Exploring Unsupervised Algorithms in Clustering Documents

Thomas Brink and Quinn Hollister

1 Dataset

We will be using a subset of wikipedia articles entitled "smallwiki" [1]. The dataset includes two collections; a "dictionary" that enumerates each word or root that can be found in any of the documents, and a "tf.idf" dataset that lists all documents and the tf-idf scores for each word in those documents. The tf-idf score is calculated via:

$$TF - IDF_{x,y} = tf_{x,y} \cdot (\log(\frac{N+1}{df_x + 1}) + 1) \tag{1}$$

where $tf_{x,y}$ corresponds to the frequency of the word x in document y, N corresponds to the number of documents, and df_x corresponds to the number of documents containing the word x. Note that the addition of one's is used to offset possible singularities.

2 Project Idea

Our main goal in this project is to uncover interesting and useful relationships from and between documents and their contained words. We will do so by exploring various clustering algorithms. Although most of the existing literature focuses on either document or word clustering, our main approach will be following [1] in simultaneously clustering documents and words. Such a bipartite clustering approach, which relies on a more general class of spectral clustering [2], allows us to concurrently extract relational information on words as well as documents.

To assess the usefulness and efficiency of this approach, we aim to compare the bipartite approach with several baseline approaches. Two of the most straightforward baselines include K-means and Hierarchical Agglomerative Clustering (HAC) [3]. Another approach would be to use a more advanced, model-based, algorithm such as a mixture model with Expectation-Maximization (EM). To evaluate the quality of the different methods, we plan to use the Adjusted Rand Index (ARI) and Silhouette Coefficient. In addition, we will be assessing computational efficiency. Namely, one characteristic of document clustering that makes the task more difficult is the nature of the underlying feature space. Each unique word or root has its own dimension, so when we have ten thousand unique words, we are working in an extremely high-dimensional feature space. This will make many naive clustering algorithms unsuitable, or at the very least inefficient.

3 Relevant Papers

1. **Co-clustering documents and words using bipartite spectral graph partitioning**. By Inderjit S. Dhillon
2. **Hierarchial Clustering Algorithms for Document Datasets**. By Ying Zhao, George Karypis & Usama Fayyad
3. **Text Document clustering using Spectral Clustering algorithm with Particle Swarm Optimization**. By R. Janani, and Dr. S. Vijayarani

4 Milestone

By the milestone, we would like to have working code that can generate clusters for at least 4 models, 2 of which will be base classifiers like K-means and hierarchical clustering, and the other 2 will be more advanced models like the ones referenced in the relevant papers section. We will need to write software that implements the 2 complex models, but most likely not for the base classifiers (which have dedicated packages available).

[1] https://courses.cs.washington.edu/courses/cse599c1/13wi/datasets.html