ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DIPARTIMENTO DI
SCIENZE STATISTICHE "PAOLO FORTUNATI"

HALLOWEEN 2021 CONFERENCE

8 - Clustering for Text Similarity

Giovanni Della Lunga

Halloween Conference in Quantitative Finance

Bologna - October 26-28, 2021

Subsection 1

Similarity

Sorting documents

Similarity

- At its core, any sorting task relies on our ability to compare two documents and determine their **SIMILARITY**.

Introduction

- Documents that are similar to each other are grouped together and the resulting groups broadly describe the overall themes, topics, and patterns inside the corpus.

Introduction

- While most document sorting is currently done manually, it is possible to achieve these tasks in a fraction of the time with the effective integration of unsupervised learning, as we will see in this lesson

Introduction

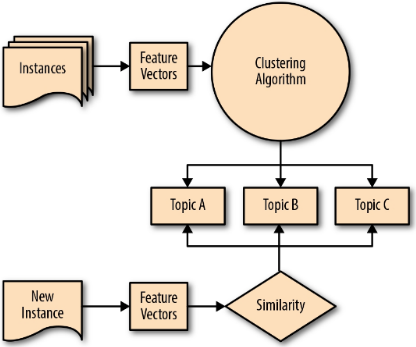
- In many situations, corpora do not arrive *pretagged* with labels ready for classification.
- In these cases, the only choice, or at least a necessary precursor for many natural language processing tasks, is an unsupervised approach.
- Clustering algorithms aim to discover **latent structure** or themes in unlabeled data using features to organize instances into meaningfully dissimilar groups.
- With text data, each **instance** is a single document or sentence, and the **features** are its tokens, vocabulary, structure, metadata, etc.

Unsupervised Learning on Text

- **Comparison between Clustering and Classification**
- The behavior of unsupervised learning methods is fundamentally different from that of supervised algorithm we have seen in the previous section;
- **Instead of learning a predefined pattern, the model attempts to find relevant patterns a priori**.
- A clustering algorithm is usually employed to create groups or topic clusters, using a distance metric such that documents that are closer together in feature space are more similar.
- New incoming documents can then be vectorized and assigned to the nearest cluster.

Unsupervised Learning on Text

- A corpus is transformed into feature vectors and a clustering algorithm is employed to create groups or topic clusters, using a distance metric such that documents that are closer together in feature space are more similar.
- New incoming documents can then be vectorized and assigned to the nearest cluster.



Unsupervised Learning on Text

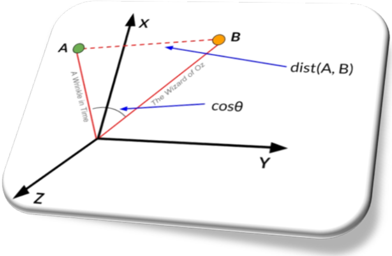
- As we have seen in section 3.1, there are a number of different measures that can be used to determine document similarity;
- Remember that, fundamentally, each relies on our ability to imagine documents as points in space, where the relative closeness of any two documents is a measure of their similarity.
- We have discussed the **cosine similarity** but there are others measure of distance we can use in clustering documents.

Unsupervised Learning on Text : Similarity

String Matching	Distance Metrics	Relational Matching	Other Matching
Edit Distance - Levenstein - Smith-Waterman - Affine **Alignment** - Jaro-Winkler - Soft-TFIDF - Monge-Elkan **Phonetic** - Soundex - Translation	- Euclidean - Manhattan - Minkowski **Text Analytics** - Jaccard - TFIDF - Cosine similarity	**Set Based** - Dice - Tanimoto (Jaccard) - Common Neighbors - Adar Weighted **Aggregates** - Average values - Max/Min values - Medians - Frequency (Mode)	- Numeric distance - Boolean equality - Fuzzy matching - Domain specific **Gazettes** - Lexical matching - Named Entities (NER)

Document Similarity

- We can measure vector similarity with cosine distance, using the cosine of the angle between the two vectors to assess the degree to which they share the same orientation.
- In effect, the more parallel any two vectors are, the more similar the documents will be (regardless of their magnitude).



Subsection 2

Clustering

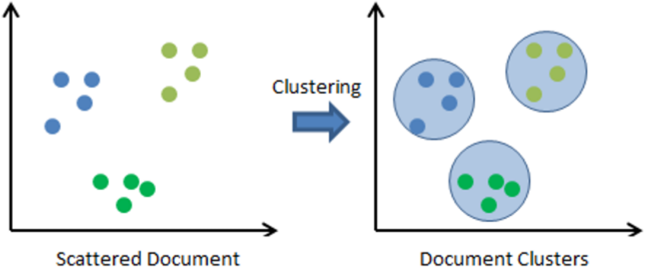
Clustering

- Now that we can quantify the similarity between any two documents, we can begin exploring unsupervised methods for finding similar groups of documents.
- **Partitive clustering** and **agglomerative clustering** are our two main approaches, and both separate documents into groups whose members share maximum similarity as defined by some distance metric.
- **We will focus on partitive methods, which partition instances into groups that are represented by a central vector (the centroid)** or described by a density of documents per cluster.
- Centroids represent an aggregated value (e.g., mean or median) of all member documents and are a convenient way to describe documents in that cluster.

Clustering

- As we have seen in the first part, clustering can be considered one of the most important unsupervised learning problem; so, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data.
- A loose definition of clustering could be *the process of organizing objects into groups whose members are similar in some way*.
- A cluster is therefore a collection of objects which are coherent internally, but clearly dissimilar to the objects belonging to other clusters.

Text Documents Clustering using K-Means Algorithm



source: *https://www.codeproject.com/Articles/439890/Text-Documents-Clustering-using-K-Means-Algorithm*

Text Documents Clustering using K-Means Algorithm

- **Document Representation**
- Each document is represented as a vector using the vector space model.
- The vector space model also called term vector model is an algebraic model for representing text document (or any object, in general) as vectors of identifiers. For example, TF-IDF weight.
- **Finding Similarity Score**
- We will use cosine similarity to identify the similarity score of a document.

Text Documents Clustering using K-Means Algorithm

- **A Practical Example: Clustering Movie Reviews**
- We are going to use data collected by Brandon Rose;
- here the link to the original post: http://brandonrose.org/top100
- And the GitHub Repository in which you can find all the data files necessary for this example: https://github.com/brandomr/document_cluster

Text Documents Clustering using K-Means Algorithm

5 Steps

- Prepare data
- Tokenizing and stemming each synopsis
- Transforming the corpus into vector space using tf-idf
- Calculating cosine distance between each document as a measure of similarity
- Clustering the documents using the k-means algorithm

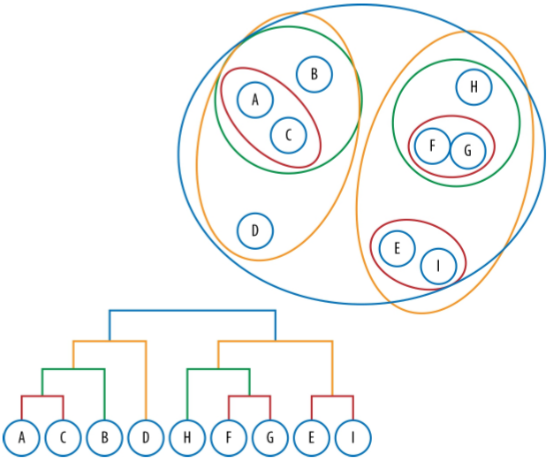
Clustering

- Working Example
- Using **08-clustering-for-text-similarity.ipynb** Notebook
- Clustering Film Reviews
- Clustering Wikipedia

Clustering

- In the previous section, we explored partitive methods, which divide points into clusters.
- By contrast, hierarchical clustering involves creating clusters that have a predetermined ordering from top to bottom.
- Hierarchical models can be either **agglomerative**, where clusters begin as single instances that iteratively aggregate by similarity until all belong to a single group, or **divisive**, where the data are gradually divided, beginning with all instances and finishing as single instances.

Hierarchical clustering



Hierarchical clustering

- Agglomerative clustering iteratively combines the closest instances into clusters until all the instances belong to a single group.
- In the context of text data, the result is a hierarchy of variable-sized groups that describe document similarities at different levels or granularities.
- Just as there are multiple ways of quantifying the difference between any two documents, there are also multiple criteria for establishing the linkages between them.
- Agglomerative clustering requires both a **distance function** and a **linkage criterion**.
- Scikit-Learn implementation defaults to the Ward criterion, which minimizes the within-cluster variance as each are successively merged.
- At each aggregation step, the algorithm finds the pair of clusters that contributes the least increase in total within-cluster variance after merging.