CS4051

Information Retrieval

Week 13

Muhammad Rafi

April 29, 2024

Text Clustering

Chapter 16 & 17

Agenda

- What is Clustering
- Classification vs. Clustering
- Clustering Hypothesis
- Hard Vs. Soft Clustering
- Flat Vs. Hierarchical Clustering
- Flat Clustering
- Partition Clustering
  - K- Means
  - Example
  - Pros and Cons

Agenda

- Hierarchical Clustering
  - Hierarchical Agglomerative Clustering (HAC)
  - Example
  - Pros and Cons
- Evaluation of Clustering Results
- Hierarchical Clustering
- Algorithm for HAC
- Variations to Clustering Algorithm
- Conclusion

Clustering

- It is an unsupervised, machine learning technique that automatically separate the similar objects in a heterogeneous collection.
- Autonomously learn the pattern of this similarity is a vital task, the features of the object implicitly identify for the clustering.
- An objective function is used to optimize the similarity between the objects in a cluster, and reduce the similarity between different clusters.

Clustering

- It is a challenging problem, as we will be doing dual optimization: High Intra-cluster similarity and Low Inter-cluster similarity
- Document clustering is a special problem, where objects to clusters are documents.
  - Natural Language Text
  - Knowing exact number of distinct groups(clusters)
  - Producing an optimal clustering arrangement
  - Label the groups
  - Evaluation of clustering results

Classification Vs. Clustering

- Classification: supervised learning
- Clustering: unsupervised learning
  - Classification: Classes are human-defined and part of the input to the learning algorithm.
  - Clustering: Clusters are inferred from the data without human input.
    - However, there are many ways of influencing the outcome of clustering: number of clusters, similarity measure, representation of documents, . . .

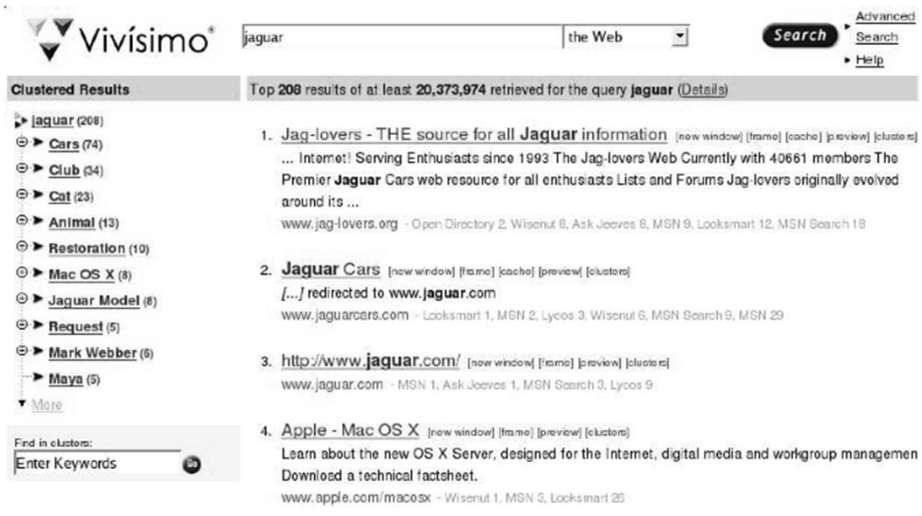
Clustering Hypothesis

- **Cluster hypothesis.** Documents in the same cluster behave similarly with respect to relevance to information needs.
- All applications of clustering in IR are based (directly or indirectly) on the cluster hypothesis.
- Van Rijsbergen's original wording: "closely associated documents tend to be relevant to the same requests".

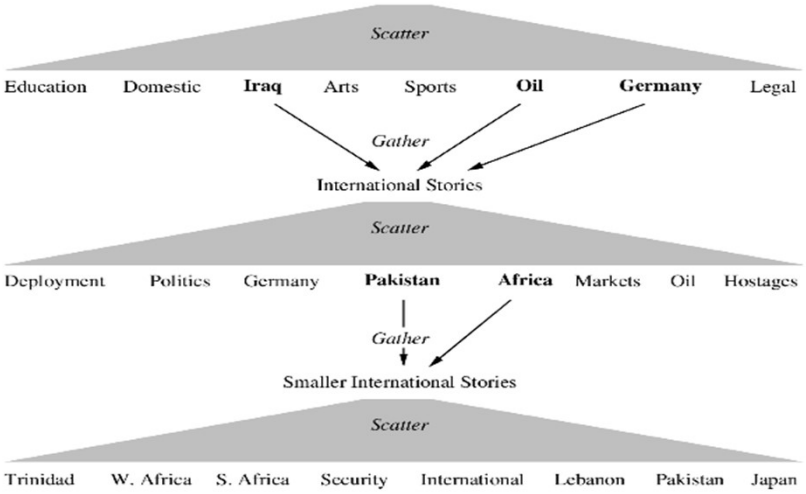
Clustering Applications

Application	What is clustered?	Benefit
Search result clustering	search results	more effective information presentation to user
Scatter-Gather	(subsets of) collection	alternative user interface: “search without typing”
Collection clustering	collection	effective information presentation for exploratory browsing
Cluster-based retrieval	collection	higher efficiency: faster search

Vivisimo –sense clustering



Scatter –Gather



Flat vs. Hierarchical Clustering

- Flat algorithms
  - Usually start with a random (partial) partitioning of docs into groups
  - Refine iteratively
  - Main algorithm: *K*-means
- Hierarchical algorithms
  - Create a hierarchy, all possible cluster level
  - Bottom-up, agglomerative
  - Top-down, divisive
  - Main algorithm: HAC

Hard vs. Soft Clustering

- Hard Clustering
  - Each document belongs to exactly one cluster.
- Soft Clustering
  - A document can belong to more than one cluster.
  - Fuzzy memebership in more then one cluster.

Flat Clustering

- Flat algorithms compute a partition of N documents into a set of K clusters.
- Given: a set of documents and the number K
- Find: a partition into K clusters that optimizes the chosen partitioning criterion
- Global optimization: exhaustively enumerate partitions, pick optimal one an-NP Hard
- Effective heuristic method: K-means algorithm

k-Mean Clustering

- k-Mean clustering, perhaps the best known clustering algorithm. It is simple, works well in many cases
- It is used as default / baseline for clustering documents.
- There are three main steps in doc. clustering
  - Representation of Documents
  - Similarity measure
  - Clustering approach

Similarity Functions or Measure

- A similarity function maps two objects into a real value between (0-1).
- Identical objects get a value 1, while totally different pairs get a lower value.

Similarity Functions or Measure

- Equal self-similarity. d(A, A) = d(B, B) for all points A and B. Therefore, s(A, A) = s(B, B) for all stimuli A and B.
- Minimality. d(A, B) > d(A, A) for all points A ≠ B. Therefore, s(A, B) < s(A, A) for all stimuli A ≠ B.
- Symmetry. d(A, B) = d(B, A) for all points A and B. Therefore, s(A, B) = s(B, A) for all stimuli A and B.
- Triangle Inequality. d(A, B) + d(B, C) ≥ d(A, C) for all points A, B, and C.

k-Mean Doc. Clustering

- It uses vector space model to represents document in feature dimensional space.
- It uses Euclidean distance to measure closeness(similarity) among documents.
- Start with an initial k-documents, for each other documents compute the distance from these k-documents.
- The smallest distance document from the k-document, is assigned to the cluster.

k-Mean

K-MEANS$(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$
1 $(\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_K) \leftarrow$ SELECTRANDOMSEEDS$(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$
2 **for** $k \leftarrow 1$ **to** K
3 **do** $\vec{\mu}_k \leftarrow \vec{s}_k$
4 **while** stopping criterion has not been met
5 **do for** $k \leftarrow 1$ **to** K
6 **do** $\omega_k \leftarrow \{\}$
7 **for** $n \leftarrow 1$ **to** N
8 **do** $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$
9 $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*
10 **for** $k \leftarrow 1$ **to** K
11 **do** $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$ *(recomputation of centroids)*
12 **return** $\{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$

Convergence of K-Means

First, there are at most k^N ways to partition N data points into k

clusters; each such partition can be called a "clustering". This is a large but finite number. For each iteration of the algorithm, we produce a new clustering based *only* on the old clustering. Notice that

1. if the old clustering is the same as the new, then the next clustering will again be the same.
2. If the new clustering is different from the old then the newer one has a lower cost

Since the algorithm iterates a function whose domain is a finite set, the iteration must eventually enter a cycle. The cycle cannot have length greater than 1, because otherwise by (2) you would have some clustering which has a lower cost than itself which is impossible. Hence the cycle must have length exactly 1. Hence k-means converges in a finite number of iterations.

k-Mean Clustering

- *K*-means algorithm is a simple yet popular method for clustering analysis
- Its performance is determined by initialisation and appropriate distance measure
- There are several variants of *K*-means to overcome its weaknesses
  - *K*-Medoids: resistance to noise and/or outliers
  - *K*-Modes: extension to categorical data clustering analysis
  - CLARA: extension to deal with large data sets
  - Mixture models (EM algorithm): handling uncertainty of clusters

Drawbacks of K-Mean

- Sensitive to the initial seeds
- Difficult to compare the results
- Fixed number of clusters, difficult to predict the actual numbers from the dataset.

Evaluating Clustering Task

- The task is very challenging and hence its evaluation is also very challenging
- The evaluation
  - Internal
    - Only data is used to evaluate the clustering results
  - External
    - Evaluation with gold standard

What is a Good Clustering?

- Internal criterion: A good clustering will produce high quality clusters in which:
  - the intra-class (that is, intra-cluster) similarity is high
  - the inter-class similarity is low
  - The measured quality of a clustering depends on both the document representation and the similarity measure used

External criteria for clustering quality

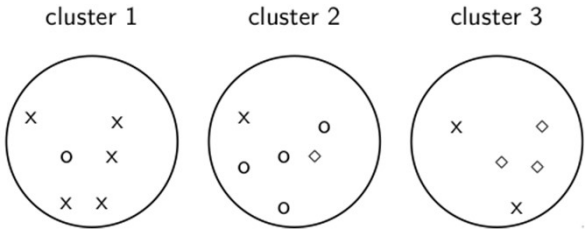
- Based on a gold standard data set, e.g., the Reuters collection we also used for the evaluation of classification
- Goal: Clustering should reproduce the classes in the gold standard
- (But we only want to reproduce how documents are divided into groups, not the class labels.)
- First measure for how well we were able to reproduce the classes: Purity

External criterion: Purity

- $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of clusters and $C = \{c_1, c_2, \ldots, c_J\}$ is the set of classes.
- For each cluster ω_k : find class c_j with most members n_{kj} in ω_k
- Sum all n_{kj} and divide by total number of points

$$\text{purity}(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

Example: Purity



- To compute purity:
  - 5 = $\max_j |\omega_1 \cap c_j|$ (class x, cluster 1);
  - 4 = $\max_j |\omega_2 \cap c_j|$ (class o, cluster 2); and
  - 3 = $\max_j |\omega_3 \cap c_j|$ (class ◇, cluster 3).
- Purity is $(1/17) \times (5 + 4 + 3) \approx 0.71$.

Random Index

- An alternative to this information-theoretic interpretation of clustering is to view it as a series of decisions, one for each of the N(N-1)/2 pairs of documents in the collection.
- We want to assign two documents to the same cluster if and only if they are similar.
- A true positive (TP) decision assigns two similar documents to the same cluster, a true negative (TN) decision assigns two dissimilar documents to different clusters.
- There are two types of errors we can commit. A (FP) decision assigns two dissimilar documents to the same cluster. A (FN) decision assigns two similar documents to different clusters.

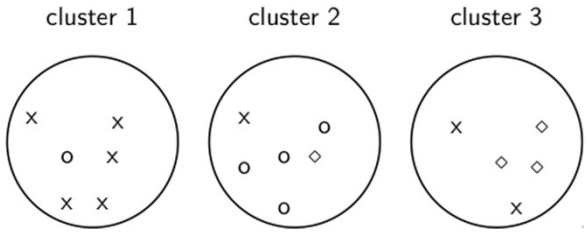
Random Index

- Definition: $\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}$
- Based on 2x2 contingency table of all pairs of documents:

	same cluster	different clusters
same class	true positives (TP)	false negatives (FN)
different classes	false positives (FP)	true negatives (TN)

- TP+FN+FP+TN is the total number of pairs.
- There are $\binom{N}{2}$ pairs for N documents.
- Example: $\binom{17}{2}$ = 136 in o/◇/x example
- Each pair is either positive or negative (the clustering puts the two documents in the same or in different clusters) . . .
- . . . and either "true" (correct) or "false" (incorrect): the clustering decision is correct or incorrect.

Example: Random Index



- Consider the same example once again, this time we need to calculate Random Index (RI)

Example: Random Index

As an example, we compute RI for the o/⋄/x example. We first compute TP + FP. The three clusters contain 6, 6, and 5 points, respectively, so the total number of "positives" or pairs of documents that are in the same cluster is:

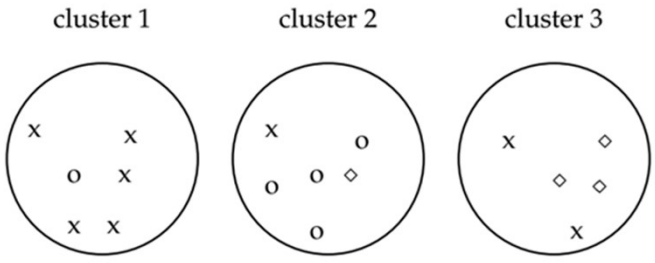
$$\mathrm{TP} + \mathrm{FP} = \binom{6}{2} + \binom{6}{2} + \binom{5}{2} = 40$$

Of these, the x pairs in cluster 1, the o pairs in cluster 2, the ⋄ pairs in cluster 3, and the x pair in cluster 3 are true positives:

$$\mathrm{TP} = \binom{5}{2} + \binom{4}{2} + \binom{3}{2} + \binom{2}{2} = 20$$

Thus, FP = 40 − 20 = 20. FN and TN are computed similarly.

Example: Random Index



To solve this problem, you need to consider this matrix:

TP: Same class + same cluster	FN: Same class + different clusters
FP: different class + same cluster	TN: different class + different clusters

Example: Random Index

Same class and in the same cluster

$$TP = \overset{X1}{\binom{5}{2}} + \overset{O2}{\binom{4}{2}} + \overset{X3}{\binom{2}{2}} + \overset{\nabla 2}{\binom{3}{2}} = 10 + 6 + 1 + 3 = 20$$

Same classes, but different clusters

$$FN = \overset{X1}{\binom{5}{1}}\overset{X2}{\binom{1}{1}} + \overset{X1}{\binom{5}{1}}\overset{X3}{\binom{2}{1}} + \overset{O1}{\binom{1}{1}}\overset{O2}{\binom{4}{1}} + \overset{X2}{\binom{1}{1}}\overset{X3}{\binom{2}{1}} + \overset{\nabla 2}{\binom{1}{1}}\overset{\nabla 3}{\binom{3}{1}} = 5 + 10 + 4 + 2 + 3 = 24$$

Different classes but in the same cluster

$$FP = \overset{X1}{\binom{5}{1}}\overset{O1}{\binom{1}{1}} + \overset{O2}{\binom{4}{1}}\overset{X2}{\binom{1}{1}} + \overset{O2}{\binom{4}{1}}\overset{\nabla 2}{\binom{1}{1}} + \overset{X2}{\binom{1}{1}}\overset{\nabla 2}{\binom{1}{1}} + \overset{\nabla 3}{\binom{3}{1}}\overset{X3}{\binom{2}{1}} = 5 + 4 + 4 + 1 + 6 = 20$$

Example: Random Index

	same cluster	different clusters
same class	TP = 20	FN = 24
different classes	FP = 20	TN = 72

RI is then

(20 + 72)/(20 + 20 + 24 + 72) ≈ 0.68.

NMI and F-Measure

- Normalized mutual information (NMI)
  - How much information does the clustering contain about the classification?
  - Singleton clusters (number of clusters = number of docs) have maximum MI
  - Therefore: normalize by entropy of clusters and classes
- F measure
  - Like Rand, but “precision” and “recall” can be weighted

Hierarchical Agglomerative Clustering

- Assumes a similarity function for determining the similarity of two instances.
- Starts with all instances in a separate cluster and then repeatedly joins the two clusters that are most similar until there is only one cluster.
- The history of merging forms a binary tree or hierarchy.

Aglommerative vs. Divisive Clustering

- Aglommerative (bottom-up) methods start with each example in its own cluster and iteratively combine them to form larger and larger clusters.
- Divisive (partitional, top-down) separate all examples immediately into clusters.

Example

```
SIMPLEHAC(d_1, ..., d_N)
 1  for n ← 1 to N
 2  do for i ← 1 to N
 3     do C[n][i] ← SIM(d_n, d_i)
 4     I[n] ← 1  (keeps track of active clusters)
 5  A ← []  (collects clustering as a sequence of merges)
 6  for k ← 1 to N − 1
 7  do ⟨i, m⟩ ← arg max_{⟨i,m⟩: i≠m ∧ I[i]=1 ∧ I[m]=1} C[i][m]
 8     A.APPEND(⟨i, m⟩)  (store merge)
 9     for j ← 1 to N
10     do C[i][j] ← SIM(i, m, j)
11        C[j][i] ← SIM(i, m, j)
12     I[m] ← 0  (deactivate cluster)
13  return A
```

Figure 17.2 A simple, but inefficient HAC algorithm.

Variations of HAC

- **Single-link clustering** (also called the connectedness, the minimum method or the nearest neighbor method) — methods that consider the distance between two clusters to be equal to the shortest distance from any member of one cluster to any member of the other cluster. If the data consist of similarities, the similarity between a pair of clusters is considered to be equal to the greatest similarity from any member of one cluster to any member of the other cluster (Sneath and Sokal, 1973).
- **Complete-link clustering** (also called the diameter, the maximum method or the furthest neighbor method) - methods that consider the distance between two clusters to be equal to the longest distance from any member of one cluster to any member of the other cluster (King, 1967).
- **Average-link clustering** (also called minimum variance method) - methods that consider the distance between two clusters to be equal to the average distance from any member of one cluster to any member of the other cluster. Such clustering algorithms may be found in (Ward, 1963) and (Murtagh, 1984).

Variations of HAC

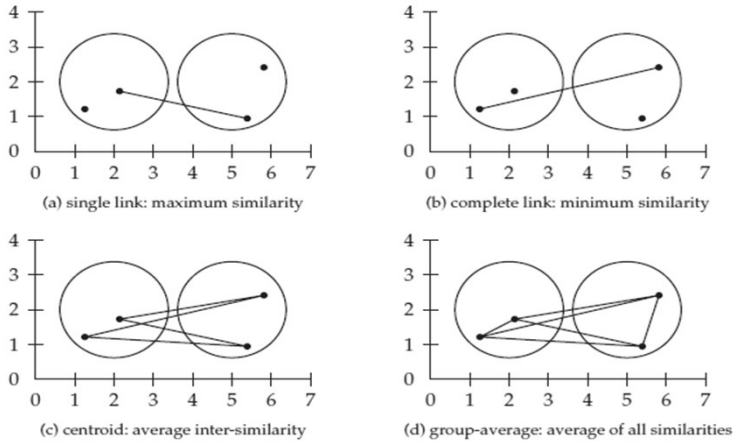


Figure 17.3 The different notions of cluster similarity used by the four HAC algorithms. An *inter-similarity* is a similarity between two documents from different clusters.

Computational Complexity

- In the first iteration, all HAC methods need to compute similarity of all pairs of n individual instances which is $O(n^2)$.
- In each of the subsequent $n-2$ merging iterations, compute the distance between the most recently created cluster and all other existing clusters.
- In order to maintain an overall $O(n^2)$ performance, computing similarity to each cluster must be done in constant time.
  - Else $O(n^2 \log n)$ or $O(n^3)$ if done naively