

PROJECT 2019

**CLUSTERING DOCUMENTS TO COMPRESS
INVERTED INDEX**

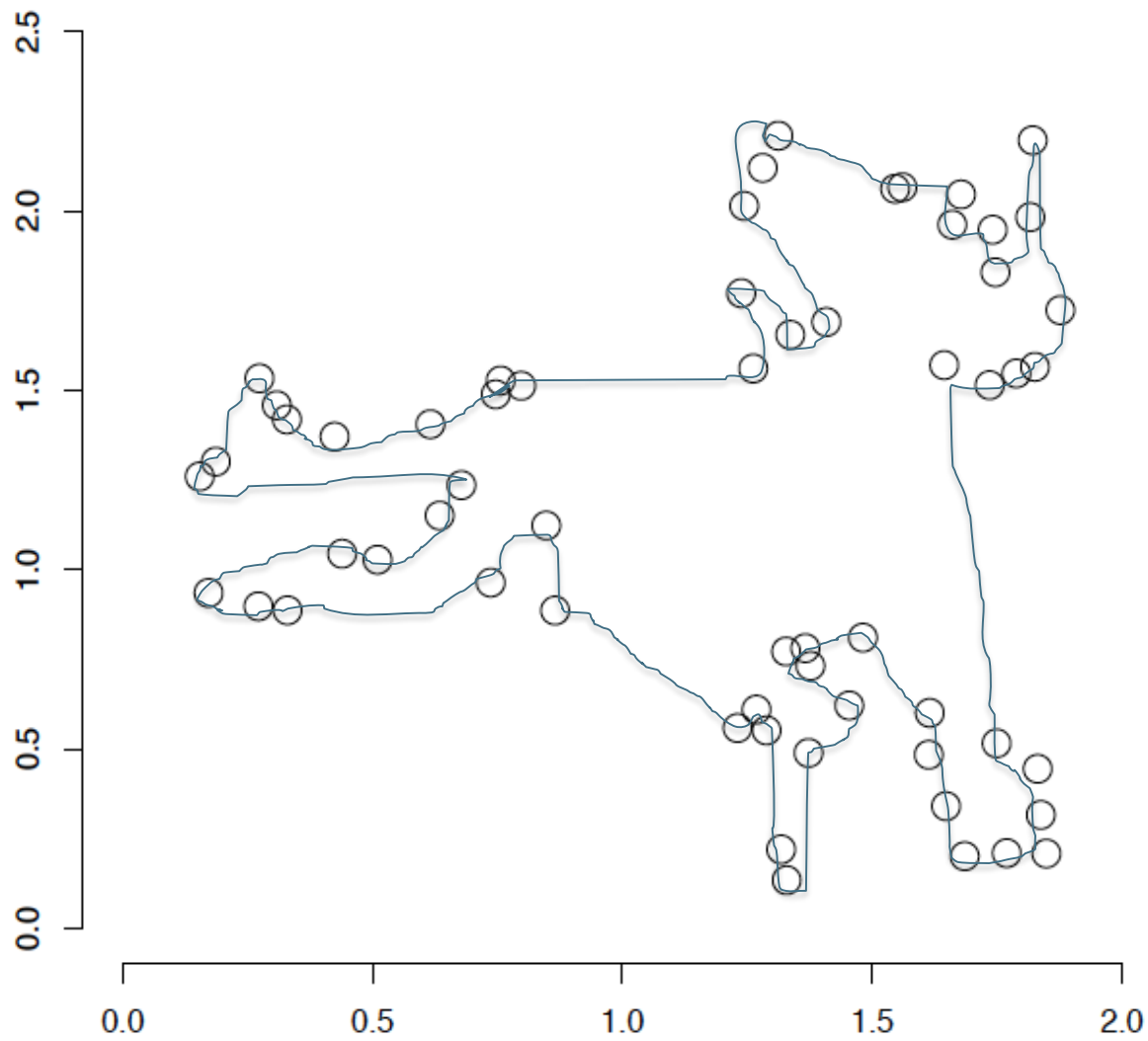
DocID reassignment

- Small **d-gaps** are much more frequent (**high probability**) than large ones within postings lists
 - this feature of posting lists is called **Clustering property**, and is passively exploited by compression algorithms
 - variable-length encoding schemes allow indexes to be compressed very well by using **shorter codes** for **small d-gaps**
- Research Question: May we permute the DocID assignment to increase the frequency of small d-gaps?
 - If yes, we may increase the compression of the index

DocID reassignment - TSP

- A technique proposed in the literature is based on the travelling salesman problem (TSP)
- The heuristic computes a *pairwise distance* between every pairs of documents
 - proportional to the number of shared terms,
 - e.g., **Jaccard distance** = $1 - JaccardSim$
- Then use TSP to find the **shorted cycle** traversing all documents in the graph.
 - The cycle is finally broken at some point
 - the DocIDs are reassigned to the documents according to the ordering established by the cycle
 - Close documents in the cycle share many terms

TSP



DocID reassignment - TSP

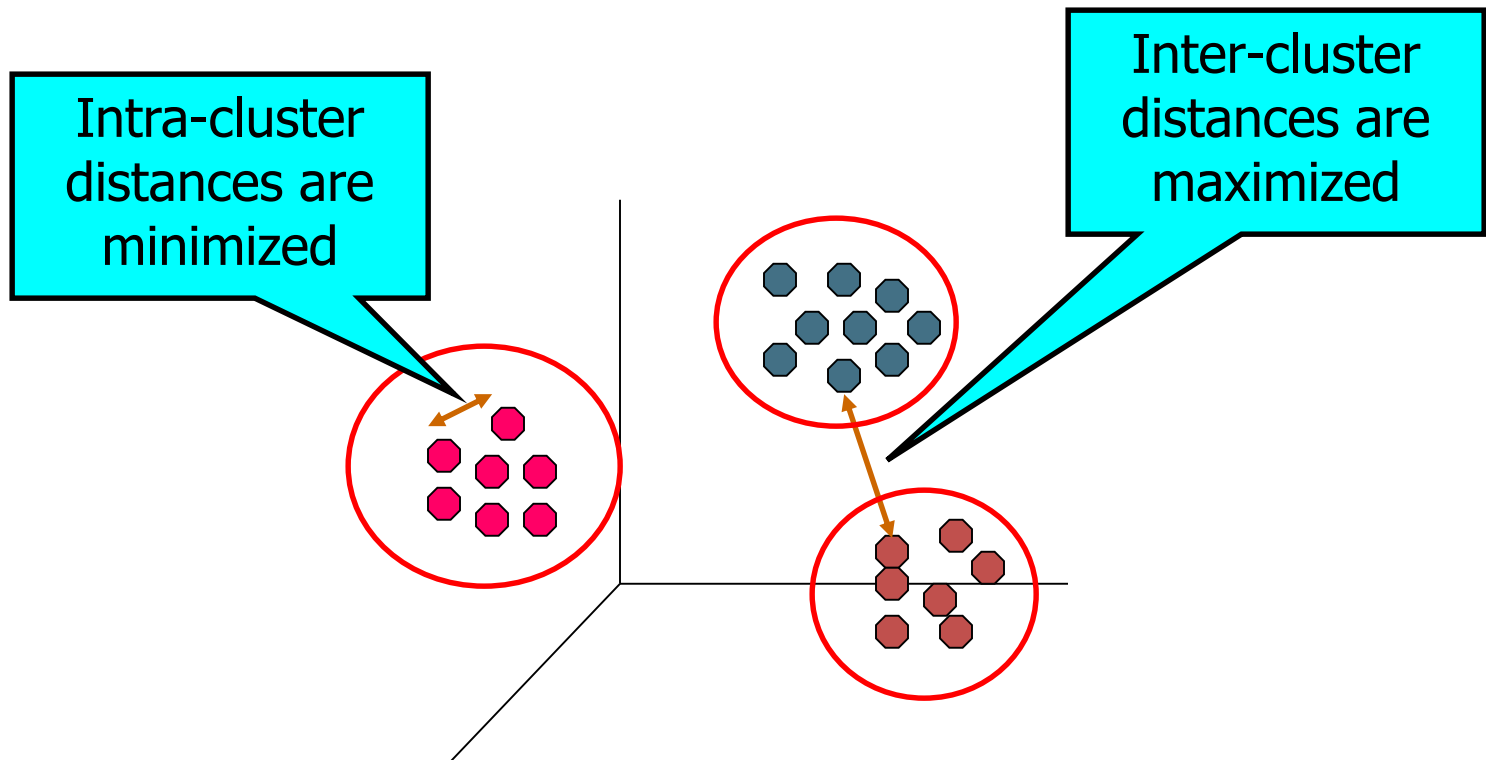
- The rationale of TSP usage
 - the TSP cycle preferably traverses edges connecting documents sharing a lot of terms (characterized by a small Jaccard distance)
 - if we assign close DocIDs to these documents, we expect a reduction in the average value of *d-gaps*, and thus in the size of the compressed inverted index
- However, this TSP approach doesn't scale

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

What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be **similar** (or **related**, or less **distant** of) to one another and different from (or unrelated to, ore more distant of) the objects in other groups



DocID reassignment:

possible scalable solution

- (1) First **cluster documents**, then (2) Exploits TSP to **reorder clusters** (rather than single documents), using the representative document of each cluster
- Possible clustering algorithm
 - **scan linearly** the documents, sorted in reverse order of length
 - Each cluster returned will be identified by a **medoid**, i.e., a document that represents all the others in the cluster
 - The medoid should be the **most centrally located** point in the cluster. *However, the stream clustering algorithm does not guarantee this property of medoids*

DocID reassignment: possible scalable solution

- Transform each document into a set of **termIDs**
- Reorder the collection according to the document length (in reverse order) and scan linearly the collection of document to clustering them using **the Jaccard distance = 1 - JaccardSim**

`C = Stream_cluster(SortedCollection, Radius)`

where C is the returned set of clusters, each cluster represented by its Medoid.

- Apply TSP to the Medoids of each cluster, using the Jaccard distances between each pair of Medoids
- Assign the DocIDs linearly cluster by cluster, using the TSP-induced order. Within each cluster the order is arbitrary.
- For each **postings list**, reassign the docIDs, compute the d-gaps, and determine the total size of all postings lists.

It is not needed to materialize the compressed posting lists, but it suffices to determine the average bits per d-gap.

DocID reassignment: possible scalable solution

- The pseudo-code of the stream algorithm that visits each document only once is the following:

Stream_cluster(SortedCollection, Radius)

C = EmptySet

for each d in SortedCollection

 Dist_c = *Min* (JaccardDistance(c, d), for each medoid c in C)

 if (Dist_c < radius) then

 add d to cluster c

 else

 make d a new medoid, and add this singleton cluster to C

return C