Document Clustering: Comparison of Similarity Measures

Shouvik Sachdeva Bhupendra Kastore

Indian Institute of Technology, Kanpur

CS365 Project, 2014

Outline

1. **Introduction**
   - The Problem and the Motivation
   - Approach
2. **Methodology**
   - Document Representation
   - Similarity Measures
   - Clustering Algorithms
   - Evaluation
3. **Related Work**
   - Past Results
   - References
4. **The End**

What is document clustering and why is it important?

- Document clustering is a method to classify the documents into a small number of coherent groups or clusters by using appropriate similarity measures.
- Document clustering plays a vital role in document organization, topic extraction and information retrieval.
- With the ever increasing number of high dimensional datasets over the internet, the need for efficient clustering algorithms has risen.

How can we solve this problem?

- A lot of these documents share a large proportion of lexically equivalent terms.
- We will exploit this feature by using a “bag of words" model to represent the content of a document.
- We will group “similar" documents together to form a coherent cluster.
- This “similarity" can be defined in various ways. In the vector space, it is closely related to the notion of distance which can be defined in several ways.
- We will try to test which similarity measure performs the best across various domains of text articles in English and Hindi.

How will we compare these similarity measures?

- We will first represent our document using the bag of words and the vector space model.
- Then we will cluster documents (now high dimensional vectors) by k-means and hierarchical clustering techniques using different similarity measures.
- Documents we will use are from varied domains from English and Hindi.
- We will then compare the performance of each similarity measure across the different kinds of documents.
- Entropy and Purity measure will be used for the purposes of evaluation.

Bag of Words: Model

- Each word is assumed to be independent and the order in which they occur is immaterial.
- Each word corresponds to a dimension in the resulting data space.
- Each document then becomes a vector consisting of non-negative values on each dimension.
- Widely used in information retrieval and text mining.

Bag of Words: Example

Here are two simple text documents:

Document 1

I don’t know what I am saying.

Document 2

I can’t wait for this to get over.

Bag of Words: Example

Now, based on these two documents, a dictionary is constructed:

“I":1
“don’t"":2
“know":3
“what":4
“am":5
“saying":6
“can’t":7
“wait":8
“for":9
“this":10
“to":11
“get":12
“over":13

Bag of Words: Example

The dictionary has 13 distinct words. Using the indices of the dictionary, the document is represented by a 13-entry vector.

Document 1

[2,1,1,1,1,1,0,0,0,0,0,0,0]

Document 2

[1,0,0,0,0,0,1,1,1,1,1,1,1]

Each entry of the vectors refers to count of the corresponding entry in the dictionary.

Representing the document formally

Let $D = \{d_1, \dots, d_n\}$ be a set of documents and $T = \{t_1, \dots, t_m\}$ be the set of distinct terms occurring in D. The document's representation in the vector space is given by an m-dimensional vector $\vec{t_d}$,

$$\vec{t_d} = (tf(d, t_1), \dots, tf(d, t_m))$$

where $tf(d, t)$ denotes the frequency of the term $t \in T$ in document $d \in D$.

Pre-processing

- First, we will remove stop words (non-descriptive such as a, and, are and do). We will use the one implemented in the Weka machine learning workbench, which contains 527 stop words.
- Second, words will be stemmed using Porter?s suffix-stripping algorithm, so that words with different endings will be mapped into a single word. For example production , produce , produces and product will be mapped to the stem produc.
- Third, we considered the effect of including infrequent terms in the document representation on the overall clustering performance and decided to discard words that appear with less than a given threshold frequency.
- We select the top 2000 words ranked by their weights and use them in our experiments.

TFIDF

- Some terms that appear frequently in a small number of documents but rarely in the other documents tend to be more relevant and specific for that particular group of documents, and therefore more useful for finding similar documents.
- To capture these terms , we transform the basic term frequencies $tf(d,t)$ into the *tfidf* (term frequency and inversed document frequency) weighting scheme.
- *Tfidf* weighs the frequency of a term t in a document d with a factor that discounts its importance with its appearances in the whole document collection, which is defined as:

$$tfidf(d,t) = tf(d,t) \times log(\frac{|D|}{df(t)})$$

- Here $df(t)$ is the number of documents in which term t appears.

Metric

A *metric space* (X, d) consists of a set X on which is defined a *distance function* which assigns to each pair of points of X a distance between them, and which satisfies the following four axioms:

1. $d(x, y) \geq 0$ for all points x and y of X;
2. $d(x, y) = d(y, x)$ for all points x and y of X;
3. $d(x, z) \leq d(x, y) + d(y, z)$ for all points x, y and z of X;
4. $d(x, y) = 0$ if and only if the points x and y coincide.

Euclidean Distance

- Standard metric for geometric problems.
- Given two documents d_a and d_b represented by their term vectors $\vec{t_a}$ and $\vec{t_b}$ respectively, the Euclidean distance is defined as

$$D_E(\vec{t_a}, \vec{t_b}) = (\sum_{t=1}^{m} |w_{t,a} - w_{t,b}|^2)^{1/2}$$

where $T = \{t_1, \ldots, t_m\}$ is the term set and the weights, $w_{t,a} = tfidf(d_a, t)$.

Cosine Similarity

- Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them.
- Given two documents $\vec{t_a}$ and $\vec{t_b}$, the Cosine similarity is defined as

  $$SIM_C(\vec{t_a}, \vec{t_b}) = \frac{\vec{t_a} \cdot \vec{t_b}}{|\vec{t_a}| \times |\vec{t_b}|}$$

  where $\vec{t_a}$ and $\vec{t_b}$ are m-dimensional vectors over the term set T.
- Non-negative and bounded between $[0, 1]$.

Jaccard Coefficient

- The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets.
- Given two documents $\vec{t_a}$ and $\vec{t_b}$, the Jaccard Coefficient is defined as

$$SIM_J(\vec{t_a}, \vec{t_b}) = \frac{\vec{t_a} \cdot \vec{t_b}}{|\vec{t_a}|^2 + |\vec{t_b}|^2 - \vec{t_a} \cdot \vec{t_b}}$$

  where $\vec{t_a}$ and $\vec{t_b}$ are *m*-dimensional vectors over the term set T.
- Non-negative and bounded between $[0, 1]$.

Pearson Correlation Coefficient

- The Pearson Correlation coefficient is a measure of the linear correlation (dependence) between two variables X and Y, giving a value between +1 and -1 inclusive, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation.
- Given two documents $\vec{t_a}$ and $\vec{t_b}$, the Pearson Correlation Coefficient is defined as

$$SIM_P(\vec{t_a}, \vec{t_b}) = \frac{m \sum_{t=1}^{m} w_{t,a} \times w_{t,b} - TF_a \times TF_b}{\sqrt{[m \sum_{t=1}^{m} w_{t,a}^2 - TF_a^2][m \sum_{t=1}^{m} w_{t,b}^2 - TF_b^2]}}$$

where $\vec{t_a}$ and $\vec{t_b}$ are m-dimensional vectors over the term set T and $TF_a = \sum_{t=1}^{m} w_{t,a}$, $w_{t,a} = tfidf(d_a, t)$.

Manhattan Distance

- The Manhattan Distance is the distance that would be traveled to get from one data point to the other if a grid-like path is followed. The Manhattan distance between two items is the sum of the differences of their corresponding components.
- Given two documents $\vec{t_a}$ and $\vec{t_b}$, the Manhattan Distance between them is defined as

$$SIM_M(\vec{t_a}, \vec{t_b}) = \sum_{t=1}^{m} |w_{t,a} - w_{t,b}|$$

where $\vec{t_a}$ and $\vec{t_b}$ are m-dimensional vectors over the term set T and $w_{t,a} = tfidf(d_a, t)$.

Chebychev Distance

- The Chebychev distance between two points is the maximum distance between the points in any single dimension.
- Given two documents $\vec{t_a}$ and $\vec{t_b}$, the Chebychev Distance is defined as

$$SIM_{Ch}(\vec{t_a}, \vec{t_b}) = \max_t |w_{t,a} - w_{t,b}|$$

where $\vec{t_a}$ and $\vec{t_b}$ are m-dimensional vectors over the term set T and $w_{t,a} = tfidf(d_a, t)$.

Hierarchical Algorithms

- Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters.
- Strategies for hierarchical clustering generally fall into two types:
    - Agglomerative (bottom up):
      This method starts with every single object (gene or sample) in a single cluster. Then, in each successive iteration, it agglomerates (merges) the closest pair of clusters by satisfying some similarity criteria, until all of the data is in one cluster. $O(n^3)$
    - Divisive(top down):
      This method starts with a single cluster containing all objects, and then successively splits resulting clusters until only clusters of individual objects remain. $O(n^2)$

Hierarchical Clustering: In action

`http://www.cs.utexas.edu/~mooney/cs391L/slides/clustering.ppt`

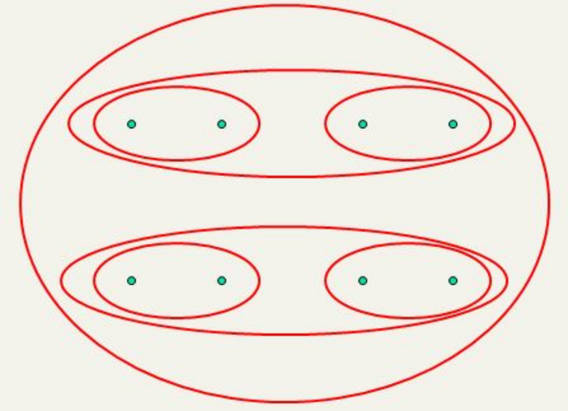


Figure: Single Link

Hierarchical Clustering: End Result

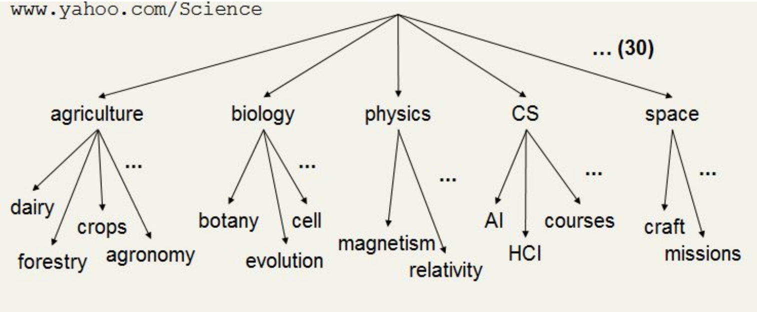


Figure: Hierarchical Clustering

k-means Algorithm

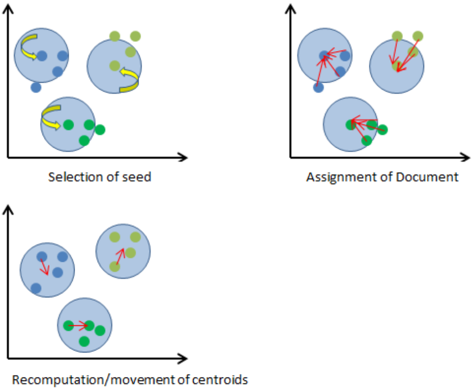
- Partitions observations into clusters resulting in a partitioning of the data space into Voronoi cells.
- First pick k, the number of clusters.
- Initialize clusters by picking one point per cluster.For instance, pick one point at random, then $k-1$ other points, each as far away as possible from the previous points.
- Try different values of k and choose the based on the average distance to the centroid.

k-means: Populating Clusters

- For each point, place it in the cluster whose current centroid it is nearest.
- After all points are assigned, fix the centroids of the *k* clusters.
- Reassign all points to their closest centroid.

k-means: In action

`http://www.codeproject.com/Articles/439890/Text-Documents-Clustering-using-K-Means-Algorithm`



Selection of seed

Assignment of Document

Recomputation/movement of centroids

Figure: *k*-means process

Effective choice for k

Effective heuristics for seed selection include:

- Excluding outliers from the seed set
- Trying out multiple starting points and choosing the clustering with the lowest cost; and
- Obtaining seeds from another method such as hierarchical clustering.

Entropy

Entropy measures the distribution of categories in a given cluster. The entropy of a cluster C_i with size n_i is defined as

$$E(C_i) = -\frac{1}{logc}\sum_{h=1}^{k}\frac{n_i^h}{n_i}log(\frac{n_i^h}{n_i})$$

where c is the total number of categories in the data set and n_i^h is the number of documents from the h^{th} class that were assigned to this cluster C_i.

Purity

Purity provides provides insight into the coherence of a cluster i.e. the degree to which a cluster contains documents from a single category. Purity for a given cluster C_i of size n_i is given by:

$$P(C_i) = \frac{1}{n_i} max_h(n_i^h)$$

where $max_h(n_i^h)$ is the number of documents that are from the dominant category in cluster C_i and n_i^h represents the number of documents from the cluster assigned to category h.

Datasets

We will use the following datasets for our project:

- 20news
- BBC/BBC Sport
- Wikipedia
- FIRE
- Classic
- r0

Anna Huang

Table 1: Purity Results

Data	Euclidean	Cosine	Jaccard	Pearson	KLD
20news	0.1	0.5	0.5	0.5	0.38
classic	0.56	0.85	0.98	0.85	0.84
hitech	0.29	0.54	0.51	0.56	0.53
re0	0.53	0.78	0.75	0.78	0.77
tr41	0.71	0.71	0.72	0.78	0.64
wap	0.32	0.62	0.63	0.61	0.61
webkb	0.42	0.68	0.57	0.67	0.75

Anna Huang

Table 2: Entropy Results

Data	Euclidean	Cosine	Jaccard	Pearson	KLD
20news	0.95	0.49	0.51	0.49	0.54
classic	0.78	0.29	0.06	0.27	0.3
hitech	0.92	0.64	0.68	0.65	0.63
re0	0.6	0.27	0.33	0.26	0.25
tr41	0.62	0.33	0.34	0.3	0.38
wap	0.75	0.39	0.4	0.39	0.4
webkb	0.93	0.6	0.74	0.61	0.51

References

- Anna Huang.
  Similarity measures for document clustering.
  In Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand, pages 49 – 56, 2008.

- D. Arthur and S. Vassilvitskii.
  k-means++ the advantages of careful seeding.
  In Symposium on Discrete Algorithms, 2007.

- Y. Zhao and G. Karypis.
  Empirical and theoretical comparisons of selected criterion functions for document clustering.
  Machine Learning, 55(3), 2004.

That’s all folks!

Thank You!
Questions?
Suggestions?