Quantitative and qualitative computational analysis of language and text similarities, clustering and classification

Damir Ćavar
CLS 2010, September 2010
University of Zadar

Agenda

- Term Weighting
- Geometrical approaches:
  - Vector spaces
  - Clustering

Term Weighting

tf-idf weighting

- Term frequency
  - Specific documents mention specific terms more frequently than others.
  - Remember the term distribution and frequency from earlier slides?
- Take document frequency into account

tf-idf weighting

- Intuition:
  - A term that is well distributed across all documents is less relevant for each individual document
  - Scale term weights such that these terms drop down in the frequency profile over all documents

tf-idf weighting

- Inverse document frequency:

$$idf_t = log\frac{N}{df_t}$$

  - *N* = number of documents in a corpus
  - df_t = number of documents in which term t occurs
  - rare terms will tend to have a high *idf*
  - frequent terms will more likely have a lower *idf*

tf-idf weighting

- Weighting terms in documents:

$$\text{tf-idf}_{t,d} = tf_{t,d} \times idf_t$$

- High: for *t* that occurs frequently in a small number of documents
- Low: for *t* that has a low frequency in the document *d*, or occurs in many documents
- Lowest: for *t* that occurs in all or most of the documents

tf-idf weighting

- Example:
  - tfidf.py and make-tfidf.py
  - generate first a df table, then test with tfidf.py on one particular document

Document Representation Models 2

Vector Space

- Mapping of terms to documents in a matrix:
  - A is a boolean (IR) or a frequency value, marking the frequency of a term *t* in document *d*

	D_1	D_2	D_3	...
t_1	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	
t_2	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	
t_3	$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	
...				

Geometry

- Vector distance
  - Euclidean distance
  - Cosine similarity
- Centroid of a set of vectors

Geometry

- Euclidean distance for two *n*-dimensional vectors:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$$

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

Geometry

- Euclidean distance example:
  - $v_1 = [\ 10, 3, 56, 3\ ]$
  - $v_2 = [\ 8, 2, 45, 1]$
- What is the distance?

Geometry

- Cosine similarity

$$similarity = cos(\theta) = \frac{A \cdot B}{\parallel A \parallel \parallel B \parallel}$$

- Dot product:

$$A \cdot B = \sum_{i=1}^{n} a_i b_i$$

- Magnitude:

$$\parallel a \parallel := \sqrt{\sum_{i=1}^{n} x_i^2}$$

Geometry

- Cosine similarity:
  - 1 means two vectors are exactly the same
  - 0 means independence
  - -1 means opposite
- Cosine similarity looks at the angle between vectors, i.e. their direction, thus is used with frequency-based vectors

Geometry

- Centroid

$$C_{ab} = \frac{1}{2}(a+b) = \left(\frac{1}{2}(a_1+b_1), \frac{1}{2}(a_2+b_2), \ldots, \frac{1}{2}(a_n+b_n)\right)$$

Geometry

- What can we now do with it?

Classification

- Class properties mapped to the centroid of document vectors that belong to the class
  - Distance metric for the decision of an unknown document either belonging to some class or not, i.e. distances between unknown document vector and class centroid

Classification

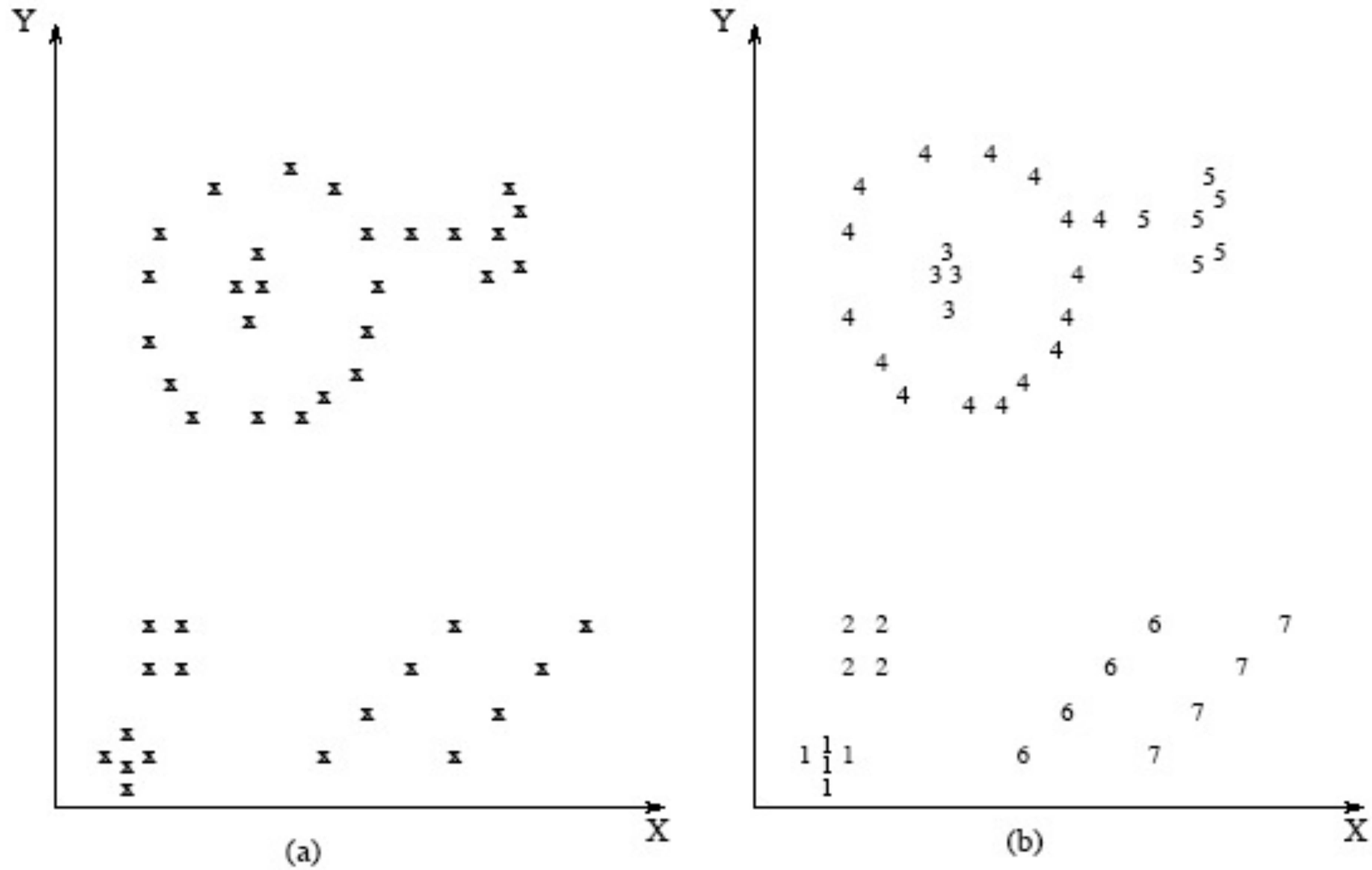
- Optimize the cluster by calculating density and identifying outliers
- Processing of probability-based vectors, frequency vectors, other properties etc.

Clustering

Clustering

- Identification of potential classes via identification of point clouds with a certain density
- Identify outliers for class centroids
- Empirically experiment with data

Clustering



Clustering

- Given a clustering criterion
  - How to find a partition into *k* groups that optimizes the criterion?
- Find all possible partitions and calculate their value of the given criterion.
- Choose the partition with the optimal value.

Clustering

- Data analysis:
  - Exploratory
    - Hypothesis creation
  - Confirmatory
    - Decision-making

Clustering

- Grouping of data:
  - Is there a correlation between data patterns?
  - Which data patterns are similar?
    - Which words are similar?
    - What kind of constructions are similar?

Clustering

- See: Robert Choate Tryon (1939) *Cluster analysis*. Edward Brothers, Ann Arbor.
  - **Cluster Analysis:** Unsupervised classification of observed groups (clusters).

Clustering

- Use:
  - No a priori hypothesis.
  - Grouping of Objects or Individuals.
  - Grouping of Variables.

Clustering

- **Clustering algorithms**
  - Vast number
  - Selection on the basis of:
    - Way in forming clusters
    - Data-structure
    - Robustness (changes, data types)

Clustering

- Further criteria
  - Data normalization
  - Choice of similarity measure
  - Data amount (small, large)
  - Use of domain knowledge or heuristics

Clustering

- Types of algorithms and techniques:
  - Hierarchical
  - Optimization
  - Density or mode-seeking
  - Clumping
  - K-means Clustering
  - Expectation Maximization (EM)

Clustering

- Formalization:
  - Feature Vector, Datum, Pattern: With d measurements: $x = (x_1, x_2, ..., x_d)$
  - $x_1, x_2, ...,$ in general: x_i is a feature or attribute of x
  - d = *dimension* of pattern or pattern space

Vector Space

- Map of features and individuals to vectors: **Feature Matrix**
  - in our case e.g. documents on rows, terms on columns (or the other way around), and fill in the frequencies x term in document

$$\mathscr{X} = \begin{bmatrix} \mathbf{x}_{1,1} & \mathbf{x}_{1,2} & \cdots & \mathbf{x}_{1,d} \\ \mathbf{x}_{2,1} & \mathbf{x}_{2,2} & \cdots & \mathbf{x}_{2,d} \\ \vdots & & & \\ \mathbf{x}_{k,1} & \mathbf{x}_{k,2} & \cdots & \mathbf{x}_{k,d} \end{bmatrix}$$

Clustering

- Formalization:
  - Pattern set: $X = \{x_1, x_2, \ldots, x_n\}$
  - The i[th] pattern in X : $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d})$
  - or

Clustering

- Class:
  - Refers to the state of nature that governs the pattern generation process.
  - Clustering techniques group patterns to classes.

Clustering

- **Hard clustering techniques:**
  - Assign a label l_i to each pattern x_i identifying its class.
  - For a set of patterns X the set of labels is $L = \{l_1, l_2, ..., l_n\}$ with $l_i \in \{1, ..., k\}$, with k the number of clusters

Clustering

- **Fuzzy clustering:**
  - Assign each pattern x_i a fractional degree of membership f_{ij} in each output cluster j.

Clustering

- **Distance measure:**
  - Specialization of a proximity measure
  - Metric on the feature space for quantifying the similarity of patterns.

Clustering

- Similarity measure:
  - For example: Euclidean Distance, Cosine Similarity, etc.

Dimension Reduction

- Using a vector space:
  - Covariance:
    - variance = average of the squared deviation of a feature from its mean
    - covariance = average of the products of the deviations of feature values from their means

Dimension Reduction

- Covariance of two features
  - Measures their tendency to vary together, i.e. co-vary.
    - Variance is the average of the squared deviation of a feature from its mean.
    - Covariance is the average of the products of the deviations of feature values from their means.

Dimension Reduction

- Covariance of two features
  - Feature *i* and Feature *j*:
    - Let $\{x_{1,i}, x_{2,i}, \ldots, x_{n,i}\}$ be a set of n examples of Feature *i*,
    - Let $\{x_{1,j}, x_{2,j}, \ldots, x_{n,j}\}$ be a corresponding set of *n* examples of Feature *j*
    - $x_{k,i}$ and $x_{k,j}$ are features of the same pattern *k*

Dimension Reduction

- Covariance of two features
  - Let m_i be the mean of Feature *i*, and m_j be the mean of Feature *j*
  - Then the covariance $c_{i,j}$ of Feature *i* and Feature *j* is:

$$\{[x_{1,i} - m_i][x_{1,j} - m_j] + \cdots + [x_{n,i} - m_i][x_{n,j} - m_j]\}/(n-1)$$

K-Means Clustering Algorithm

Clustering

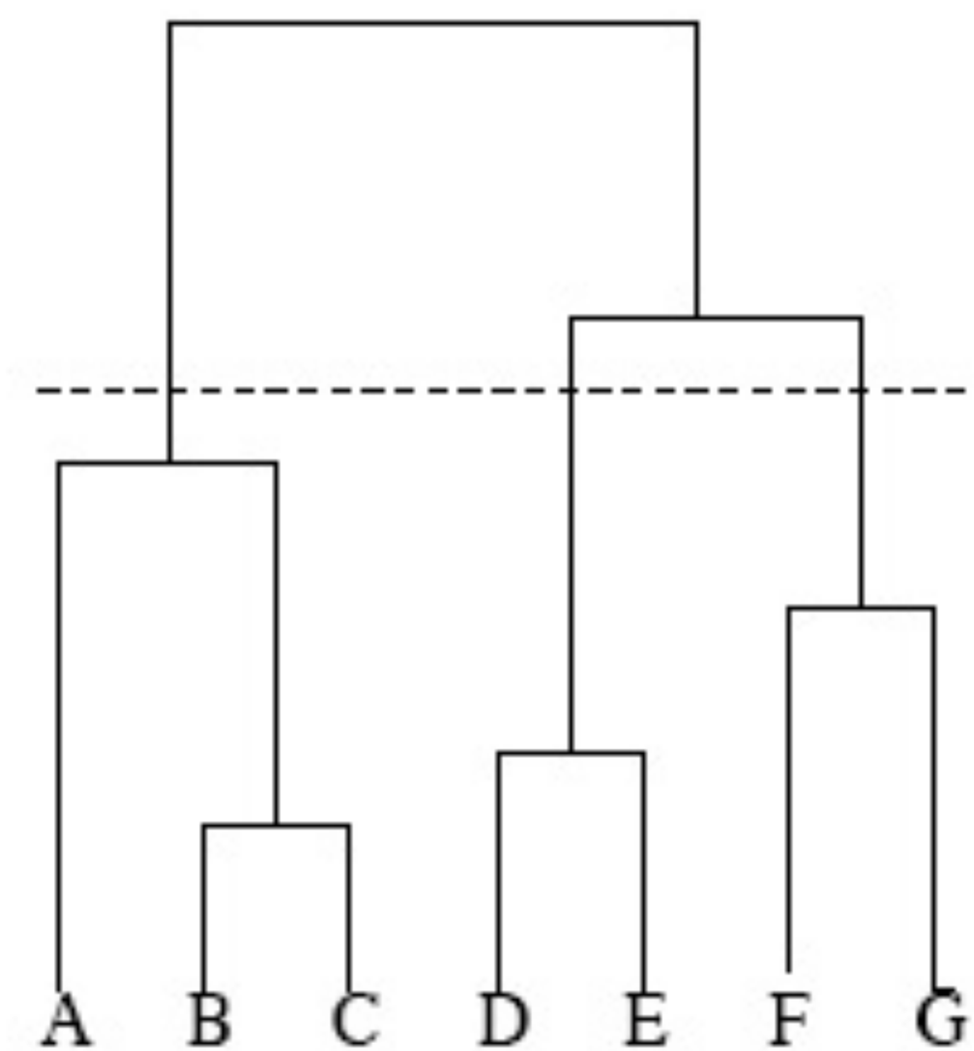
- **K-means** (Optimization Clustering): generates
  - *k* number of disjoint clusters (non-hierarchical)
  - globular clusters (spherical, elliptical, convex)
- properties:
  - numerical
  - unsupervised
  - iterative

Clustering

- **K-means**
  - *k* clusters
  - At least one element per cluster
  - No overlapping clusters
  - Not hierarchical

Clustering

- Hierarchical clusters:



Clustering

- **K-means**
  - Every member of a cluster is closer (given a metric, e.g. Euclidean Distance, Cosine Similarity) to its cluster than to any other cluster
  - Procedure

Clustering

- **K-means**
  - Initial partitioning of data set into *k* clusters
  - For each data point: calculate distance to each cluster
  - If one data point is closer to another cluster, relocate it
  - Repeat until no further relocations possible

K-Means

- Example:

Individual	Feature 1	Feature 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

K-Means

- **Initialization Step 1:** *k*=2, pick the most distant individuals and assign them each to one cluster:

	Individual	Centroid
Cluster 1	1	(1.0, 1.0)
Cluster 2	4	(5.0, 7.0)

K-Means

- **Initialization Step 2:** Assign each of the remaining vectors to its closest cluster
  - for each remaining vector:
    - calculate the distance to all centroids
    - assign it to the closest
    - recalculate the target centroid

K-Means

- Distance: for example Euclidean Distance:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

- Centroid:

$$C_{ab} = \frac{1}{2}(a+b) = \left(\frac{1}{2}(a_1+b_1), \frac{1}{2}(a_2+b_2), \ldots, \frac{1}{2}(a_n+b_n)\right)$$

K-Means

- Example: given two vectors *p* = (3, 5) and *q* = (7, 9), the Euclidean Distance is:

$$d(p, q) = \sqrt{(3-7)^2 + (5-9)^2} = \sqrt{32} \approx 5.657$$

K-Means

- Example: Cluster *x* with two vectors assigned to it, *x* = {(3, 5), (7, 9)}
  - *n* = |*x*| = 2

$$\bar{x} = \frac{\sum_{i=1}^{2} x_i}{2} = \frac{(3,5)+(7,9)}{2} = \frac{(3+7,5+9)}{2} = \left(\frac{10}{2}, \frac{14}{2}\right) = (5,7)$$

K-Means

- Initial clustering after initialization

	Cluster 1		Cluster 2	
	Individuals	Centroid	Individuals	Centroid
Step 1	1	(1.0, 1.0)	4	(5.0, 7.0)
Step 2	1, 2	(1.3, 1.5)	4	(5.0, 7.0)
Step 3	1, 2, 3	(1.8, 2.3)	4	(5.0, 7.0)
Step 4	1, 2, 3	(1.8, 2.3)	4, 5	(4.3, 6.0)
Step 5	1, 2, 3	(1.8, 2.3)	4, 5, 6	(4.3, 5.7)
Step 6	1, 2, 3	(1.8, 2.3)	4, 5, 6, 7	(4.1, 5.4)

K-Means

- Initial partitioning and clustering criterion:

	Individual	Centroid	Sum of errors
Cluster 1	1, 2, 3	(1.8, 2.3)	6.84
Cluster 2	4, 5, 6, 7	(4.1, 5.4)	5.38
total			12.22

K-Means

- Error = for every point distance to centroid
  - Criterion: the smaller the sum of square errors, the better the cluster
- Sum of all cluster errors, where the cluster error is the sum of square Euclidean Distances (for example) of each assigned vector to the centroid

K-Means

- Optimization Loop:
  - For each vector
    - Check whether it is still closer to its centroid/cluster, and not to another
    - If closer to another centroid, reassign it to it, recalculate the two centroids again
    - If there is no improvement of the error over all, STOP

K-Means

- Comparison in the Optimization Step:

Individual	Distance to C1	Distance to C2
1	1.5	5.4
2	0.4	4.3
3	2.1	1.8
4	5.7	1.8
5	3.2	0.7
6	3.8	0.8
7	2.8	1.1

K-Means

- Individual 3 should be assigned to cluster 2, instead of 1, which it is closer to, with a resulting improvement of the clustering criterion (from 12.22 to 8.53)

	Individual	Centroid	Sum of SQR errors
Cluster 1	1, 2	(1.3, 1.5)	0.63
Cluster 2	3, 4, 5, 6, 7	(3.9, 5.1)	7.9
total			8.53

K-Means

- Robust and fast
- Supervised: you need to know how many clusters you expect
- Specific cluster shapes will not be discovered

K-Means

- Initial set of *k* clusters can affect the results: Local Minima
- Not good with non-globular clusters
- Supervised, since *k* has to be predefined

Evaluation?

Clustering

- **Evaluation:**
  - Pre-clustering evaluation: Cluster tendency
  - Post-clustering evaluation: Cluster validity
    - Rather subjective
      - Valid: if clusters are not the result of an artifact or randomly chosen

Clustering

- **Evaluation: Cluster validity**
  - External assessment:
    - Compare recovered structure to some a priori structure
    - Automatically compare taxonomies, hierarchical trees, distance of centroids etc.

Clustering

- **Evaluation: Cluster validity**
  - Internal assessment:
    - Are resulting clusters intrinsically appropriate for the data.
  - Relative test:
    - Compare two resulting clusters and measure relative merit.