat. Group a set of instances into a set of clusters.

  {1, 2, 3, 4} → {1, 3, 4}, {2}



- Hierarchical. Create a binary tree over all instances where each node represents a cluster of a certain size.

  {1, 2, 3, 4} → { { {{1}, {3}}, {4} }, {6} }



What type to use in text mining?

- In many settings, the final goal is to obtain a flat clustering.
- Flat clusterings can also be obtained through cuts in a hierarchy tree.
- What type to choose is rather an implementation decision, related to the effectiveness and efficiency of clustering.
- This is different if the hierarchical information is really required.

  For instance, when a taxonomy of related concepts shall be created.

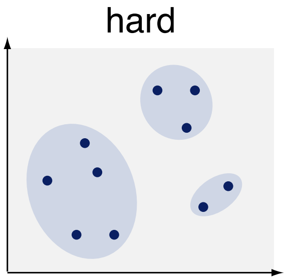
Text Mining using Clustering

Hard vs. Soft Clustering

Hard and soft clustering

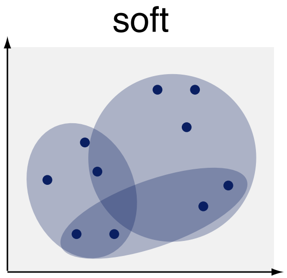
- Hard. The clustering creates a partition, such that each instance $\mathbf{x}^{(i)}$ belongs to a single cluster C_j.

  {1, 2, 3, 4} → C_1 = {1, 3, 4}, C_2 = {2}



- Soft. The clustering creates overlapping clusters, such that each instance $\mathbf{x}^{(i)}$ belongs to each cluster C_j with some weight $w_j^{(i)} \in [0, 1]$, $\sum_j w_j^{(i)} = 1$.

  {1, 2, 3, 4} → C_1 = (1, 0.6, 0.8, 0), C_2 = (0, 0.4, 0.2, 1)



What type to use in text mining?

- Hard clustering is used to identify a set of classes.
- Soft clustering can be understood as defining weighted concepts based on the classes, which is preferred where overlap is assumed.

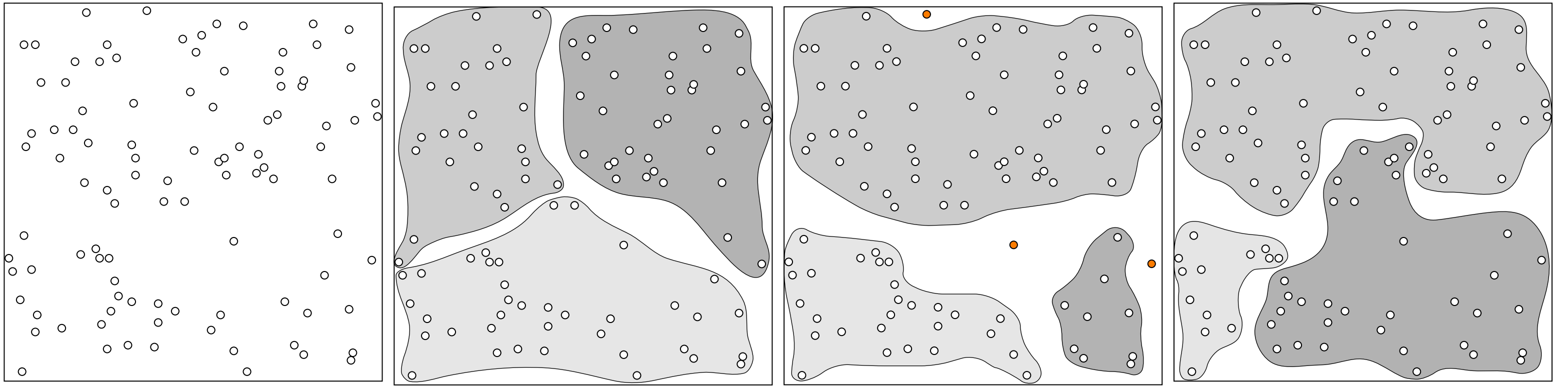
  An example is *topic modeling*, which finds overlapping topics (see below).

Unsupervised Learning

Evaluation of Clustering

The clustering evaluation problem

> *“The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.”*
>
> (Jain and Dubes, 1990)



Random points k-means DBSCAN Complete link

Unsupervised Learning

Evaluation of Clustering: Goals and Measures

Possible goals of clustering evaluation

- Rank alternative clusterings by their quality.
- Determine the ideal number of clusters.
- Relate found clusters to externally provided class information.
- Find out what clustering algorithm is suitable.
- Find out whether data contains non-random structures.

Evaluation measures

- Internal. Analyze intrinsic characteristics of a clustering.

  Example: Average cluster distance.
- External. Analyze how close a clustering is to an (external) reference, i.e., to a test set with ground-truth information.

  Example: The (average) purity of all clusters.
- Relative. Analyze the sensitivity of internal measures during clustering.

  Not in the focus of this course.

Similarity Measures

Similarity Measures

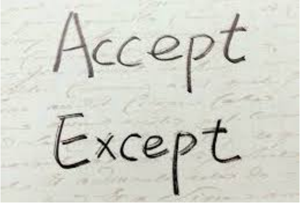
What is a similarity measure?

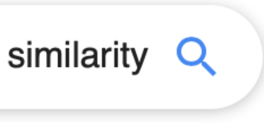
- A real-valued function sim that quantifies how similar two instances o_1, o_2 of the same concept O are.
- Usually, values of sim range between 0 (no similarity) and 1 (identity).
- In text mining, instances are (the representations of) input text spans.

Various use cases in text mining

- Clustering
- Spelling correction
- Retrieval of relevant web pages
- Detection of related documents
- Paraphrase recognition
- (Near-) Duplicate or plagiarism detection

... and many more





Similarity Measures

Text Similarity

Similarity in text mining

- Similarity between the *form* of two texts or text spans.
- Similarity between the *meaning* of two texts or text spans.

Similar form, different meaning: “This is shit.” vs. “This is *the* shit.”

Other way round: “Obama visited the capital of France.” vs. “Barack Obama was in Paris.”

- Ultimately, similarity measures usually aim to capture meaning.
- But form is often used as a proxy.

Text similarity measures

- Vector-based. Mainly, for similarity between feature vectors.
- Difference-based. For spelling similarities.
- Thesaurus-based. For synonymy-related similarities.
- Distributional. For similarities in the contextual usage.

Clustering is mostly based on the first, but the others may still be used internally.

Vector-based Similarity Measures

Vector-based similarity measure

- A function that quantifies the similarity of instances o_1, o_2 based on their vector representations $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$, usually encoding features.
- The values at $x_i^{(1)}$ and $x_i^{(2)}$ at each position i are compared individually.

Feature-based representation (recap)

- A feature vector is an ordered set of values of the form $\mathbf{x} = (x_1, \ldots, x_m)$.
- Each feature x_j denotes a measurable property of an input, $1 \leq j \leq m$.

  We consider only real-valued features here.
- Each instance o_j is mapped to a vector $\mathbf{x}^{(i)} = (x_1^{(u)}, \ldots, x_m^{(i)})$ where $x_j^{(i)}$ denotes the value of feature x_j.

  We consider only values normalized to the range $[0, 1]$ here.

Similarity measures and clustering

- Most clustering algorithms rely on vector-based similarity measures.

Vector-based Similarity Measures

Concept

Measuring similarity between vectors

- Compare two vectors of the same representation with each other.

$$\mathbf{x}^{(1)} = (1.0, 0.1, 0.3),\ \mathbf{x}^{(2)} = (0.0, 0.1, 0.6) \quad \text{for } \mathbf{x} = (\text{red, green, blue})$$

- Compute similarity individually at each position of the vectors.

$$sim_1(1.0, 0, 0) = 0.0 \quad sim_2(0.1, 0, 1) = 1.0 \quad sim_3(0.3, 0.6) = 0.5$$

- Aggregate all individual similarities in some way.

$$sim(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \frac{0.0+1.0+0.5}{3} \approx 0.5$$

Measuring similarity between all vector pairs

Instance	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	$\ldots$	$\mathbf{x}^{(n)}$
$\mathbf{x}^{(1)}$	1	$sim(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$	$\ldots$	$sim(\mathbf{x}^{(1)}, \mathbf{x}^{(n)})$
$\mathbf{x}^{(2)}$	-	1	$\ldots$	$sim(\mathbf{x}^{(2)}, \mathbf{x}^{(n)})$
$\vdots$				
$\mathbf{x}^{(n)}$	-	-	$\ldots$	1

Vector-based Similarity Measures

Similarity and Distance

Similarity vs. distance

- Internally, clustering algorithms compute distances between instances.
- Similarity can be seen as the inverse of distance.
- With normalized values, deriving one from the other is straightforward.

Properties of a distance function (aka metric)

- Non-negativity. $d(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \ \geq \ 0$
- Identity. $d(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}) \ = \ 0$
- Symmetry. $d(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \ = \ d(\mathbf{x}^{(2)}, \mathbf{x}^{(1)})$
- Subadditivity. $d(\mathbf{x}^{(1)}, \mathbf{x}^{(3)}) \ \leq \ d(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) + d(\mathbf{x}^{(2)}, \mathbf{x}^{(3)})$

  Clustering does not necessarily require subadditivity.

Concrete measures

- Numerous vector-based similarity/distance measures exist. (Cha, 2007)
- Four main measures. *Cosine*, *Jaccard*, *Euclidean*, *Manhattan*.

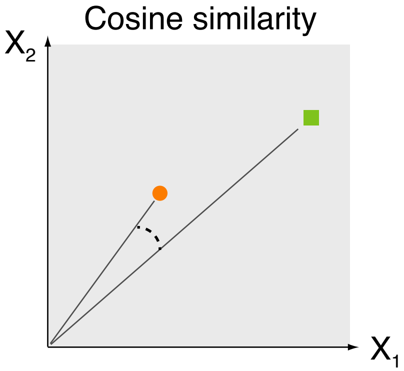
Vector-based Similarity Measures

Cosine Similarity

Cosine similarity (aka cosine score)

- Cosine similarity captures the cosine of the angle between two vectors.
- The smaller the angle, the more similar the vectors.

  Cosine is maximal (1.0) for 0°.



$$sim_{Cosine}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \frac{\mathbf{x}^{(1)} \cdot \mathbf{x}^{(2)}}{||\mathbf{x}^{(1)}|| \cdot ||\mathbf{x}^{(2)}||} = \frac{\sum_{j=1}^{m} x_j^{(1)} \cdot x_j^{(2)}}{\sqrt{\sum_{j=1}^{m} x_j^{(1)^2}} \cdot \sqrt{\sum_{j=1}^{m} x_j^{(2)^2}}}$$

Notice

- Cosine similarity abstracts from the length of the vectors.
- Angle computation works for any number of dimensions.
- Cosine similarity is probably the most widely-used similarity measure.

Vector-based Similarity Measures

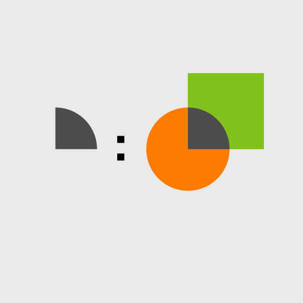
Jaccard Similarity

Jaccard similarity (aka Jaccard coefficient/index)

- Jaccard similarity captures how large the intersection of two sets is compared to their union.
- With respect to vector representations, this makes at least sense for boolean features.

  For others, if there may be a reasonable way of thresholding.

Jaccard similarity



$$sim_{Jaccard}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \frac{|\mathbf{x}^{(1)} \cap \mathbf{x}^{(2)}|}{|\mathbf{x}^{(1)} \cup \mathbf{x}^{(2)}|} = \frac{|\mathbf{x}^{(1)} \cap \mathbf{x}^{(2)}|}{|\mathbf{x}^{(1)}| + |\mathbf{x}^{(2)}| - |\mathbf{x}^{(1)} \cap \mathbf{x}^{(2)}|}$$

$$= \frac{\sum_{x_j^{(1)}=x_j^{(2)}} 1}{\sum_{x_j^{(1)}} 1 + \sum_{x_j^{(2)}} 1 - \sum_{x_j^{(1)}=x_j^{(2)}} 1}$$

Notice

- The Jaccard similarity does *not* consider the size of the difference between feature values.

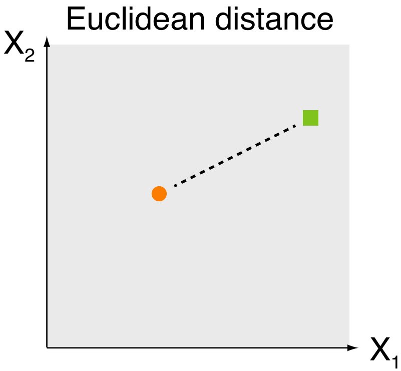
Vector-based Similarity Measures

Euclidean Similarity

Euclidean distance

- The Euclidean distance captures the straight-line distance between two feature vectors.

$$d_{Euclidean}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \;=\; \sqrt{\sum_{j=1}^{m} |x_j^{(1)} - x_j^{(2)}|^2}$$



Euclidean similarity

- If all feature values are normalized to $[0, 1]$, the Euclidean similarity is:

$$sim_{Euclidean}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \;=\; 1 - \frac{d_{Euclidean}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})}{\sqrt{m}}$$

Notice

- Euclidean spaces generalize to any number of dimensions $m \geq 1$. Here, this means to any number of features.

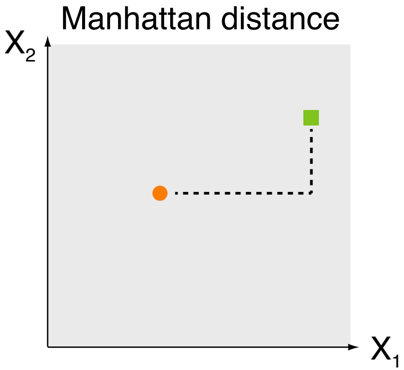
Vector-based Similarity Measures

Manhattan Similarity

Manhattan distance (aka city block distance)

- The Manhattan distance is the sum of all absolute differences between two feature vectors.

$$d_{Manhattan}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \;=\; \sum_{j=1}^{m} |x_j^{(1)} - x_j^{(2)}|$$



Manhattan similarity

- If all feature values are normalized to $[0, 1]$, the Manhattan similarity is:

$$sim_{Manhattan}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \;=\; 1 - \frac{d_{Manhattan}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})}{m}$$

Notice

- Manhattan distance and Euclidean distance are both special cases of the *Minkowski distance*.

$$d_{Minkowski}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \sqrt[p]{\sum_{j=1}^{m} |\mathbf{x}_j^{(1)} - \mathbf{x}_j^{(2)}|^p} \quad \text{for any } p \in \mathbb{N}^+$$

Vector-based Similarity Measures

When to Use What Measure?

Comparison of the measures

- Cosine. Focuses on those features that occur. Targets tasks where a vector's direction matters rather than its length.

  A typical task is matching queries with documents in web search.
- Jaccard. Less exact than cosine similarity, but therefore tends to be more robust (i.e., it "overfits" less).
- Euclidean and Manhattan. Target tasks where a value of 0 does not mean the absence of a feature.
- Euclidean vs. Manhattan. Depends on whether sensitivity to outliers in certain features is preferred or not.

Similarity as a hyperparameter

- There is not one best measure for all tasks.
- One way to deal with this is to evaluate different measures.
- In some tasks, measures can also be used simultaneously.

Similarity between Strings

Limitation of vector-based measures in text mining

- Similarity is restricted to corresponding feature values, $x_j^{(1)}, x_j^{(2)}$.
- Most features are derived directly from text spans.
- Similarity of different forms with similar meaning is missed...

"traveling" vs. "travelling" "woodchuck" vs. "groundhog" "Trump" vs. "The President"

... unless it is explicitly accounted for.

Similar strings

- Writing variation. Spelling errors, language differences, or extra words.
- Synonyms. Different words that refer to similar concepts.
- Relatedness. Different concepts that are related in a way that should be seen as similar in a given application.

... and similar

Similarity between Strings

Edit Distance

What is (minimum) edit distance?

```
I N T E * N T I O N
| | | | | | | | | |
d s s   i s
| | | | | | | | | |
* E X E C U T I O N
```

- The minimum number (or cost) of editing operations needed to transform one string to another.
- Editing operations. Insertion, deletion, substitution.
- Weighted edit distance. Different edits vary in costs.

How to compute edit distance?

	E	X	E
I	**s**(I,E)	**i**(*,X)	
N	**d**(N,*)	**s**(N,X)	
T			

- Sequence alignment using dynamic programming.
- Equals shortest path search in a weighted graph.

Selected applications

- Spelling correction, e.g., in search engines.

  “wreckonize speach” → Did you mean “recognize speech”?

- Near-duplicate identification, e.g., in plagiarism detectors.

Similarity between Strings

Thesaurus Methods

What are synonyms?

- Terms that have the same meaning in some or all contexts.

  “couch” vs. “sofa” “big” vs. “large” “water” vs. “H_20” “vomit” vs. “throw up”

- There are hardly any perfectly synonymous terms.

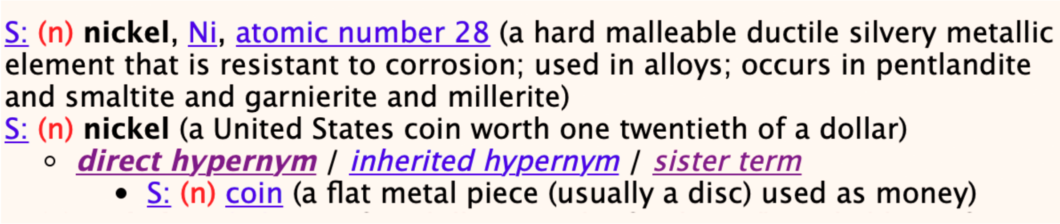
  Even seemingly identical terms usually differ in terms of politeness, slang, genre, etc.

- Synonymy is a relation between senses rather than words.

  “big” vs. “large” → “Max became kind of a <insert> brother to Linda.”

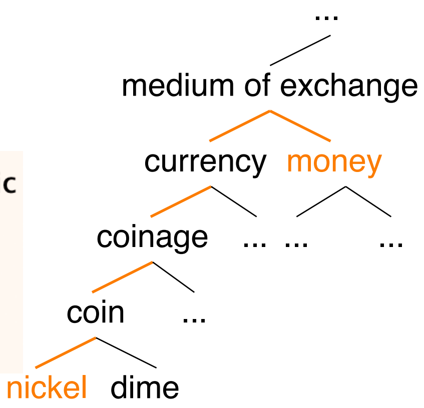
Similarity with thesaurus methods

- Compute distance in thesauri, such as *WordNet*.
  wordnetweb.princeton.edu/perl/webwn



S: (n) **nickel**, Ni, atomic number 28 (a hard malleable ductile silvery metallic element that is resistant to corrosion; used in alloys; occurs in pentlandite and smaltite and garnierite and millerite)
S: (n) **nickel** (a United States coin worth one twentieth of a dollar)

- ○ ***direct hypernym*** / *inherited hypernym* / *sister term*
  - S: (n) coin (a flat metal piece (usually a disc) used as money)



- Different libraries for such measures exist.

Similarity between Strings

Distributional Similarity

Limitation of thesaurus methods

- Many words and virtually all phrases are missing.
- Verbs and adjectives are not as hierarchically structured as nouns.
- Thesauri are not available for all languages.

"You shall know a word by the company it keeps!" (Firth, 1957)

Idea of distributional similarity

- Two words are similar, if they have similar word contexts, i.e., if they have similar words around them.
- Two words are synonyms, if they have almost identical contexts.

"Many people like tesgüino." "A bottle of tesgüino is on the table."
"Tesgüino makes you drunk." "Tesgüino is brewed from cereal grains."

→ An alcoholic beverage like beer.

Similarity between Strings

Pointwise Mutual Information

Word-context matrix

- Cooccurrences of words in a corpus within a window of some number of words (say, 20).

	computer	data	pinch	result	sugar
apricot	0	0	1	0	1
pineapple	0	0	1	0	1
digital	2	1	0	1	0
information	1	6	0	4	0

Pointwise mutual information (PMI) of word pairs

- Quantifies whether two words w_i and w_j co-occur more than if they were independent.

$$PMI(w_i, w_j) \;=\; log_2 \frac{P(w_i, w_j)}{P(w_i) \cdot P(w_j)}$$

- Extensions. Avoid bias towards infrequent words, consider syntax, ...

  Often, the *positive PMI (PPMI)* is considered where all values < 0 are replaced with 0.

PMI approximated based on the matrix

P(“information“, “data”) $= \frac{6}{19} = 0.32$ P(“information”) $= \frac{11}{19} = 0.58$ P(“data”) $= \frac{7}{19} = 0.37$

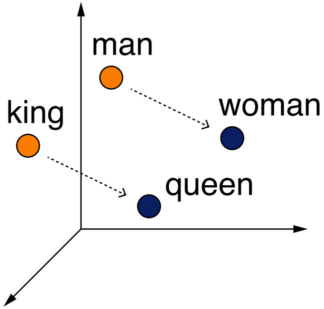
→ PMI(“information”, “data”) $= log_2 \frac{0.32}{0.37 \cdot 0.58} = 0.58$

Similarity between Strings

Word Embeddings

Extension of the distributional idea

- Representation of a word by the context it occurs in.
- To do so, words are mapped to an *embedding space* where contextually related words are similar.



Word embedding (aka word vector)

- A real-valued vector that represents the *distributional semantics* of a particular word in the embedding space.

$$\text{“king”} \quad \rightarrow \quad v_{king} = (0.13, 0.02, 0.1, 0.4, \ldots, 0.22)$$

- The longer the vector, the more variance is kept (typical: 100–500).

Some properties of embedding spaces

- Similar context results in similar embeddings. projector.tensorflow.org
- Analogies are arithmetically represented. turbomaze.github.io/word2vecjson

$$v_{king} - v_{man} + v_{woman} \approx v_{queen} \qquad v_{france} - v_{paris} + v_{berlin} \approx v_{germany}$$

Similarity between Strings

Embedding Models

Word embedding model

- A function that maps each known word to its word embedding.
- Such mappings are created unsupervised based on huge corpora, capturing the likelihood of words occurring in sequence.

  The technical details are beyond the scope of this course.

Several software libraries and pre-trained models exist

- Libraries. Glove, word2vec, Fasttext, Flair, Bert, ...
- Models. GoogleNews-vectors, ConceptNet Numberbatch, ...

From word embeddings to text embeddings

- Simple. Average the embeddings of each word in a text.
- More sophisticated. Learn embeddings for sentences or similar.
- In general, the longer the text, the harder it is to capture its semantics in an embedding.

Similarity between Strings

From Strings back to Texts

Encoding similarities in feature vectors

- String similarities can be used in diverse ways within features.

Frequency of ~~“money”~~ the sense “the most common medium of exchange”

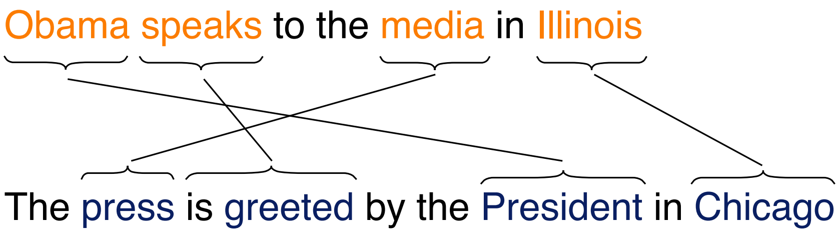
Frequency of all writings of “traveling”

- Embeddings may simply be used as feature vectors.

“nickel” → (0.14, 0.03, 0.44, . . . , 0.22)   “money” → (0.18, 0.06, 0.49, . . . , 0.01)

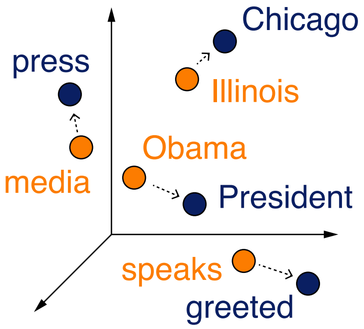
Word Mover’s Distance (Kusner et al., 2015)

- The distance of the optimal alignment of two texts.
- Represents texts by sequences of word embeddings.



Obama speaks to the media in Illinois

The press is greeted by the President in Chicago

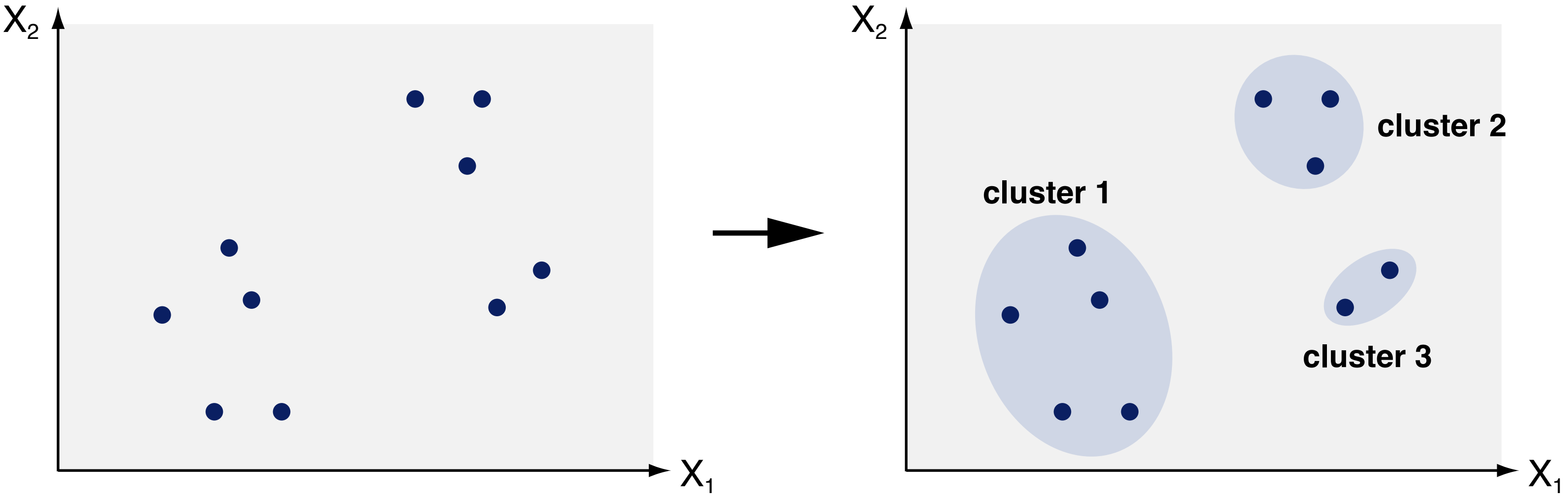


Hard and Soft Flat Clustering

Hard Flat Clustering

What is hard flat clustering?

- A clustering 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.

Hard Flat Clustering

Two Main Types of Algorithms

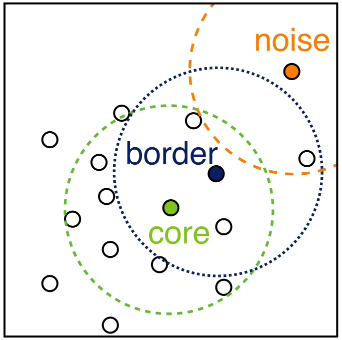
Iterative algorithms

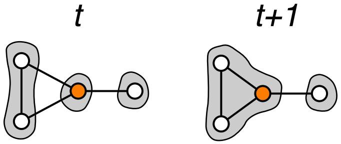
- Iterative clustering and re-assignment of instances to clusters.
- Exemplar-based. Instances are considered in isolation when adding them to clusters (e.g., k-*means*).

  We focus on this type here.
- Exchange-based. Instances are exchanged between pairs of clusters (e.g., *Kerninghan-Lin*).

Density-based algorithms

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

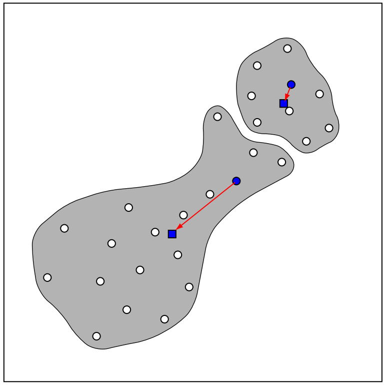




Flat Clustering with k-means

What is k-means?

- A simple hard flat clustering algorithm that creates $k \geq 1$ clusters.
- Instances are assigned to the cluster whose *centroid* is closest to them, i.e., the mean of all instances in a cluster.
- 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 includes a maximum number of iterations.
- Medoid. A generalization of the centroid computed in some way. Algorithms based on the medoid are not called k-means anymore.

Flat Clustering with k-means

Pseudocode

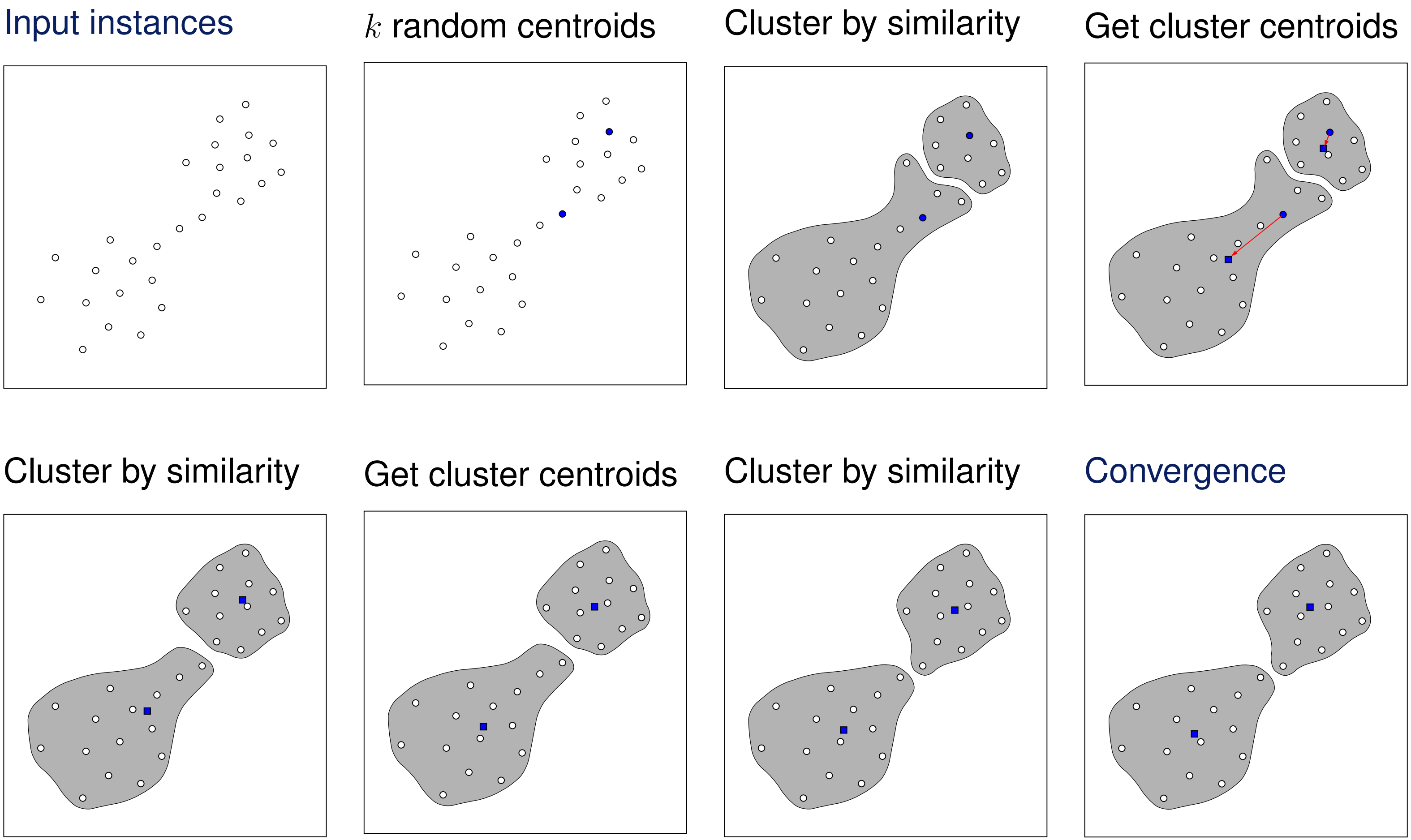
Signature

- 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 $\leftarrow \emptyset$
    Instance [] centroids $\leftarrow$ chooseRandomInstances($X$, $k$)
    repeat
        Instance [] prevCentroids $\leftarrow$ centroids
        for int i $\leftarrow$ 1 to $k$ do clusters[i] $\leftarrow \emptyset$
        for each x $\in X$ do // create clusters
            int z $\leftarrow$ 1
            for int j $\leftarrow$ 2 to $k$ do // find nearest centroid
                if $sim$(x,centroids[j]) > $sim$(x,centroids[z]) then z$\leftarrow$j
            clusters[z] $\leftarrow$ clusters[z] $\cup$ {x}
        for int i $\leftarrow$ 1 to $k$ do // update centroids
            centroids[i] $\leftarrow$ computeMeans(clusters[i])
    until prevCentroids = centroids // convergence
    return clusters
```

Flat Clustering with k-means

Example for $k = 2$ (recap)



Input instances

k random centroids

Cluster by similarity

Get cluster centroids

Cluster by similarity

Get cluster centroids

Cluster by similarity

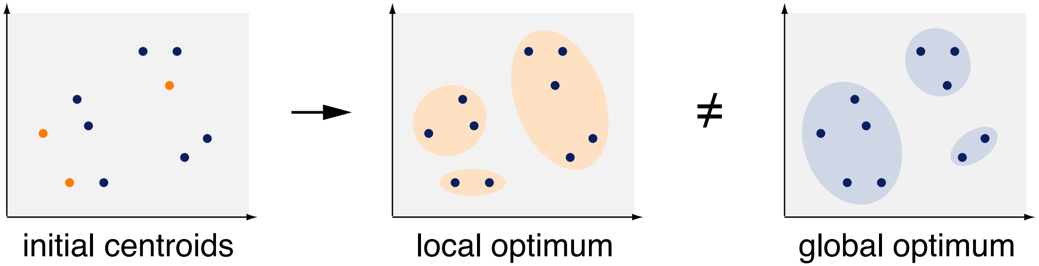
Convergence

Flat Clustering with k-means

Random Initialization

Problem

- Assume we are given an intrinsic or extrinsic cost function L, i.e., there is an optimal clustering.
- k-means converges when it has found a local mininum.
- Due to the random choice of centroids, it may not find the global one.



Approach

- To account for this, k-means may be repeated several (e.g., 100) times.
- The best found local optimum is chosen then.
- An alternative is to pick good initial centroids using expert knowledge.

Flat Clustering with k-means

Number of Clusters

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 $k = 1$
- Intrinsic. Maximum cluster distance. $\rightarrow$ highest for $k = |X|$
- Extrinsic. Purity of clusters. $\rightarrow$ 1.0 for $k = |X|$
- Extrinsic. Macro/Micro F_1-score. $\rightarrow$ 1.0 for $k = |X|$

Approaches

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

  Both approaches have a visual intuition, but work mathematically.

Flat Clustering with k-means

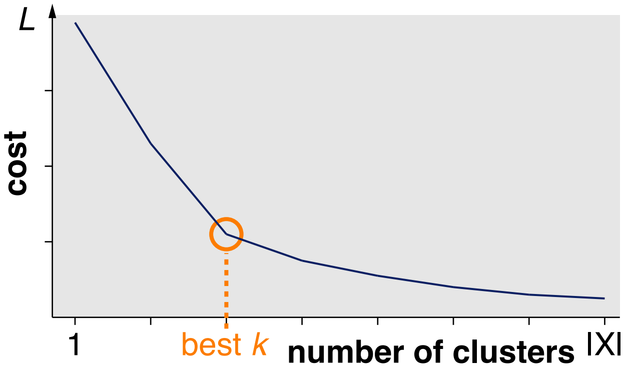
Elbow Criterion

What is the elbow criterion?

- A method to find the best value of a hyperparameter, e.g., k in k-means.

  Other algorithms also have hyperparameters, e.g., DBSCAN has a neighborhood size.

- Requires some cost function L.



Input

- A *set* of clusterings $\mathbf{C} = \{C_1, \ldots, C_p\}$ for hyperparameter values $k_1, \ldots, k_p$.
- A cost $L(C_i)$ 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 k with the maximum second derivate.

  This reflects the point where the cost reduction changes strongest.

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

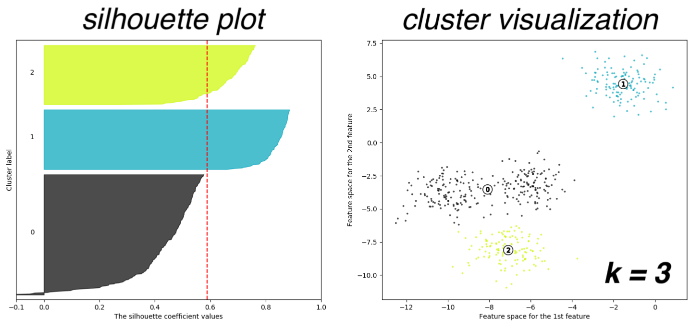
Flat Clustering with k-means

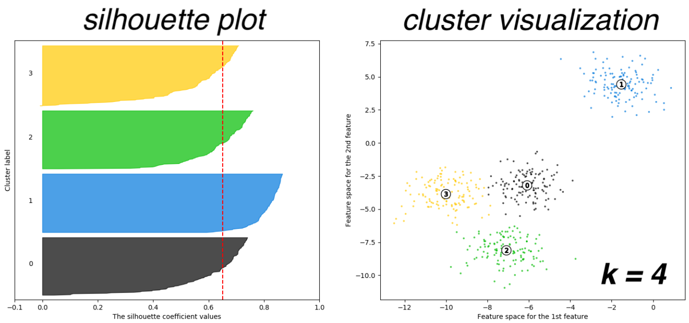
Silhouette Analysis

What is 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

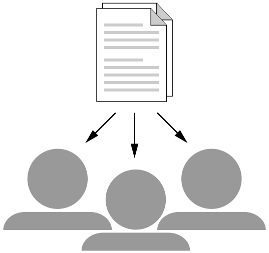
- Visually. Pick the k where many scores (x-axis) are above average, and where the cluster size (y-axis) is balanced.

  Multiple similar candidates might be hard to choose between.

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

Authorship Attribution

What is authorship attribution?



- The text analysis that reveals the authors of texts.
- Tackled in text mining as a downstream task.

  Related tasks: Authorship verification, plagiarism detection, ...

Settings

- Supervised. Given a set of n training texts with p known authors, learn a mapping from texts to authors.
- Unsupervised. Given a set of n training texts (usually assumed to be single-authored), group them by their author.

Observations

- Unlike in most tasks, computers tend to be better than humans here.
- Features that capture style are mostly in the focus.
- Some successful features capture subconscious language use.

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

Authorship Attribution

CLEF 2016 Shared Task on Author Clustering *

Shared task

- Participants develop competing approaches for the same task and data.

Task definition "Author Clustering"

- Given a corpus with up to 100 texts, identify the number k of authors and assign each text to the cluster representing its author.
- Training sets are given; results are averaged over unseen test sets.

18 training sets and test sets

- Six sources. Opinion articles and reviews in Dutch, English, and Greek.
- Three datasets per source. Differ in terms of the number of authors.
- Most texts range between 400 and 800 words.

Eight participating teams

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

Authorship Attribution

k-means Approaches in the Shared Task *

Mansoorizadeh et al.

- Features. Word and POS unigrams and bigrams, sentence lengths, punctuation n-grams with $n \geq 2$.

  Texts lower-cased, no features discarded, feature values normalized.
- Similarity. Cosine score.
- Choosing k. Creation of a similarity graph with similarity threshold 0.5. The number of subgraphs defines the k used for k-means.

Sari and Stevenson

- Features. TF-IDF on the 5000 top character n-grams with $n \in \{3, \ldots, 8\}$, average word embeddings.

  Embeddings: GoogleNews-vector (English), self-trained (Dutch), none (Greek).
- Similarity. Cosine score.
- Choosing k. Silhouette analysis based on k-means. The k with the highest Silhouette score ist taken.

Authorship Attribution

Shared Task Results Averaged over All Test Sets *

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

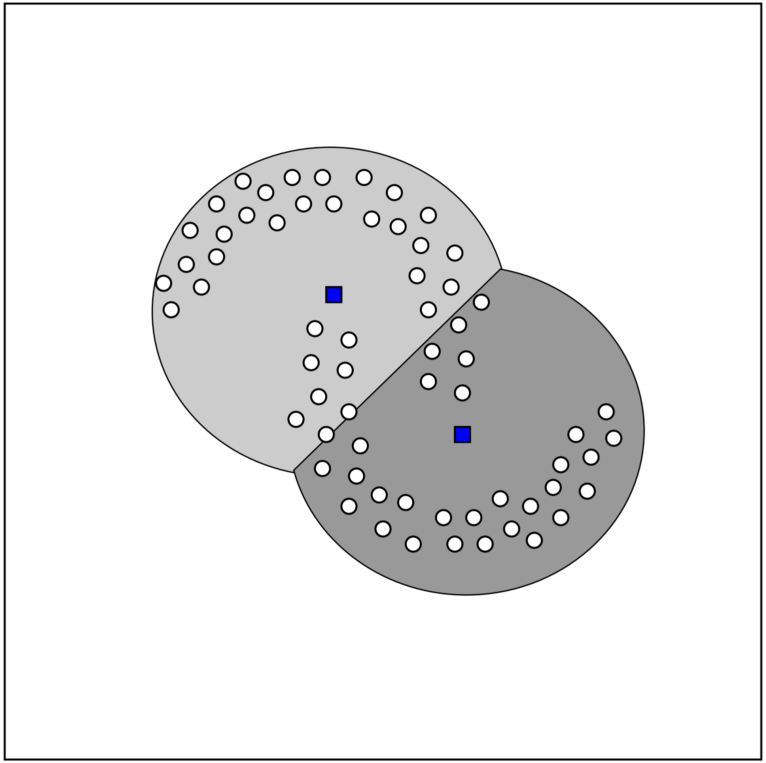
- B^3 precision. Proportion of texts in the cluster of d by the author of d.
- B^3 recall. Proportion of texts by the author of d found in the cluster of d.

The values are averaged over all texts. F$_1$-score as usual.

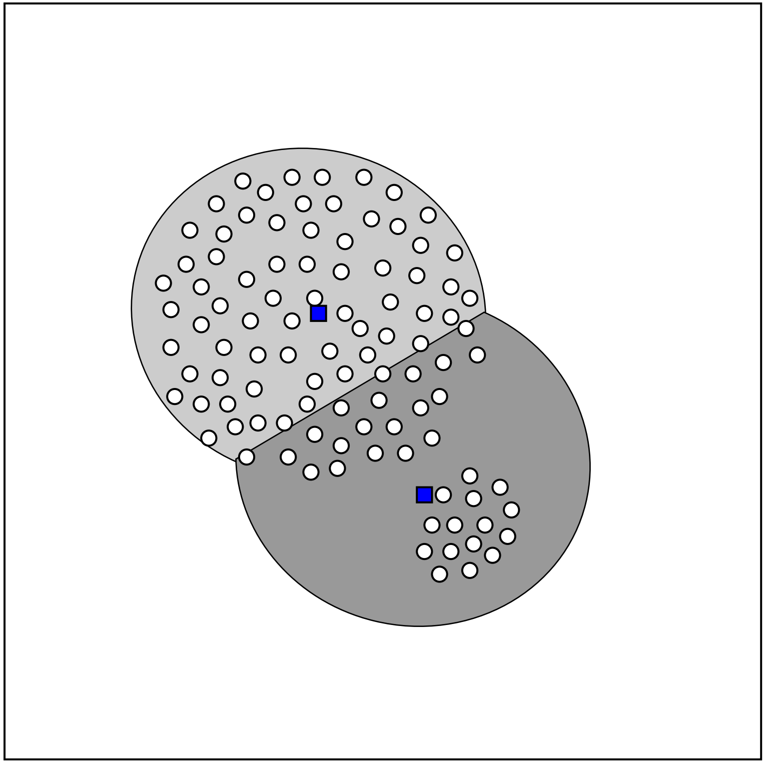
Hard Flat Clustering

Issues with Iterative, Exemplar-based Clustering Algorithms

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



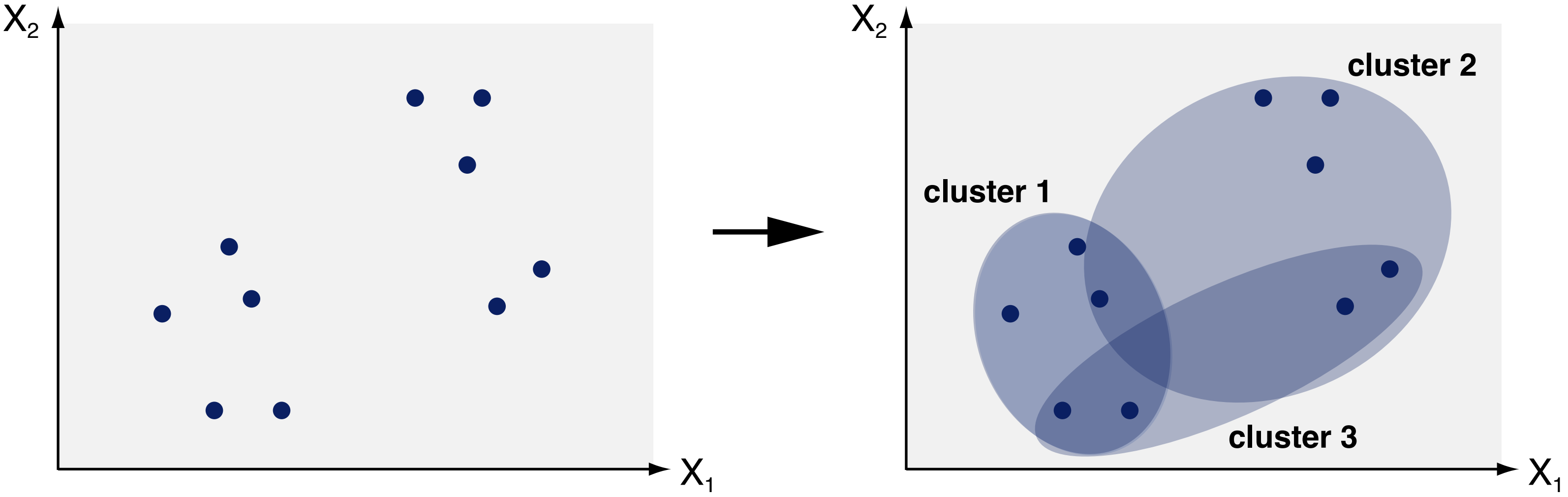
Similarly, they fail to detect clusters with large difference in size.



Soft Flat Clustering

What is soft flat clustering?

- A clustering 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 weighted mapping $X \rightarrow \{(c, w_c) \mid c \in C, w \in [0, 1]\}$, such that $\forall \mathbf{x}^{(i)} \in X : \sum_{c \in C} w_c^{(i)} = 1$.



Number of clusters k

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

Soft Flat Clustering

Idea and Algorithms

Idea of soft clustering

- Given the following five sentences:

  “Max likes to eat broccoli and bananas.” → 1.0 topic A
  “Tim had a banana and spinach smoothie for breakfast.” → 1.0 topic A
  “Dogs and cats are pets.” → 1.0 topic B
  “Linda adopted a cat yesterday.” → 1.0 topic B
  “The hamster munches on a piece of broccoli.” → 0.6 topic A, 0.4 topic B

- A clustering algorithm might identify two soft clusters:

  Topic A representing food   Topic B representing pets

- Each sentence can then be assigned a weight for each cluster.

Selected algorithms used for soft clustering

- Fuzzy k-means clustering
- Gaussian mixture models
- Latent Dirichlet allocation

Topic Modeling

What is 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?

- Finds low-dimensional representations of high-dimensional text.
- Injects some kind of meaning into a vocabulary.
- Enables to concisely summarize texts and to capture their similarity.

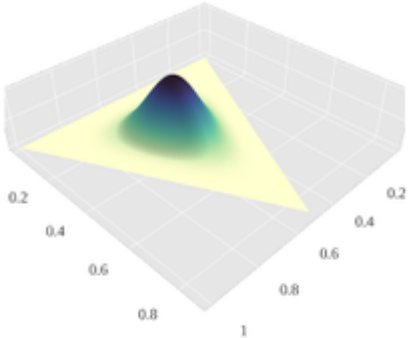
How to do topic modeling?

- A number of 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.
- Machine learning toolkits such as *scikit-learn* include LDA.

Topic Modeling

Latent Dirichlet Allocation (LDA)



What is 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 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.

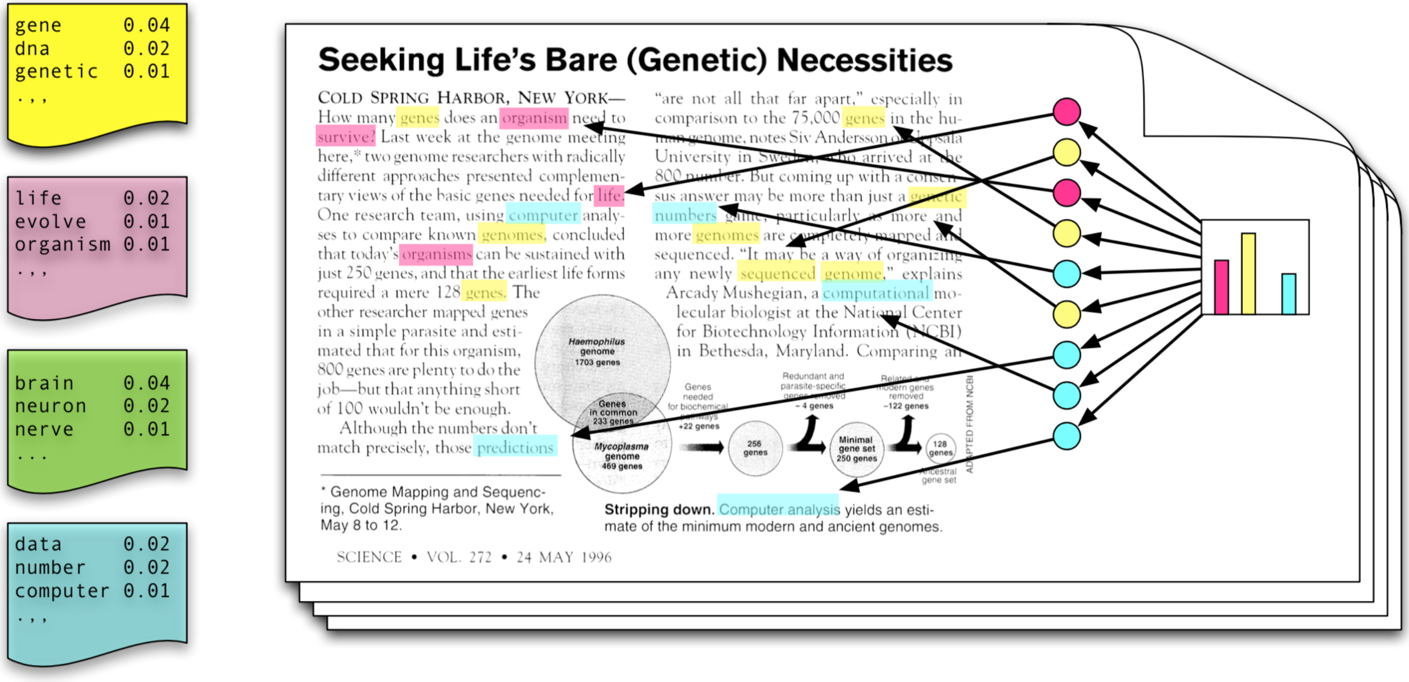
Notice

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

  The mathematical details are beyond the scope of this course.

Topic Modeling

Assumptions behind LDA



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
...	

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—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 in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways +22 genes
256 genes
Redundant and parasite-specific genes removed – 4 genes
Minimal gene set 250 genes
Related and modern genes removed –122 genes
128 genes
Ancestral gene set
ADAPTED FROM NCBI

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

[*] Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

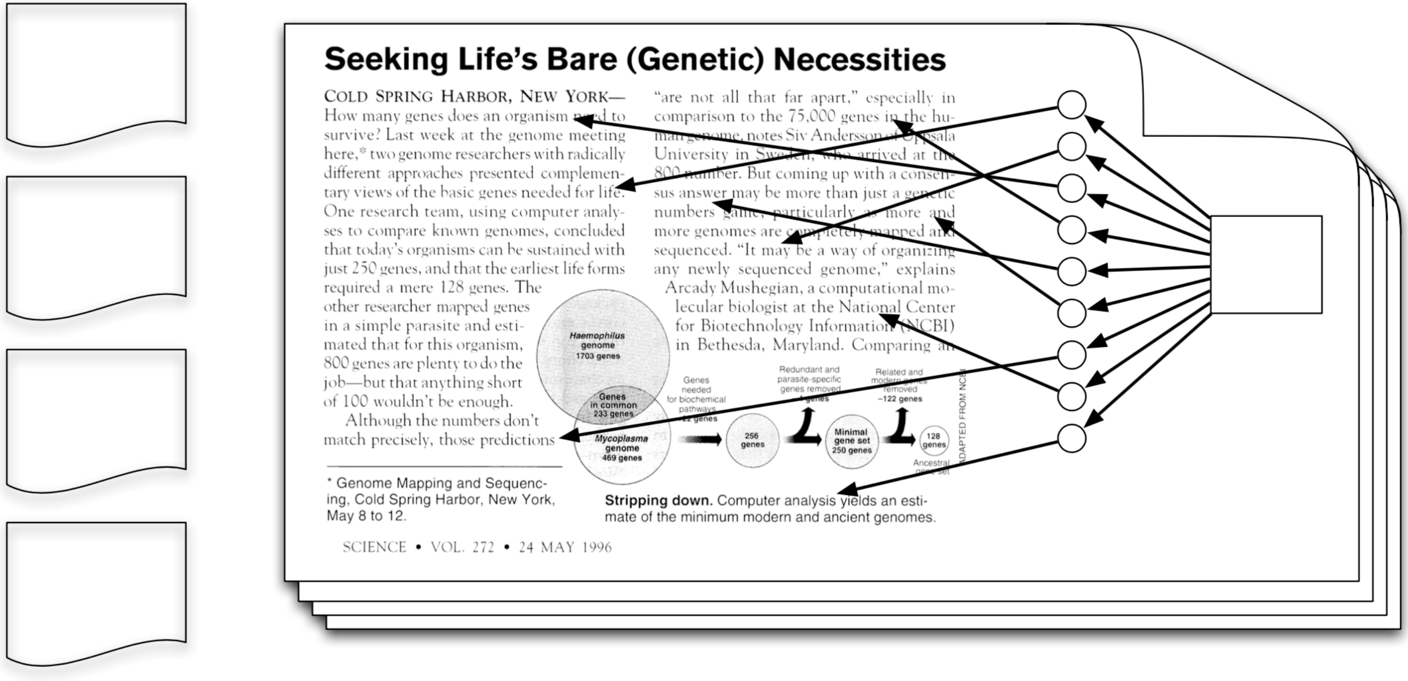
SCIENCE • VOL. 272 • 24 MAY 1996

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



Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—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 in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways +22 genes
256 genes
Redundant and parasite-specific genes removed −4 genes
Minimal gene set 250 genes
Related and modern genes removed −122 genes
128 genes
Ancestral gene set
ADAPTED FROM NCBI

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

SCIENCE • VOL. 272 • 24 MAY 1996

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 k of topics 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

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

Topic Modeling

Example Topic Models *

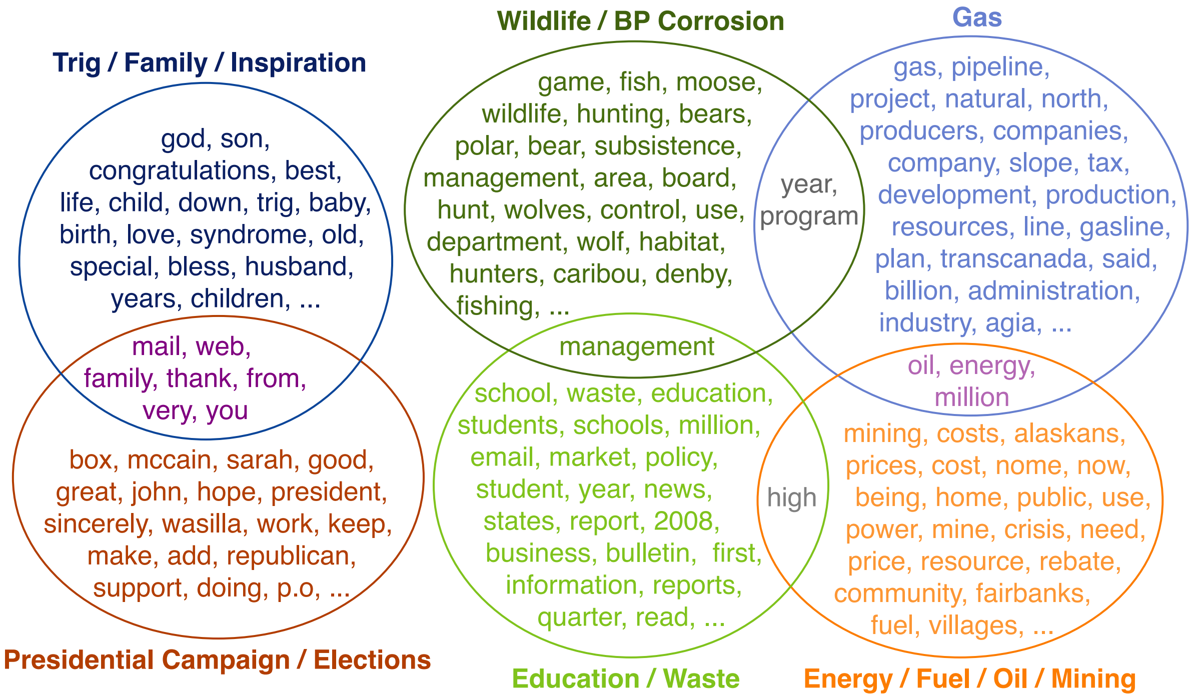
Case study

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)



Trig / Family / Inspiration

god, son, congratulations, best, life, child, down, trig, baby, birth, love, syndrome, old, special, bless, husband, years, children, ...

mail, web, family, thank, from, very, you

box, mccain, sarah, good, great, john, hope, president, sincerely, wasilla, work, keep, make, add, republican, support, doing, p.o, ...

Presidential Campaign / Elections

Wildlife / BP Corrosion

game, fish, moose, wildlife, hunting, bears, polar, bear, subsistence, management, area, board, hunt, wolves, control, use, department, wolf, habitat, hunters, caribou, denby, fishing, ...

year, program

management

Gas

gas, pipeline, project, natural, north, producers, companies, company, slope, tax, development, production, resources, line, gasline, plan, transcanada, said, billion, administration, industry, agia, ...

oil, energy, million

school, waste, education, students, schools, million, email, market, policy, student, year, news, states, report, 2008, business, bulletin, first, information, reports, quarter, read, ...

high

mining, costs, alaskans, prices, cost, nome, now, being, home, public, use, power, mine, crisis, need, price, resource, rebate, community, fairbanks, fuel, villages, ...

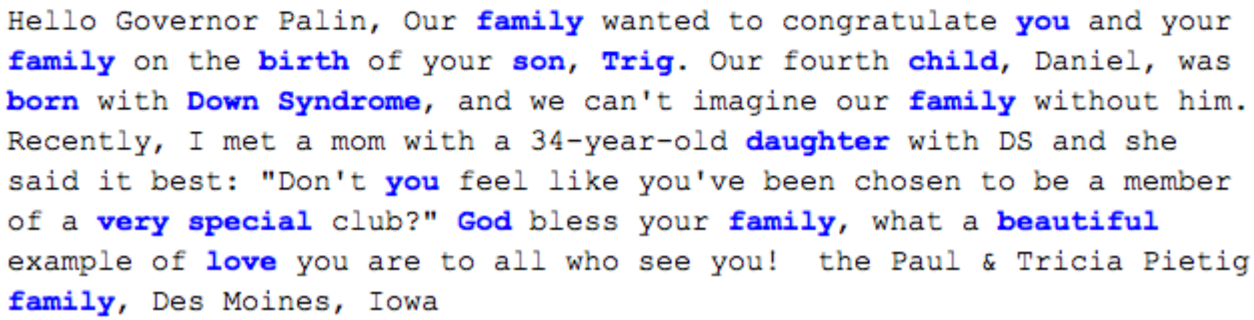
Education / Waste

Energy / Fuel / Oil / Mining

Topic Modeling

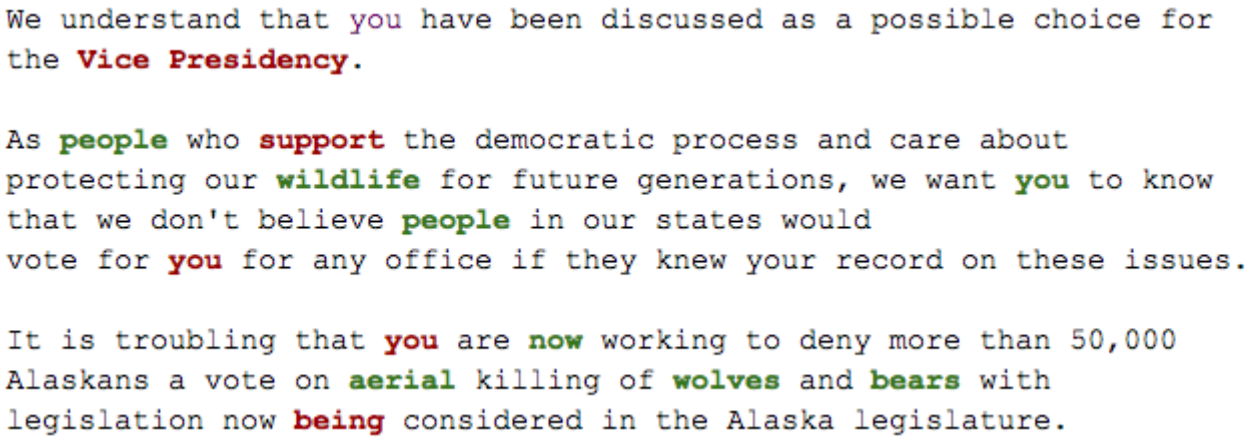
Example Texts with Highlighted Topic Words *

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.

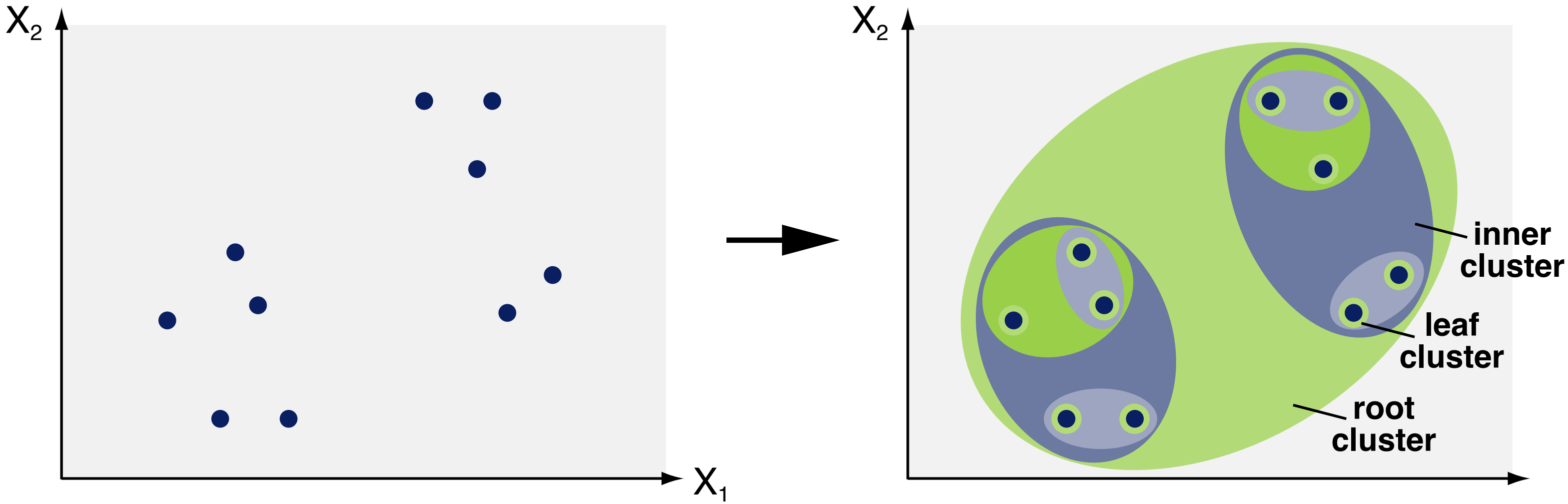
It is troubling that **you** are **now** working to deny more than 50,000 Alaskans a vote on **aerial** killing of **wolves** and **bears** with legislation now **being** considered in the Alaska legislature.

Hierarchical Clustering

Hierarchical Clustering

What is hierarchical clustering?

- A clustering that creates a binary tree over instances, which represents the sequential merging of the 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

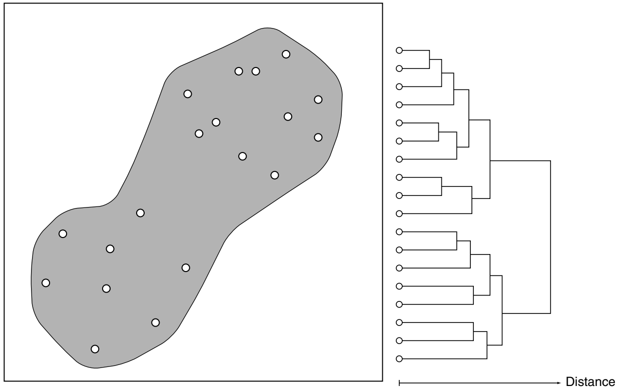
- A flat clustering can be derived via cuts in the hierarchy tree.

Hierarchical Clustering

Two Main Techniques

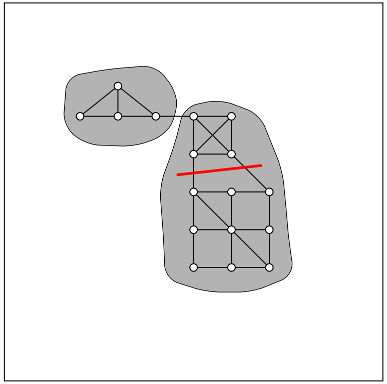
Agglomerative hierarchical clustering

- Incrementally create tree bottom-up, beginning with single instances.
- Merge clusters based on the distances between the instances they contain.



Divise hierarchical clustering

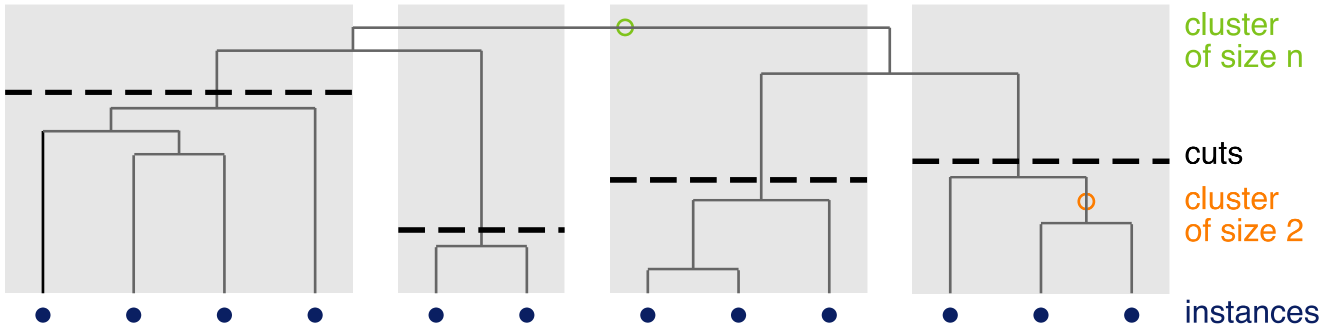
- Incrementally divide instances into smaller clusters (top-down).
- The most common algorithm is *MinCut*. It models the set of instances as a weighted graph.
- MinCut repeatedly splits clusters by finding the minimum cut in a subgraph.



Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering in a nutshell

- Merge closest pair of clusters.
- Represent clusters and how they are merged as a dendrogram.
- Repeat until only one cluster remains.



Dendrograms

- A dendrogram is a diagram representing a tree.
- Used to visualize what clusters are merged and when they are merged.
- Also, they may illustrate cuts done to obtain a flat clustering.

Agglomerative Hierarchical Clustering

Pseudocode

Signature

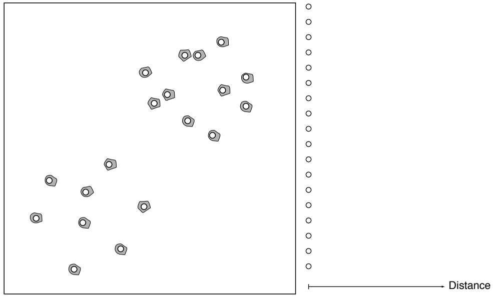
- Input. A set of instances X.
- Output. A binary tree $\langle V, E \rangle$ containing all clusters.

```
agglomerativeHierarchicalClustering(Set<Instance> $X$)
    Set<Set<Instance>> clusters ← $\{\{\mathbf{x}^{(i)}\} \mid \mathbf{x}^{(i)} \in X\}$ // curr. clusters
    Set<Set<Instance>> $V$ ← clusters // tree nodes
    Set<Set<Instance>[]> $E$ ← $\emptyset$ // 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 $\langle V, E \rangle$
```

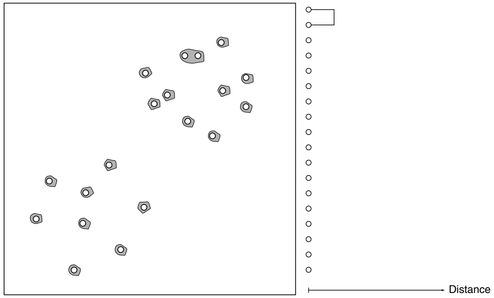
Agglomerative Hierarchical Clustering

Example (recap)

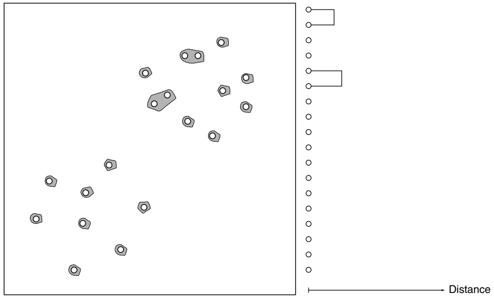
One cluster per instance



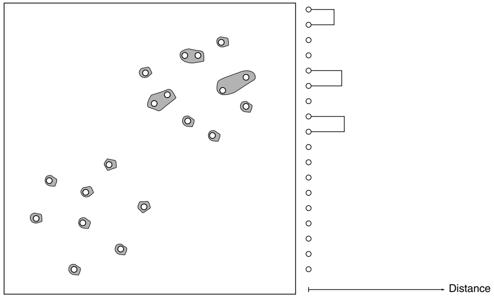
Merge closest cluster pair



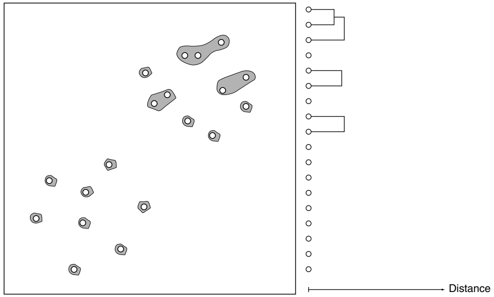
Repeat cluster merging



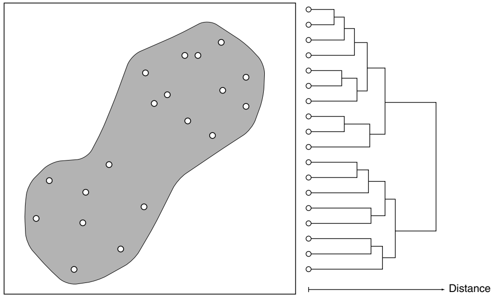
Repeat cluster merging



Repeat cluster merging ...



Terminate with one cluster



Agglomerative Hierarchical Clustering

Similarity Re-Computation after each Merging Step

$t = 0$

	C_1	C_2	...	C_n
C_1	0	$sim(C_1, C_2)$	...	$sim(C_1, C_n)$
C_2	-	0	...	$sim(C_2, C_n)$
$\vdots$				
C_n	-	-	...	0

$\equiv$

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	...	$\mathbf{x}^{(n)}$
$\mathbf{x}^{(1)}$	0	$sim(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$	...	$sim(\mathbf{x}^{(1)}, \mathbf{x}^{(n)})$
$\mathbf{x}^{(2)}$	-	0	...	$sim(\mathbf{x}^{(2)}, \mathbf{x}^{(n)})$
$\vdots$				
$\mathbf{x}^{(n)}$	-	-	...	0

↓

$t = i$

	C_{i_1}	C_{i_2}	$\ldots$	$C_{i_{n-i}}$
C_{i_1}	0	$sim(C_1, C_{i_2})$	$\ldots$	$sim(C_{i_1}, C_{i_{n-i}})$
C_{i_2}	-	0	$\ldots$	$sim(C_{i_2}, C_{i_{n-i}})$
$\vdots$				
$C_{i_{n-i}}$	-	-	$\ldots$	0

↓

$t = n - 1$ C_{n_1}

Agglomerative Hierarchical Clustering

Cluster Similarity

Cluster similarity has two components

- Measure. Captures similarity of instances (or cluster representatives).

  Same measures as before: Cosine similarity, euclidean similarity, ...
- Aggregation. Captures cluster similarity based on instance similarity.

How to aggregate similarity?

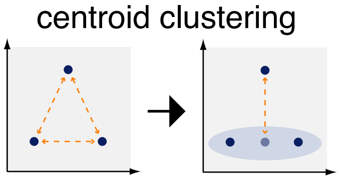
- Several methods to obtain cluster similarity with different characteristics exist. They may result in fully different clusterings.
- Most common. *Single link*, *complete link*, *group-average link*.

  A few other methods exist but are omitted here, e.g., the *Ward criterion*.

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

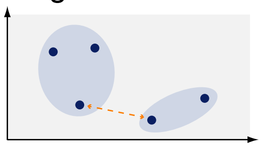
Cluster Similarity Aggregation Methods

Single link clustering

- Using 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

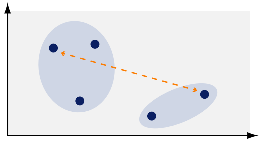


Complete link clustering

- Using 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

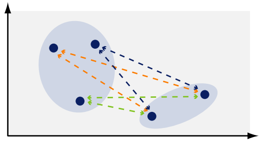


Group-average link clustering

- Averaging 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



Agglomerative Hierarchical Clustering

Characteristics of Aggregation Methods

Overview of characteristics

Characteristic	**Single link**	**Complete link**	**Group-average link**
Cluster number	Low	High	Medium
Cluster form	Extended	Small	Compact
Chaining tendency	High	Low	Low
Outlier visibility	High	Low	Medium
Noise susceptibility	High	High	Low
Monotonicity	✓	✓	✓
Run-time complexity	$\mathcal{O}(V^2)$	$\mathcal{O}(V^2 \cdot \text{log}(V))$	$\mathcal{O}(V^2 \cdot \text{log}(V))$

The *cluster number* refers to intermediate steps of the clustering process.

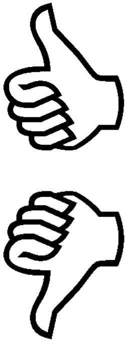
Remarks

- Single link. Can be computed efficiently with a minimum spanning tree.
- Complete link. Much faster than group-average link in practice.
- Group-average link. May also detect “potato-shaped” clusters.

  Due to run-time, single link and complete link are most popular.

Review Sentiment Analysis

What is sentiment analysis?



- The text analysis that assesses whether a text or a span of text conveys sentiment.
- One of the most tackled downstream tasks in text mining.

  HIgh industrial importance, e.g., in reputation management.
- Usually tackled with supervised classification.

Sentiment: Polarity vs. scores

- Polarity. *Positive* or *negative*, sometimes also *neutral* or similar.
- Scores. Scores from a numeric scale, e.g., $\{1, \ldots, 5\}$ or $[0, 1]$.
- Related. Subjectivity, emotion, stance, and similar.

Reviews

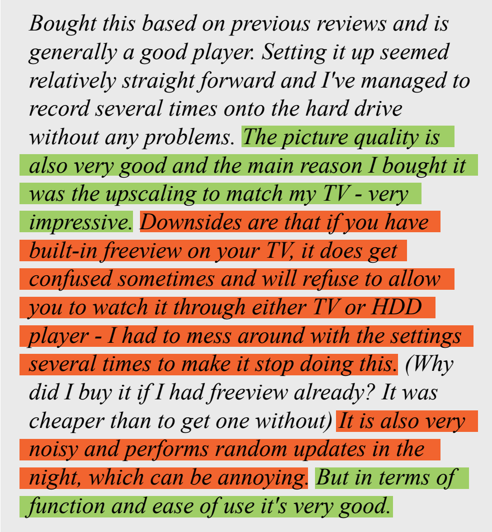
- Consumer judgments of products, services, and works of arts.

  For example, reviews of electronic devices, books, hotels, movies, etc.
- Reviews often comprise several “local” sentiments on different aspects.

Review Sentiment Analysis

Reviews across Topical Domains

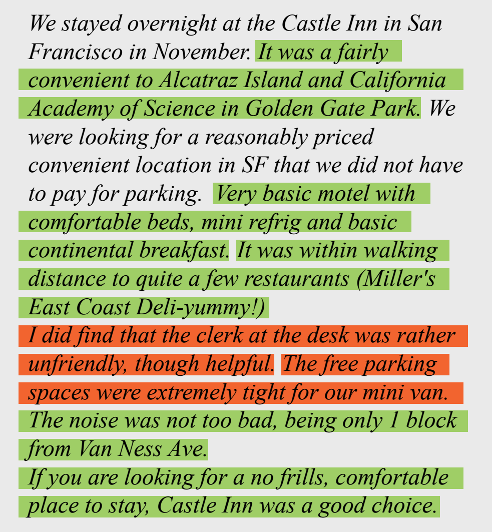
Product review
from Amazon



Bought this based on previous reviews and is generally a good player. Setting it up seemed relatively straight forward and I've managed to record several times onto the hard drive without any problems. The picture quality is also very good and the main reason I bought it was the upscaling to match my TV - very impressive. Downsides are that if you have built-in freeview on your TV, it does get confused sometimes and will refuse to allow you to watch it through either TV or HDD player - I had to mess around with the settings several times to make it stop doing this. (Why did I buy it if I had freeview already? It was cheaper than to get one without) It is also very noisy and performs random updates in the night, which can be annoying. But in terms of function and ease of use it's very good.

Global sentiment:
neutral (3 out of 5)

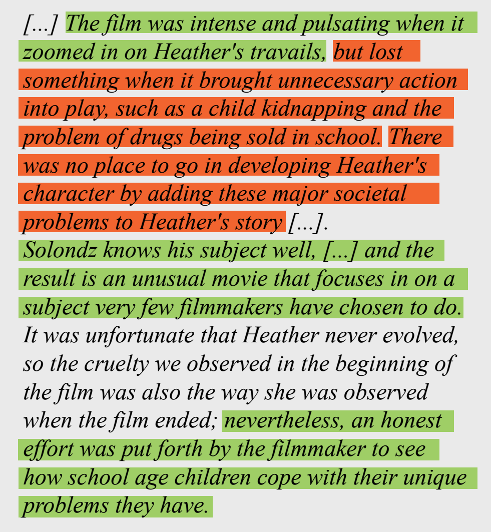
Hotel review
from TripAdvisor



We stayed overnight at the Castle Inn in San Francisco in November. It was a fairly convenient to Alcatraz Island and California Academy of Science in Golden Gate Park. We were looking for a reasonably priced convenient location in SF that we did not have to pay for parking. Very basic motel with comfortable beds, mini refrig and basic continental breakfast. It was within walking distance to quite a few restaurants (Miller's East Coast Deli-yummy!) I did find that the clerk at the desk was rather unfriendly, though helpful. The free parking spaces were extremely tight for our mini van. The noise was not too bad, being only 1 block from Van Ness Ave. If you are looking for a no frills, comfortable place to stay, Castle Inn was a good choice.

Global sentiment:
neutral (3 out of 5)

Movie review
from Rotten Tomatoes



[...] The film was intense and pulsating when it zoomed in on Heather's travails, but lost something when it brought unnecessary action into play, such as a child kidnapping and the problem of drugs being sold in school. There was no place to go in developing Heather's character by adding these major societal problems to Heather's story [...]. Solondz knows his subject well, [...] and the result is an unusual movie that focuses in on a subject very few filmmakers have chosen to do. It was unfortunate that Heather never evolved, so the cruelty we observed in the beginning of the film was also the way she was observed when the film ended; nevertheless, an honest effort was put forth by the filmmaker to see how school age children cope with their unique problems they have.

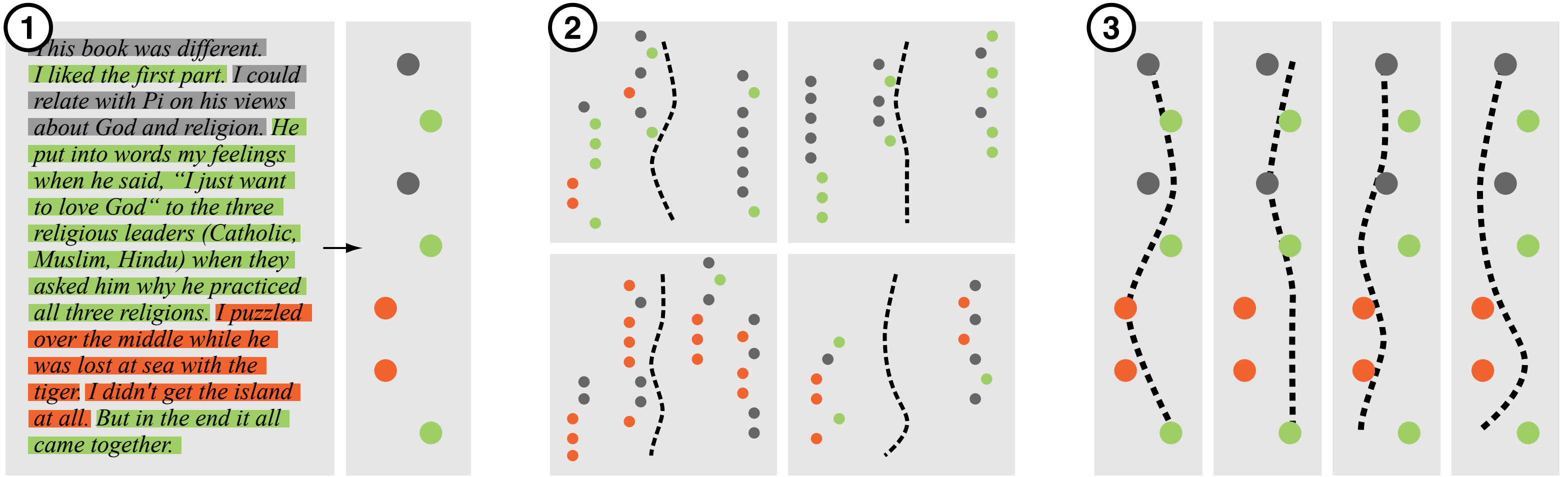
Global sentiment:
neutral (2 out of 3)

Review Sentiment Analysis

Flow Pattern Recognition *

Idea for cross-domain review sentiment analysis

- Model review discourse by the *local sentiment flow* in the review.
- Hypothesis. Similar flows occur across review domains.



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.

Approach

1. Represent a review by its 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.

Review Sentiment Analysis

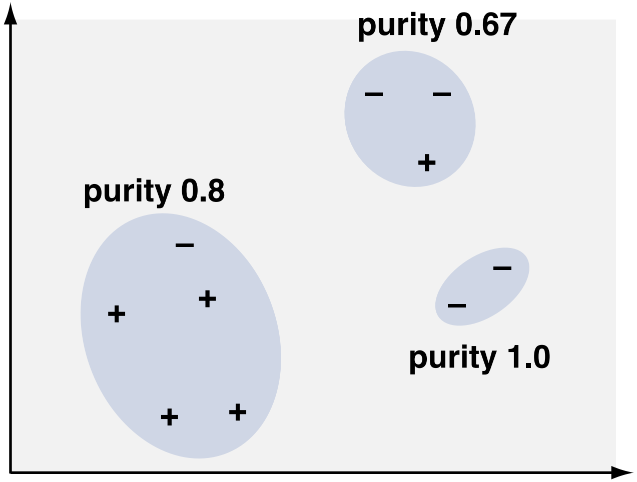
Flow Patterns as Features

Idea of the flow patterns

- Used as complex features in review sentiment analysis.
- Should indicate global sentiment and have high commonness.

Supervised clustering (recap)

- Cluster instances with known classes.
- The *purity* of clusters can be measured, i.e., the fraction of instances whose class equals the majority class.
- Goal. Ensure that all clusters have a certain minimum purity τ.



Goals of flow clustering

- Maximize the purity of the clusters
- Minimize the number of clusters, i.e., maximize their mean size.

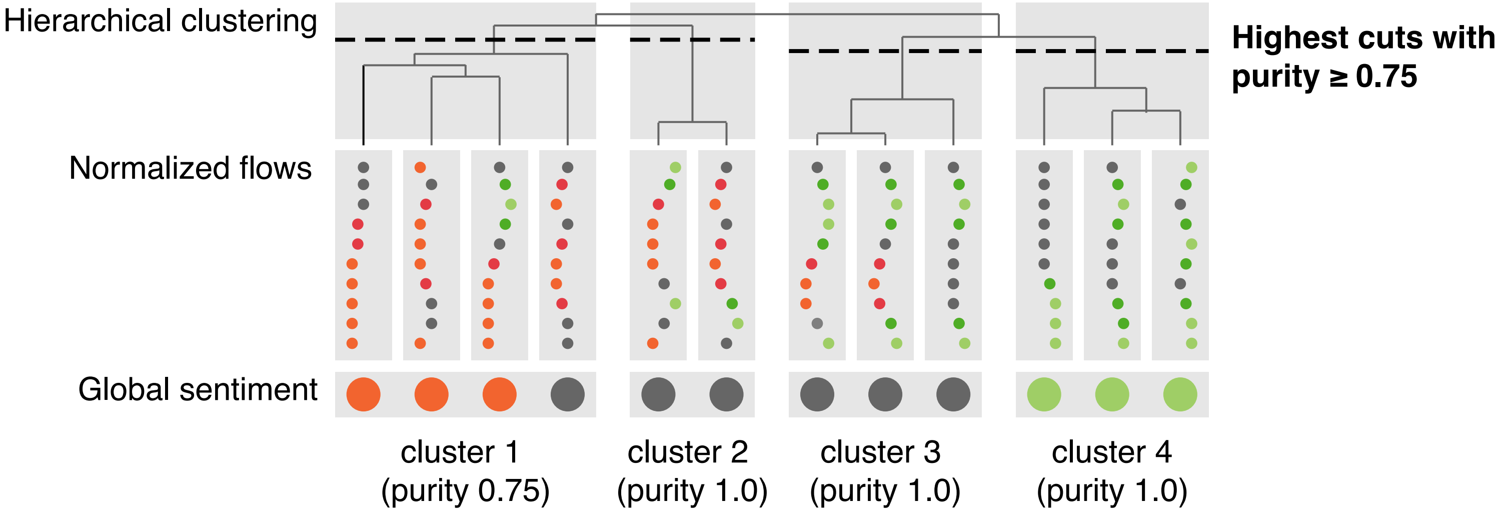
Review Sentiment Analysis

Supervised Clustering of Flows

Flow clustering process

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

Example for τ = 0.75



Review Sentiment Analysis

From Clusters to Patterns *

How to obtain 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.

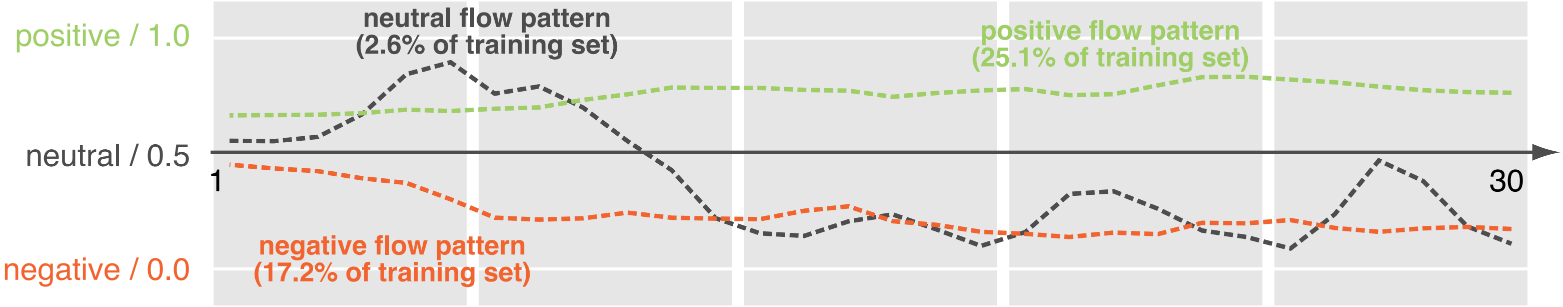
- Local sentiment can be interpreted as a real value in $[0,1]$. The mean of all flows in a cluster may then define a flow pattern.

  Negative: 0.0, neutral 0.5, positive 1.0.

Example flow patterns

- The three most common patterns in 900 TripAdvisor reviews.

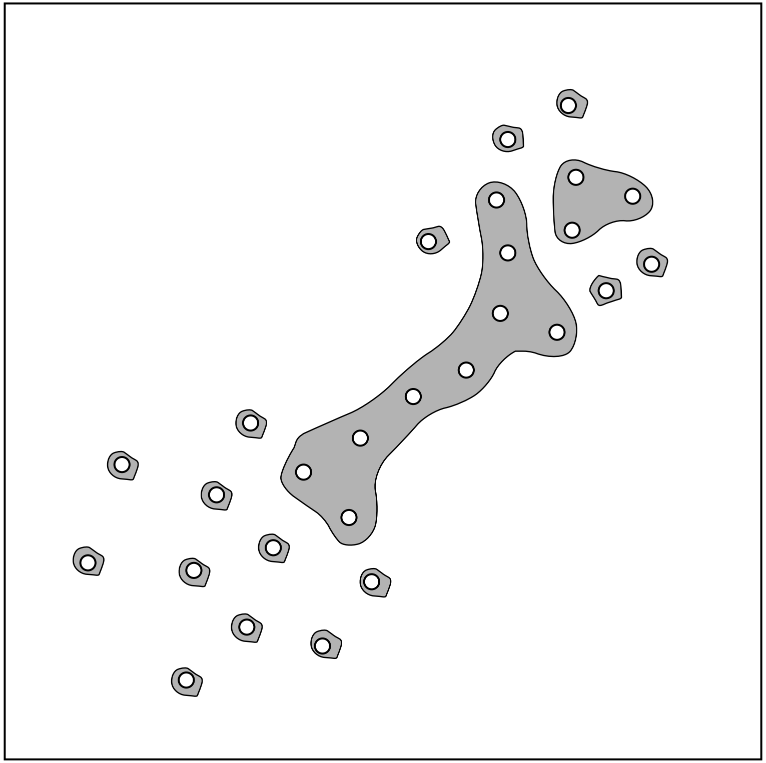
  We will see in the next lecture part how effective they are as features.



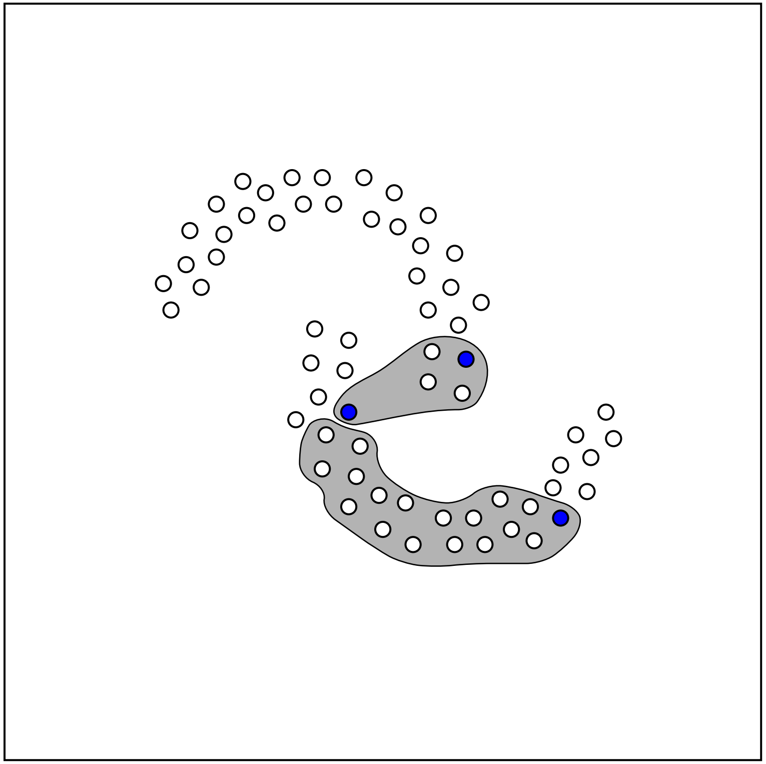
Hierarchical Clustering

Issues with Hierarchical Clustering Algorithms

Chaining problem of clustering using single-link similarity



Nesting problem of clustering using complete-link similarity

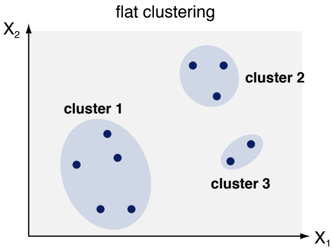


Conclusion

Summary

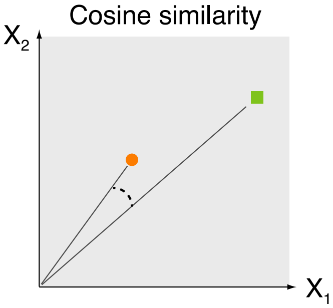
Text Mining using clustering

- Mostly unsupervised grouping of texts and text spans.
- Targets situations where classes are unknown.
- Relies on similarities between instances.



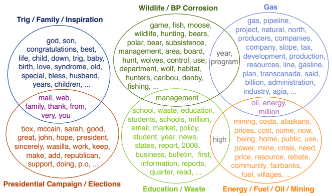
Similarity measures

- Vector-based measures to compare feature vectors.
- Synonyms and embeddings generalize beyond words.
- Clustering often uses cosine similarity or similar.



Clustering techniques

- Hard clustering models disjunct classes of instances.
- Soft clustering (incl. topic modeling) models overlaps.
- Hierarchical clustering stepwise organizes instances.


Trig / Family / Inspiration
god, son, congratulations, best, life, child, down, trig, baby, birth, love, syndrome, old, special, bless, husband, years, children, ...
mail, web, family, thank, from, very, you
box, mccain, sarah, good, great, john, hope, president, sincerely, wasilla, work, keep, make, add, republican, support, doing, p.o, ...
Presidential Campaign / Elections
Wildlife / BP Corrosion
game, fish, moose, wildlife, hunting, bears, polar, bear, subsistence, management, area, board, hunt, wolves, control, use, department, wolf, habitat, hunters, caribou, denby, fishing, ...
year, program
Gas
gas, pipeline, project, natural, north, producers, companies, company, slope, tax, development, production, resources, line, gasline, plan, transcanada, said, billion, administration, industry, agia, ...
management
school, waste, education, students, schools, million, email, market, policy, student, year, news, states, report, 2008, business, bulletin, first, information, reports, quarter, read, ...
Education / Waste
high
oil, energy, million
mining, costs, alaskans, prices, cost, nome, now, being, home, public, use, power, mine, crisis, need, price, resource, rebate, community, fairbanks, fuel, villages, ...
Energy / Fuel / Oil / Mining

References

Some content and examples taken from

- David J. Blei (2012). Probabilistic Topic Models. Tutorial at the 29th International Conference on Machine Learning (ICML 2012). http://www.cs.columbia.edu/~blei/talks/Blei_ICML_2012.pdf
- Sung-Hyuk Cha (2007). Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. International Journal of Mathematical Models and Methods in Applied Sciences, 1(4):300–307.
- Daniel Jurafsky and Christopher D. Manning (2016). Natural Language Processing. Lecture slides from the Stanford Coursera course. https://web.stanford.edu/~jurafsky/NLPCourseraSlides.html.
- Efstathios Stamatatos, Michael Tschuggnall, Ben Verhoeven, Walter Daelemans, Günther Specht, Benno Stein, and Martin Potthast (2016). Clustering by Authorship Within and Across Documents. In Working Notes of the CLEF 2016 Evaluation Labs.
- Benno Stein and Theodor Lettmann (2010). Data Mining. Lecture Slides. https://webis.de/lecturenotes/slides.html#data-mining
- Henning Wachsmuth (2015): Text Analysis Pipelines — Towards Ad-hoc Large-scale Text Mining. LNCS 9383, Springer.
- 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.