Similarity and clustering
Dr. Ahmed Rafea
Outline

• Motivation
• Clustering: An Overview
• Approaches
• Partitioning Approaches
• Geometric Embedding Approaches
• Web pages Clustering: An Example
Motivation

• Problem: Query word could be ambiguous:
  – Eg: Query “Star” retrieves documents about astronomy, plants, animals etc.
  – Solution: Visualisation
    • Clustering document responses to queries along lines of different topics.

• Problem 2: Manual construction of topic hierarchies and taxonomies
  – Solution:
    • Preliminary clustering of large samples of web documents.

• Problem 3: Speeding up similarity search
  – Solution:
    • Restrict the search for documents similar to a query to most representative cluster(s).
Clustering: An Overview (1/3)

- **Task**: Evolve measures of similarity to cluster a collection of documents/terms into groups within which similarity within a cluster is larger than across clusters.

- **Cluster Hypothesis**: Given a `suitable` clustering of a collection, if the user is interested in document/term \( d/t \), he is likely to be interested in other members of the cluster to which \( d/t \) belongs.

- **Similarity measures**
  - Represent documents by TFIDF vectors
  - Distance between document vectors
  - Cosine of angle between document vectors

- **Issues**
  - Large number of noisy dimensions
  - Notion of noise is application dependent
Clustering: An Overview (2/3)

- **Two important paradigms:**
  - Bottom-up agglomerative clustering
  - Top-down partitioning

- **Visualisation techniques:** Embedding of corpus in a low-dimensional space

- **Characterising the entities:**
  - *Internally:* Vector space model, probabilistic models
  - *Externally:* Measure of similarity/dissimilarity between pairs
Clustering: An Overview (3/3)

• Parameters
  - Similarity measure: (e.g.: cosine similarity) \( \rho(d_1,d_2) \)
  - Distance measure: (e.g.: Euclidian distance) \( \delta(d_1,d_2) \)
  - Number “k” of clusters

• Issues
  - Large number of noisy dimensions
  - Notion of noise is application dependent
Clustering: Approaches

• Partitioning Approaches
  – Bottom-up clustering
  – Top-down clustering

• Geometric Embedding Approaches
  – Self-organization map
  – Multidimensional scaling
  – Latent semantic indexing

• Generative models and probabilistic approaches
  – Single topic per document
  – Documents correspond to mixtures of multiple topics
Partitioning Approaches (1/5)

• Partition document collection into \( k \) clusters

• Choices: \( \{D_1, D_2, ..., D_k\} \)
  - Minimize intra-cluster distance \( \sum_i \sum_{d_1, d_2 \in D_i} \delta(d_1, d_2) \)
  - Maximize intra-cluster semblance \( \sum_i \sum_{d_1, d_2 \in D_i} \rho(d_1, d_2) \)

• If cluster representations \( D_i \) are available
  - Minimize \( \sum_{i \in D_i} \sum_{d \in D_i} \delta(d, D_i) \)
  - Maximize \( \sum_{i \in D_i} \sum_{d \in D_i} \rho(d, D_i) \)

• Soft clustering
  - \( d \) assigned to \( D_i \) with “confidence” \( z_{d,i} \)
  - Find \( z_{d,i} \) so as to minimize \( \sum_i \sum_{d \in D_i} z_{d,i} \delta(d, D_i) \) or maximize \( \sum_i \sum_{d \in D_i} z_{d,i} \rho(d, D_i) \)

• Two ways to get partitions - *bottom-up clustering* and *top-down clustering*
Partitioning Approaches (2/5)

- **Bottom-up clustering (HAC)**
  - Initially, $G$ is a collection of singleton groups, each with one document.
  - Repeat
    - Find $\Gamma, \Delta$ in $G$ with max similarity measure, $s(\Gamma \cup \Delta)$
    - Merge group $\Gamma$ with group $\Delta$
  - For each $\Gamma$ keep track of best $\Delta$
  - Use above info to plot the hierarchical merging process (DENDOGRAM)
  - To get desired number of clusters: cut across any level of the dendogram
A Dendogram presents the progressive, hierarchy-forming merging process pictorially.
Partitioning Approaches (4/5)

- **Bottom-up**
  - Requires quadratic time and space

- **Top-down or move-to-nearest**
  - Internal representation for documents as well as clusters
  - Partition documents into `k` clusters
  - 2 variants
    - “Hard” (0/1) assignment of documents to clusters
    - “soft” : documents belong to clusters, with fractional scores

  - **Termination**
    - when assignment of documents to clusters ceases to change much OR
    - When cluster centroids move negligibly over successive iterations
Partitioning Approaches (5/5)

- **Top-down clustering**
  - *Hard k-Means*: Repeat...
    - Choose \( k \) arbitrary ‘centroids’
    - Assign each document to nearest centroid
    - Recompute centroids
  - *Soft k-Means*:
    - Don’t break close ties between document assignments to clusters
    - Don’t make documents contribute to a single cluster which wins narrowly
      - Contribution for updating cluster centroid \( \mu_c \) from \( d \) document related to the current similarity between and \( \mu_c \):
        \[
        \Delta \mu_c = \eta \frac{\exp(-|d - \mu_c|^2)}{\sum_{\gamma} \exp(-|d - \mu_\gamma|^2)} (d - \mu_c)
        \]
        \[
        \mu_c = \mu_c + \Delta \mu_c
        \]
Geometric Embedding Approaches (1/2)

- **Self-Organization Map (SOM)**
  - Like soft k-means
    - Determine association between clusters and documents
    - Associate a representative vector $\mu_c$ with each cluster and iteratively refine $\mu_c$
  - Unlike k-means
    - Embed the clusters in a low-dimensional space right from the beginning
    - Large number of clusters can be initialized even if eventually many are to remain devoid of documents
    - Each cluster can be a slot in a square/hexagonal grid.
    - The grid structure defines the neighborhood $N(c)$ for each cluster $c$
    - Also involves a proximity function $h(c, \gamma)$ between clusters $\gamma$ and $c$
Geometric Embedding Approaches (2/2)

• SOM : Update Rule
  - Like Neural network
    • Data item $d$ activates neuron (closest cluster) $c_d$ as well as the neighborhood neurons $N(c_d)$
    • Eg Gaussian neighborhood function
      $$h(c, \gamma) = \exp\left(\frac{\| \mu_c - \mu_\gamma \|^2}{2\sigma^2(t)}\right)$$
    • Update rule for node $\gamma$ under the influence of $d$ is:
      $$\mu_\gamma(t+1) = \mu_\gamma(t) + \eta(t) h(\gamma, c_d)(d - \mu_\gamma)$$
    • Where $\eta(t)$ is the learning rate parameter
Web Pages Clustering: An Example (1/8)

- **Content-link Clustering**
  - The content-link hypertext clustering uses a hybrid similarity function that includes hyperlink and term components.
    - The first component, \( S_{\text{links}}^{ij} \), measures the similarity between hypertext documents \( d_i \) and \( d_j \) based on their hyperlink structures.
    - The second component, \( S_{\text{terms}}^{ij} \), measures the similarity between hypertext documents \( d_i \) and \( d_j \) based on the document terms.
  - The similarity between two hypertext documents, \( S_{\text{hybrid}}^{ij} \), is a function of \( S_{\text{links}}^{ij} \) and \( S_{\text{terms}}^{ij} \), as shown in this equation:
    \[
    S_{\text{hybrid}}^{ij} = F(S_{\text{terms}}^{ij} ; S_{\text{links}}^{ij})
    \]
Web Pages Clustering: An Example (2/8)

• A Simple Hyperlink Similarity Function
  - The measure of the hyperlink similarity between two documents, captures three important notions
    • A path between two documents,
    • The number of ancestor documents that refer to both documents in question, and
    • The number of descendant documents that both documents refer to.
Web Pages Clustering: An Example (3/8)

• Direct Paths
  - We hypothesize that the similarity between two documents varies inversely with the length of the shortest path between the two documents.
  - A link between documents $d_i$ and $d_j$ establishes a semantic relation between the two documents.
  - As the length of the shortest path between the two documents increases, the semantic relation between the two documents tends to weaken.
  - Because the hypertext links are directional, we consider both shortest path $d_i \rightarrow d_j$ and $d_j \rightarrow d_i$.
  - This Equation shows $S_{ij}^{spl}$, the component of the hyperlink similarity function that considers shortest paths between the documents:
    \[ S_{ij}^{spl} = \frac{1}{2} (spl_{ij}^{\rightarrow}) + \frac{1}{2} (spl_{ji}^{\rightarrow}) \]
Web Pages Clustering: An Example (4/8)

• Common Ancestors

- The similarity between two documents is proportional to the number of ancestors that the two documents have in common.

- As with $S^{\text{spl}}_{ij}$, the semantic relation tends to weaken as the paths between the citing articles $a_i$'s and the cited document $c_i$'s increases. This equation shows $S^{\text{anc}}_{ij}$,

$$S^{\text{anc}}_{ij} = \sum_{x \in \text{common ancestors}} \frac{1}{2(spl^{x_i} + spl^{x_j})}$$
Web Pages Clustering: An Example (5/8)

• Common Descendants
  - The similarity between two documents is also proportional to the number of descendants that the two documents have in common.
  - This Equation shows $S_{ij}^{\text{dsc}}$,

$$S_{ij}^{\text{dsc}} = \frac{1}{2(spl_{ix}^j + spl_{ix}^i)} \sum_{x \in \text{common descendants}}$$
Web Pages Clustering: An Example (6/8)

- Complete Hyperlink Similarity
  
  - The complete hyperlink similarity function between two hyperlink documents $d_i$ and $d_j$, $S_{ij}^{\text{links}}$, is a linear combination of the above components:

$$S_{ij}^{\text{links}} = W_d \cdot S_{ij}^{\text{dse}} + W_a \cdot S_{ij}^{\text{anc}} + W_s \cdot S_{ij}^{\text{spl}}$$
Web Pages Clustering: An Example (7/8)

- Term-Based Document Similarity Function
  - The weight function, in this work, used term frequency and document size factors, but did not include collection frequency.
  - Term weights also consider term attributes. The weight function assigned a larger factor to terms with attributes title, header, keyword and address than the weight factor assigned to text terms.
Web Pages Clustering: An Example (8/8)

• Term-Based Document Similarity Function

  • The total weight \( w_{ki} \) of a term \( t_i \) in document \( d_k \) is calculated based on the term similarity function as shown in the figure.

  • The weight factor \( w_{at} \) is configurable on a per server basis, but defaults to 10 for titles, 5 for headers, keywords, and addresses, and 1 for text attribute types.

  • The term-based similarity function \( S_{terms}^{ij} \) between documents \( d_i \) and \( d_j \) is the normalized dot product of the terms vectors representing each document.

    \[
    S_{terms}^{ij} = \sum_t w_{it} \cdot w_{jt}
    \]

Let

\[ t_{f_{ki}} = \text{term frequency of } t_i \text{ in } d_k \]
\[ w_{tf_{ki}} = \text{contribution to weight from frequency } t_{f_{ki}} \]
\[ w_{ds_{ki}} = \text{contribution to weight from size of } d_k \]
\[ w_{at_{ki}} = \text{contribution to weight from term attribute} \]

then

\[ w_{tf_{ki}} = (0.5 + 0.5 \frac{t_{f_{ki}}}{\max_j \{t_{f_{kj}}\}}) \] \hspace{1cm} (6)
\[ w_{ds_{ki}} = \frac{1}{\sqrt{\sum_i (w_{at_{ki}} w_{tf_{ki}})^2}} \] \hspace{1cm} (7)
\[ w_{ki} = w_{tf_{ki}} \cdot w_{ds_{ki}} \cdot w_{at_{ki}} \] \hspace{1cm} (8)