

Introduction to
Information Retrieval

Clustering

Chris Manning, Pandu Nayak, and Prabhakar Raghavan

Introduction to Information Retrieval

Today's Topic: Clustering

- Document clustering
 - Motivations
 - Document representations
 - Success criteria
- Clustering algorithms
 - Partitional
 - Hierarchical

Introduction to Information Retrieval Ch. 16

What is clustering?

- Clustering**: the process of grouping a set of objects into classes of similar objects
 - Documents within a cluster should be similar.
 - Documents from different clusters should be dissimilar.
- The commonest form of *unsupervised learning*
 - Unsupervised learning = learning from raw data, as opposed to supervised data where a classification of examples is given
 - A common and important task that finds many applications in IR and other places

Introduction to Information Retrieval Ch. 16

A data set with clear cluster structure

- How would you design an algorithm for finding the three clusters in this case?

Introduction to Information Retrieval Sec. 16.1

Applications of clustering in IR

- Whole corpus analysis/navigation**
 - Better user interface: search without typing**
- For improving recall in search applications
 - Better search results (like pseudo RF)
- For better navigation of search results
 - Effective "user recall" will be higher
- For speeding up vector space retrieval
 - Cluster-based retrieval gives faster search

Introduction to Information Retrieval

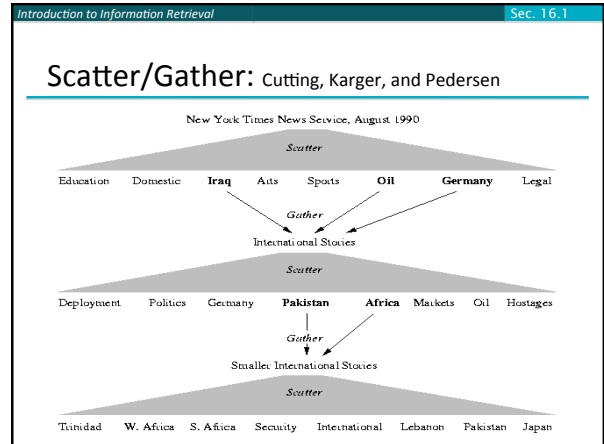
Yahoo! Hierarchy *isn't* clustering but *is* the kind of output you want from clustering

www.yahoo.com/Science

Introduction to Information Retrieval

Google News: automatic clustering gives an effective news presentation metaphor

The screenshot shows the Google News homepage with several news items. The items are organized into clusters, with related stories grouped together. For example, there are clusters for 'Pakistan protests over US missile strikes', 'Drop That Cigarette, Today is The Great American Cigarette Day', and 'Parino: Bush would sign jobs bill benefits retention'.



Introduction to Information Retrieval

Sec. 16.1

Applications of clustering in IR

- Whole corpus analysis/navigation
 - Better user interface: search without typing
- For improving recall in search applications
 - Better search results (like pseudo RF)
- For better navigation of search results
 - Effective "user recall" will be higher
- For speeding up vector space retrieval
 - Cluster-based retrieval gives faster search

Introduction to Information Retrieval

Sec. 16.1

For improving search recall

- Cluster hypothesis** - Documents in the same cluster behave similarly with respect to relevance to information needs
- Therefore, to improve search recall:
 - Cluster docs in corpus a priori
 - When a query matches a doc D , also return other docs in the cluster containing D
- Hope if we do this: The query "car" will also return docs containing *automobile*
 - Because clustering grouped together docs containing *car* with those containing *automobile*.

Why might this happen?

Introduction to Information Retrieval

Sec. 16.1

Applications of clustering in IR

- Whole corpus analysis/navigation
 - Better user interface: search without typing
- For improving recall in search applications
 - Better search results (like pseudo RF)
- For better navigation of search results
 - Effective "user recall" will be higher
- For speeding up vector space retrieval
 - Cluster-based retrieval gives faster search

Introduction to Information Retrieval

The screenshot shows a search engine results page for the query "clustering". The results are grouped into clusters. The first cluster is titled "Clustering" and includes links to "Lower Latency In Your Data Center with Intel's Cluster Ready Solution", "Load Balancing 101", "Search, Engine", "Hierarchical", "Definition", and "High availability". The second cluster is titled "Computer cluster - Wikipedia, the free encyclopedia" and includes links to "Middleware such as MPI (Message Passing Interface) or PVM (Parallel Virtual Machine) permits computer clustering programs to be portable to a computer cluster", "Writer's Web: Frowning Clustering", and "Printing: Clustering Mikala Devotion & Joe Essel (printable version here) Clustering is a type of printing that allows you to explore many different writing techniques and techniques of clustering".

yippy.com - grouping search results 12

Applications of clustering in IR

- Whole corpus analysis/navigation
 - Better user interface: search without typing
- For improving recall in search applications
 - Better search results (like pseudo RF)
- For better navigation of search results
 - Effective “user recall” will be higher
- **For speeding up vector space retrieval**
 - **Cluster-based retrieval gives faster search**

Issues for clustering

- Representation for clustering
 - Document representation
 - Vector space? Normalization?
 - Need a notion of similarity/distance
- How many clusters?
 - Fixed a priori?
 - Completely data driven?
 - Avoid “trivial” clusters - too large or small
 - If a cluster's too large, then for navigation purposes you've wasted an extra user click without whittling down the set of documents much.

Notion of similarity/distance

- Ideal: semantic similarity.
- Practical: term-statistical similarity (docs as vectors)
 - Cosine similarity
 - For many algorithms, easier to think in terms of a *distance* (rather than *similarity*) between docs.
 - We will mostly speak of Euclidean distance
 - But real implementations use cosine similarity

Hard vs. soft clustering

- Hard clustering: Each document belongs to exactly one cluster
 - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.
 - Makes more sense for applications like creating browsable hierarchies
 - You may want to put a pair of sneakers in two clusters: (i) sports apparel and (ii) shoes
 - You can only do that with a soft clustering approach.
- We won't do soft clustering today. See IIR 16.5, 18

Clustering Algorithms

- Flat algorithms
 - Usually start with a random (partial) partitioning
 - Refine it iteratively
 - K means clustering
 - (Model based clustering)
- Hierarchical algorithms
 - Bottom-up, agglomerative
 - (Top-down, divisive)

Partitioning Algorithms

- Partitioning method: Construct a partition of n documents into a set of K clusters
- Given: a set of documents and the number K
- Find: a partition of K clusters that optimizes the chosen partitioning criterion
 - Globally optimal
 - Intractable for many objective functions
 - Ergo, exhaustively enumerate all partitions
 - Effective heuristic methods: K -means and K -medoids algorithms

See also Kleinberg NIPS 2002 – impossibility for natural clustering

Introduction to Information Retrieval Sec. 16.4

K-Means

- Assumes documents are real-valued vectors.
- Clusters based on *centroids* (aka the *center of gravity* or mean) of points in a cluster, c :

$$\bar{\mu}(c) = \frac{1}{|c|} \sum_{x \in c} \bar{x}$$
- Reassignment of instances to clusters is based on distance to the current cluster centroids.
 - (Or one can equivalently phrase it in terms of similarities)

Introduction to Information Retrieval Sec. 16.4

K-Means Algorithm

Select K random docs $\{s_1, s_2, \dots, s_k\}$ as seeds.

Until clustering *converges* (or other stopping criterion):

For each doc d_i :

Assign d_i to the cluster c_j such that $dist(x_i, s_j)$ is minimal.

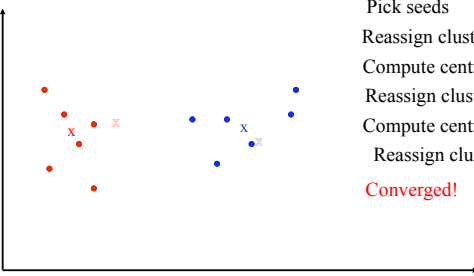
(Next, update the seeds to the centroid of each cluster)

For each cluster c_j

$s_j = \mu(c_j)$

Introduction to Information Retrieval Sec. 16.4

K Means Example (K=2)



Pick seeds
 Reassign clusters
 Compute centroids
 Reassign clusters
 Compute centroids
 Reassign clusters
Converged!

Introduction to Information Retrieval Sec. 16.4

Termination conditions

- Several possibilities, e.g.,
 - A fixed number of iterations.
 - Doc partition unchanged.
 - Centroid positions don't change.

Does this mean that the docs in a cluster are unchanged?

Introduction to Information Retrieval Sec. 16.4

Convergence

- Why should the K -means algorithm ever reach a *fixed point*?
 - A state in which clusters don't change.
- K -means is a special case of a general procedure known as the *Expectation Maximization (EM) algorithm*.
 - EM is known to converge.
 - Number of iterations could be large.
 - But in practice usually isn't

Introduction to Information Retrieval Sec. 16.4

Convergence of K-Means

- Residual Sum of Squares (RSS), a goodness measure of a cluster, is the sum of squared distances from the cluster centroid:
 - $RSS_j = \sum_i |d_i - c_j|^2$ (sum over all d_i in cluster j)
- $RSS = \sum_j RSS_j$
- Reassignment monotonically decreases RSS since each vector is assigned to the closest centroid.
- Recomputation also monotonically decreases each RSS_j because ...

Cluster recomputation in K-means

- $RSS_j = \sum_i |d_i - c_j|^2 = \sum_i \sum_k (d_{ik} - c_{jk})^2$
 - i ranges over documents in cluster j
- RSS_j reaches minimum when:
 - $\sum_i -2(d_{ik} - c_{jk}) = 0$ (for each c_{jk})
 - $\sum_i c_{jk} = \sum_i d_{ik}$
 - $m_j c_{jk} = \sum_i d_{ik}$ (m_j is # of docs in cluster j)
 - $c_{jk} = (1/m_j) \sum_i d_{ik}$
- K-means typically converges quickly

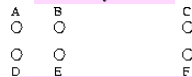
Time Complexity

- Computing distance between two docs is $O(M)$ where M is the dimensionality of the vectors.
- Reassigning clusters: $O(KN)$ distance computations, or $O(KNM)$.
- Computing centroids: Each doc gets added once to some centroid: $O(NM)$.
- Assume these two steps are each done once for l iterations: $O(lKNM)$.

Seed Choice

- Results can vary based on random seed selection.
- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings.
 - Select good seeds using a heuristic (e.g., doc least similar to any existing mean)
 - Try out multiple starting points
 - Initialize with the results of another method.

Example showing sensitivity to seeds



In the above, if you start with B and E as centroids you converge to {A,B,C} and {D,E,F}
If you start with D and F you converge to {A,B,D,E} {C,F}

K-means issues, variations, etc.

- Recomputing the centroid after every assignment (rather than after all points are re-assigned) can improve speed of convergence of K-means
- Assumes clusters are spherical in vector space
 - Sensitive to coordinate changes, weighting etc.
- Disjoint and exhaustive
 - Doesn't have a notion of "outliers" by default
 - But can add outlier filtering

Dhillon et al. ICDM 2002 - variation to fix some issues with small document clusters

How Many Clusters?

- Number of clusters K is given
 - Partition n docs into predetermined number of clusters
- Finding the "right" number of clusters is part of the problem
 - Given docs, partition into an "appropriate" number of subsets.
 - E.g., for query results - ideal value of K not known up front - though UI may impose limits.

K not specified in advance

- Say, the results of a query.
- Solve an optimization problem: penalize having lots of clusters
 - application dependent, e.g., compressed summary of search results list.
- Tradeoff between having more clusters (better focus within each cluster) and having too many clusters

K not specified in advance

- Given a clustering, define the Benefit for a doc to be the cosine similarity to its centroid
- Define the Total Benefit to be the sum of the individual doc Benefits.

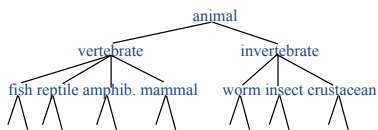
Why is there always a clustering of Total Benefit n ?

Penalize lots of clusters

- For each cluster, we have a Cost C .
- Thus for a clustering with K clusters, the Total Cost is KC .
- Define the Value of a clustering to be = $\text{Total Benefit} - \text{Total Cost}$.
- Find the clustering of highest value, over all choices of K .
 - Total benefit increases with increasing K . But can stop when it doesn't increase by "much". The Cost term enforces this.

Hierarchical Clustering

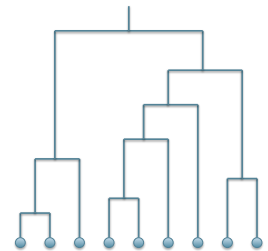
- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of documents.



- One approach: recursive application of a partitioning clustering algorithm.

Dendrogram: Hierarchical Clustering

- Clustering obtained by cutting the dendrogram at a desired level: each **connected component** forms a cluster.



34

Hierarchical Agglomerative Clustering (HAC)

- Starts with each doc in a separate cluster
 - then repeatedly joins the closest pair of clusters, until there is only one cluster.
- The history of merging forms a binary tree or hierarchy.

Note: the resulting clusters are still "hard" and induce a partition

Closest pair of clusters

- Many variants to defining closest pair of clusters
- Single-link**
 - Similarity of the *most* cosine-similar (single-link)
- Complete-link**
 - Similarity of the "furthest" points, the *least* cosine-similar
- Centroid**
 - Clusters whose centroids (centers of gravity) are the most cosine-similar
- Average-link**
 - Average cosine between all pairs of elements

Introduction to Information Retrieval Sec. 17.2

Single Link Agglomerative Clustering

- Use maximum similarity of pairs:

$$sim(c_i, c_j) = \max_{x \in c_i, y \in c_j} sim(x, y)$$

- Can result in “straggly” (long and thin) clusters due to chaining effect.
- After merging c_i and c_j , the similarity of the resulting cluster to another cluster, c_k , is:

$$sim((c_i \cup c_j), c_k) = \max(sim(c_i, c_k), sim(c_j, c_k))$$

Introduction to Information Retrieval Sec. 17.2

Single Link Example

Introduction to Information Retrieval Sec. 17.2

Complete Link

- Use minimum similarity of pairs:

$$sim(c_i, c_j) = \min_{x \in c_i, y \in c_j} sim(x, y)$$

- Makes “tighter,” spherical clusters that are typically preferable.
- After merging c_i and c_j , the similarity of the resulting cluster to another cluster, c_k , is:

$$sim((c_i \cup c_j), c_k) = \min(sim(c_i, c_k), sim(c_j, c_k))$$

Introduction to Information Retrieval Sec. 17.2

Complete Link Example

Introduction to Information Retrieval

General HAC algorithm and complexity

- Compute similarity between all pairs of documents $O(N^2)$
- Do $N - 1$ times
 - Find closest pair of documents/clusters to merge
 - Naïve: $O(N^2)$
 - Priority Queue: $O(N)$
 - Single link: $O(N)$
 - Update similarity of all documents/clusters to new cluster
 - Naïve: $O(N)$
 - Priority Queue: $O(N \log N)$
 - Single link: $O(N)$

Best merge persistent!

Introduction to Information Retrieval Sec. 17.3

Group Average

- Similarity of two clusters = average similarity of all pairs within merged cluster.

$$sim(c_i, c_j) = \frac{1}{|c_i \cup c_j|(|c_i \cup c_j| - 1)} \sum_{\bar{x} \in (c_i \cup c_j)} \sum_{\bar{y} \in (c_i \cup c_j), \bar{y} \neq \bar{x}} sim(\bar{x}, \bar{y})$$

- Compromise between single and complete link.
- Two options:
 - Averaged across all ordered pairs in the merged cluster
 - Averaged over all pairs *between* the two original clusters
- No clear difference in efficacy

Introduction to Information Retrieval Sec. 17.3

Computing Group Average Similarity

- Always maintain sum of vectors in each cluster.

$$\bar{s}(c_j) = \sum_{x \in c_j} \bar{x}$$

- Compute similarity of clusters in constant time:

$$\text{sim}(c_i, c_j) = \frac{(\bar{s}(c_i) + \bar{s}(c_j)) \cdot (\bar{s}(c_i) + \bar{s}(c_j)) - (|c_i| + |c_j|)}{(|c_i| + |c_j|)(|c_i| + |c_j| - 1)}$$

Introduction to Information Retrieval Sec. 16.3

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

Introduction to Information Retrieval Sec. 16.3

External criteria for clustering quality

- Quality measured by its ability to discover some or all of the hidden patterns or latent classes in gold standard data
- Assesses a clustering with respect to ground truth ... requires *labeled data*
- Assume documents with C gold standard classes, while our clustering algorithms produce K clusters, $\omega_1, \omega_2, \dots, \omega_K$ with n_i members.

Introduction to Information Retrieval Sec. 16.3

External Evaluation of Cluster Quality

- Simple measure: purity, the ratio between the dominant class in the cluster ω_i and the size of cluster ω_i

$$\text{Purity}(\omega_i) = \frac{1}{n_i} \max_j (n_{ij}) \quad j \in C$$
- Biased because having n clusters maximizes purity
- Others are entropy of classes in clusters (or mutual information between classes and clusters)

Introduction to Information Retrieval Sec. 16.3

Purity example

Cluster I Cluster II Cluster III

Cluster I: Purity = $1/6 (\max(5, 1, 0)) = 5/6$

Cluster II: Purity = $1/6 (\max(1, 4, 1)) = 4/6$

Cluster III: Purity = $1/5 (\max(2, 0, 3)) = 3/5$

Introduction to Information Retrieval Sec. 16.3

Rand Index measures between pair decisions. Here RI = 0.68

Number of point pairs	Same Cluster in clustering	Different Clusters in clustering
Same class in ground truth	20	24
Different classes in ground truth	20	72

Note: A hand-drawn oval encircles the 20 and 72 values in the table.

Rand index and Cluster F-measure

$$RI = \frac{A + D}{A + B + C + D}$$

Compare with standard Precision and Recall:

$$P = \frac{A}{A + B} \quad R = \frac{A}{A + C}$$

People also define and use a cluster F-measure, which is probably a better measure.

Final word and resources

- In clustering, clusters are inferred from the data without human input (unsupervised learning)
- However, in practice, it's a bit less clear: there are many ways of influencing the outcome of clustering: number of clusters, similarity measure, representation of documents, . . .
- Resources
 - IIR 16 except 16.5
 - IIR 17.1–17.3