Application of the fusion-fission metaheuristic to document clustering

Charles-Edmond Bichot[1]

École Centrale de Lyon, 36 av. Guy de Collongue, 69134 Ecully, France
bichot@ec-lyon.fr

Following the work of Inderjit S. Dhillon [2], this paper presents the document clustering problem as a graph partitioning problem. To solve this problem, we use the fusion-fission metaheuristic which has already been applied to several graph partitioning problems [1]. The results obtained with the fusion-fission algorithm are better than those of Graclus, a state of the art graph partitioning software created by Inderjit S. Dhillon. But surprisingly, regarding Inderjit S. Dhillon's objective function, these results are also always better than those of the real partitioning of the documents. This unexpected fact incite us to conclude that Inderjit S. Dhillon's method to convert a document clustering problem into a graph partitioning problem is wanting. Replacing the normalized cut objective function by an another objective function should minimized this problem. However, we do not suggest yet a new objective function more efficient than the normalized cut.

1 Introduction

Clustering is the partitioning of a data set into subsets, or clusters, so that the data in each subset share some common trait. Common traits are often defined as distance measures. Given a collection of unlabeled documents, the document clustering problem is to partition the documents into different clusters such that document sharing the same topics are grouped together. A common way to cluster documents is based on their word distributions. Documents which share the same vocabulary are partitioned together.

The document clustering problem can be viewed as a dual document and word clustering problem (see [2]). The idea is to extract words of documents and to create a word by document matrix A whose rows correspond to words and columns to documents. Each non-zero entry a_{ij} of this matrix corresponds to the number of occurrence of the word number i in the document number j. Then the adjacency matrix of the bipartite graph is constructed. The graph partitioning problem is solved by minimizing the normalized cut objective function.

Experiments have been made by combining set of documents of different subjects together. Thus, we know the best normalized cut values of each of these experiments. Indeed, each of these values is the normalized cut value of the partition formed by parts corresponding to each set of document.

2 Bipartite graph model

Let $G = (V, E)$ be an undirected weighted graph with a set of vertices $V = \{1, 2, \dots, n\}$ and a set of edges E. For all (i, j) in E, let e_{ij} be the weight of the edge (i, j). The weight of each vertex is equal to the sum of the weights of edges incident on it : $weight(i) = \sum_j e_{ij}$.

Given a partition of V into k subsets $\pi_k = \{V_1, \dots, V_k\}$, the cut between them is defined as : $cut(\pi_k) = \sum_{i<j} cut(V_i, V_j)$, where $cut(V_i, V_j) = \sum_{k \in V_i, l \in V_j} e_{kl}$

Then, the normalized cut objective function is defined as follows :

$$Ncut(\pi_k) = \sum_i \frac{Cut(V_i, V - V_i)}{Cut(V_i, V)}$$

A document set is represented as an undirected bipartite graph $G = (D, W, E)$ where $D = \{d_1, \dots, d_n\}$ is the set of documents and $W = \{w_1, \dots, w_m\}$ the set of words. D and W are two sets of vertices which union is V. An edge d_i, w_j exists if the word w_j appears in the document d_i. There are no edges between documents or between words. The weight on the edge d_i, w_j is the number of time the word w_j occurs in the document d_i.

Consider the $m \times n$ word by document matrix A such that the value of row i and column j is $a_{ij} = e_{ij}$. The adjacency matrix of the bipartite graph is :

$$M = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$$

The first m vertices index the words and the last n vertices index the documents.

3 Extracting words of documents

The process of extracting words of documents is a three step process :

1. The tokenize step. It consists in extracting tokens from the document. Word separators and numbers are extracted out of the document. The result is a set of tokens.
2. The stop words step. Stop words ("and", "to", "the", ...) are removed of the set of tokens. The result is a list of meaning words.
3. The stemming step. The Porter stemming algorithm has been used for this step. In this step, the commoner morphological and inflexional endings from words are removed. This is a normalization process which is essential for word enumeration in documents. The result is a list of stemmed words.

4 Results

The document sets we used can be downloaded at `ftp://ftp.cs.cornell.edu/pub/smart`. Four experiments were made, two of them are presented in table 1. The first experiment is a compounding of Medline and Cranfield document sets and the second is a compounding of Medline, Cranfield and Cisi document sets. For all of the experiments, the fusion-fission algorithm finds partitions with a normalized cut value lower than the normalized cut value of the original document set partition which should be the minimum value. However, despite the normalized cut value of partitions found by fusion-fission is lower, the corresponding clustering are not the same, but worse than those of the original document set partitions. On the other hand, the state of the art graph partitioning package Graclus always find partitions with a upper normalized cut value than those of the original document set partitions and the clusters it find are worse than those found by fusion-fission.

These results encourage us to conclude that Dhillon's method to convert a document clustering problem into a graph partitioning problem is wanting. Especially, it seems that the problem becomes of the normalized cut objective function, then replacing it by another objective function should minimized this problem, but probably not cancelled it.

Algorithm	Cluster	Medline	Cranfield
Real clusters	D_0	1033	0
	D_1	0	1400
fusion-fission	D_0	1019	0
	D_1	14	1400
Graclus	D_0	765	0
	D_1	268	1400

2 433 docs, 10 683 words, 129 601 edges

Algorithm	Cluster	Medline	Cranfield	Cisi
Real clusters	D_0	1033	0	0
	D_1	0	1400	0
	D_2	0	0	1460
fusion-fission	D_0	903	0	4
	D_1	7	1385	8
	D_2	123	13	1448
Graclus	D_0	874	0	0
	D_1	18	1384	9
	D_2	141	14	1451

3 893 docs, 13 192 words, 192 857 edges

Table 1. Comparision between real clusters and those found by fusion-fission and Graclus

References

1. C.-E. Bichot. A new meta-method for graph partitioning. In *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*, pages 3498–3505, June 2008.
2. I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 269–274, 2001.
3. I. S. Dhillon, Y. Guan, and B. Kulis. A fast kernel-based multilevel algorithm for graph clustering. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 629–634, 2005.