Cluster analysis in document networks

C. K. dos Santos, A. G. Evsukoff & B. S. L. P. de Lima
COPPE/UFRJ, Federal University of Rio de Janeiro,
Rio de Janeiro, Brazil

Abstract

Text or document clustering is a subset of a larger field of data clustering and has been one of the research hotspots in text mining. On the other hand, recent studies have shown that many real systems may be represented as complex networks with astonishing similar proprieties. In this work a document corpora is represented as a complex network of documents, in which the nodes represent the documents and the edges are weighted according to the similarities among documents. The detection of community structures in complex networks can be seen as the cluster analysis in document networks. Recently community detection algorithms based on spectral proprieties of the underlying has shown good results. The main motivation for applying those methods is that they have shown to be robust to the high dimensionality of feature space and also to the inherent data sparsity resulting from text representation in the vector space model. The aim of this paper is to present the application of the community structures algorithms for text mining. Experiments have been carried out on the document clustering problems taken from 20 newsgroup document corpora to evaluate the performance of the proposed approach.

Keywords: text mining, document clustering, complex networks, community detection, spectral clustering.

1 Introduction

Unstructured information in document databases presents intrinsic characteristics such that the classical data mining algorithms can be adapted to solve text mining tasks. One of the most usual representations for text mining relies on the vector space information retrieval model of documents [1]. In such a model the order of words is not considered and each document in a collection is represented by a vector, of which the components are related to relevant words appearing in

WIT Transactions on Information and Communication Technologies, Vol 40, © 2008 WIT Press
www.witpress.com, ISSN 1743-3517 (on-line)
doi:10.2495/DATA080101

the document collection. The resulting table, often called the Bag-of-Words (*BoW*), can then be processed by data mining algorithms [2].

The main focus of this paper is document clustering, which is one of the most challenge problems in text mining research. Spectral clustering algorithms provide robustness to high dimensionality and sparsity of feature space defined over the vector space model used to represent documents.

In this work, the document collection is analyzed as a complex network, i.e. a graph, of which the nodes are the documents and the edges are weighted according to document similarities. The theory of complex networks is a very active multidisciplinary research field and many new results are available [3–5]. One of the most active areas in the study of complex network is the detection of community structure in networks [6–13], which has many applications in social and biological sciences.

Document clusters can be regarded as communities structures in complex network analysis, such that the best clustering assignments can be determined from the one resulting in the best community structure in the network. A robust approach to this problem is the maximization of the function known as "modularity", introduced by Newman and Girvan [8].

Community detection algorithms are based on the concept of similarity between records instead of distance, as other algorithms do. The purpose of a similarity graph is to connect data points in "local neighborhoods". Each data point corresponds to a vertex in the neighborhood graph. Depending on the type of similarity graph, "close" vertices will be connected.

The algorithms are usually formulated as graph partition problem where the weight of each edge is the similarity between points that correspond to vertex connected by the edge. The goal of the algorithm is find the minimum weight cuts in the graph, but this problem can be addressed by the means of linear algebra, in particular by the eigenvalues decomposition techniques [14]. They have thus strong connection to community structure detection [15].

In next section, the basic pre-processing task for text mining is introduced. In section three the representation of the document collection as a complex network is presented and the algorithm to detect communities in the network is applied to document clustering. Section four the results are presented and discussed. The paper ends with the conclusions and future work in section five.

2 Documents pre-processing

Text mining refers to the detection of trends, patterns, or similarities in natural language text. Given a collection of text documents, often the need arises to classify the documents into groups or clusters based on similarity of content. For a relatively small collection, it may be possible to manually perform the partitioning of documents into specific categories. But to partition large volumes of text, the process would be extremely time consuming.

We consider the application of clustering to the self organization of a textual document collection. Instead of exploring the whole collection of documents, a user can then browse the resulting clusters to identify and retrieve relevant

documents. As such, clustering provides a summarized view of the information space by grouping documents by topics. Clustering is often the only viable solution to organize large text collections into topics.

Document clustering involves two phases: first, feature extraction maps each document to a point in high-dimensional space and then, clustering algorithms automatically group the points into a structure of clusters.

As usual in text mining document clustering requires a pre-processing stage, where the unstructured document collection is structured into a numeric table, often called the Bag-of-Words (*BoW*).

Document pre-processing includes the elimination of "stopwords" and the application of stemming algorithms. Nevertheless, the most important issue is the counting of frequencies that will actually be stored in the *BoW*.

The *BoW* is a table, of which the lines are related to the documents and the columns are related to the words (terms) that appear in the entire collection. The collection of document is thus represented by the set $D = \{D_i, i = 1...N\}$ and the set of all terms is denoted as $T = \{T_j, j = 1...M\}$.

Although it is generally more interesting to store the *BoW* using special data structures due to dimensions involved, mathematically it can be considered as a $N \times M$ sparse matrix of which an element x_{ij} represents an index that related the term $T_j \in T$ in the document $D_i \in D$. The vector representation $\mathbf{x}_i = (x_{i1}, \ldots, x_{iM})$ is usually adopted in information retrieval, such that classical results of linear algebra can be employed [1]. This has been called the vector space representation of documents. A similar representation can be used for terms, according to the objectives of the study.

The most usual metric to compute indexes in the *BoW* is the *TF-IDF* (term frequency–inverse document frequency) index, which is based on information theory and defines the importance of a term in the document set.

The *TF-IDF* index is the product of two factors: the term frequency and the logarithm of the inverse document frequency. The frequency of a term T_j in the document D_i is the number of occurrences of T_j in D_i, divided by the total number of terms in the document:

$$TF(D_i, T_j) = \frac{n_{ij}}{\sum_{k=1}^{M} n_{ik}} \tag{1}$$

where n_{ij} is the number of occurrences of the term T_j in the document D_i.

The second factor is inverse document frequency of the term T_j, computed as:

$$IDF(T_j) = \log\left(\frac{N}{N_j}\right) = \log N - \log N_j \tag{2}$$

where N_j is the number of documents that contains the term T_j at least once (or other pre-defined value).

The *TF-IDF* index, which is actually stored at the BoW is computed as:

$$BoW(D_i, T_j) = x_{ij} = TF(D_i, T_j).IDF(T_j) \tag{3}$$

The *TF-IDF* index results in a weighted frequency such that a very frequent term that appears in almost every document will have a low *IDF* while a term that occurs in a few documents will have higher *IDF* value. The composed *TF-IDF* index may be interpreted as the importance of the term T_j to the document D_i.

When the documents in the document set are related to a great number of subjects, the *BoW* computed by the *TF-IDF* index will generally result in a sparse matrix since many terms do not appear in all documents in the collection. The spectral clustering algorithms will be presented in next section, where considered data are based on TF-IDF measures described here.

3 Clustering document networks

Many systems can be represented as networks. That is, a set of nodes joined in pairs by edges. The study of networked systems has experienced particular interest in the last decade [3–5]. One issue that has received a considerable attention is the identification of the community structure in networks [6–13]. In this work community structure is used to cluster documents represented as a document network, as described next.

3.1 Document networks

The document corpora can be regarded as a complex network, of which the nodes are related to the documents and the edges are weighted by the similarities among them.

The network is formally denoted as a undirected and weighted graph $G = (V, E)$, such that the affinity matrix is symmetric, real valued and its elements represent the similarity between two documents. When constructing similarity graphs the goal is to model the local neighborhood relationships between the data points, in this case, documents.

Several similarities metrics may be used to compute the affinity matrix allowing different results. In this works the Gaussian function is used as similarity metric, such that each element of affinity matrix $\mathbf{A}$ is defined as:

$$a_{ij} = \begin{cases} \exp\left(-\dfrac{\left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2}{2\sigma^2} \right), & if\ i \neq j \\ 0 & otherwise \end{cases} \tag{4}$$

where σ is a dispersion parameters that controls the spread of the similarity function.

Most of the algorithms in community detection don't apply to weighted networks, such that two nodes are connected if the similarity of the corresponding data points is greater than epsilon (a threshold defined ad-hoc). This approach is often called the ε-neighborhood approach [14].

In the next section, algorithms to detect community structures in networks are presented for clustering the document networks.

3.2 Community structure in networks

A community structure in a network is defined as a group of vertices that have a high density of edges within them, with a lower density of edges between groups. Formally, for all nodes i in the community C the number of connections node belonging its own community k_i^{in} is larger than k_i^{out}, the number of connections it has to the rest of the network. [13], such that:

$$k_i^{in} > k_i^{out}, \forall i \in C. \tag{6}$$

Further, they define a community in a weak sense, such that the sum of internal connections is larger than the sum of external ones:

$$\sum_{i \in C} k_i^{in} > \sum_{i \in C} k_i^{out} \tag{7}$$

Newman and Girvan [8] defined a quantitative measure to evaluate an assignment of nodes into communities called modularity, that can be used to compare different assignments of nodes into communities quantitatively. The network modularity Q is defined as:

$$Q = \sum_i (e_{ii} - a_i^2) \tag{8}$$

where the index i runs over all communities. The fraction of all links connecting nodes in group i and j is denoted by e_{ij}. Hence, e_{ii} is the fraction of all links lying within group i. The fraction of all links connecting to nodes in group i is denoted by $a_i = \sum_j e_{ij}$. One can interpret a_i^2 as the expected fraction of internal links in group i, if the network was random and the nodes were distributed randomly into the different groups.

If the number of within-community edges is no better than random, then the value $Q = 0$. A value $Q = 1$, which is the maximum, indicate strong community structure. In practice however, values typically fall in the range from about 0.3 to 0.7 [8].

The modularity matrix approach can be optimized by eigenvalues analysis as described next.

3.3 Spectral modularity optimization

The algorithm that has been proposed by Newman for community structure detection [9] uses a new matrix **B** that represents a characteristic matrix for network in terms of its spectral properties, called modularity matrix. This

approach is a reformulation of the modularity function Q, presented in (8) and that can be conveniently written in its generalized matrix form as:

$$Q = \frac{1}{4m}\mathbf{s}^{\mathrm{T}}\mathbf{B}^{(g)}\mathbf{s} \tag{9}$$

where $\mathbf{s}$ is the column vector whose elements are the $s_i \in \{-1,1\}$, that represents the elements belonging to group 1 if $s_i = 1$, and to group 2 if $s_i = -1$. The total number of edges in the networks m is defined as:

$$m = \frac{1}{2}\sum_i d_i \tag{10}$$

The network vertices degree d_{ij} is defined as (5), and the real symmetric generalized modularity matrix $\mathbf{B}^{(g)}$ has elements:

$$B_{ij}^{(g)} = B_{ij} - \delta_{ij}\sum_{k\in g} B_{ij} \tag{11}$$

where δ_{ij} is the Kronecker δ-symbol, $k \in g$ represents the k elements belonging to g group, and the traditional modularity matrix $\mathbf{B}$ has its elements defined as:

$$B_{ij} = A_{ij} - \frac{d_i d_j}{2m} \tag{12}$$

This formulation allows maximize the modularity by choosing an appropriate division of the network based on the signs of eigenvector elements related to largest (positive) eigenvalues of $\mathbf{B}^{(g)}$, computed by the eigendecomposition:

$$\mathbf{B}^{(g)} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}} \tag{13}$$

where $\mathbf{V}$ is the orthogonal matrix of the eigenvectors and $\mathbf{\Lambda}$ is a real diagonal matrix of the eigenvalues. Not all eigenvalues must be computed such that efficient algorithms can be used.

The elements which sign are positives stay on a cluster and those with negative ones on the other, as seen above. The procedure follows subdividing the network repeatedly and computing the modularity function upon the partitions. If exists no division of the sub network that will increase the modularity of the network, then there is nothing to be gained by dividing the sub network and the procedure must be broken.

This happen when there is no positive eigenvalues to the matrix $\mathbf{B}^{(g)}$, providing the termination check of the subdivision process trough the leading eigenvalue, make the network indivisible.

This spectral approach is very interesting because there is no necessity to known in advance the number k of cluster, once it is determined by the procedure itself, without the necessity of the another algorithm like k-means for example, as happen with another spectral algorithms.

The Newman's algorithm is composed of these steps:

1. Compute the affinity matrix $\mathbf{A}$ like (4);
2. Compute the generalized modularity matrix $\mathbf{B}^{(g)}$ for all network elements like (11);
3. Make the eigendecomposition of the $\mathbf{B}^{(g)}$ and discover the leading eigenvalue;
4. Construct the first division of the network based on signs of the eigenvector related to the leading eigenvalue;
5. Compute modularity Q of the partitioning based on (9);
6. Verify if the leading eigenvalue is positive;
7. In the case of the leading eigenvalue to be positive then go to step 2 and continue the procedure;
8. In the case of the leading eigenvalue to be negative then the procedure must be left alone and to be finished.

The result of the application of this algorithm to some benchmark corpora are shown in the next section.

4 Experiment results

To test the algorithm presented in the previous section, we have applied it to three document clustering problems. All that based on 20 Newsgroup dataset(http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html), which is a collection of 20,000 messages, collected from 20 different newsgroups. One thousand messages from each of the twenty newsgroups were chosen at random and partitioned by newsgroup name. In this works a extraction of the entire dataset was used as described by Table 1.

Table 1: Document collections.

Problem #	Classes extracted from the Corpora	Documents per Class
1	(1) sci.crypt; (2) sci.electronics; (3) sci.med; (4) sci.space	500
2	(1) comp.sys.ibm.pc.hardware; (2) sci.space; (3) talk.politics.guns; (4) soc.religion.christian;	500
3	(1) alt.atheism; (2) sci.crypt; (3) comp.sys.mac.hardware; (4) misc.forsale; (5) rec.motorcycles; (6) soc.religion.christian; (7) talk.politics.misc	300

The computation of the affinity matrix, despite its symmetry, is the most time consuming step of each experiment. The remaining steps are performed quickly.

In the first experiment, about scientific documents, the algorithm found 5 clusters, and a modularity value of 0.6049. Although there are only four groups into that document collection, the five groups found are a good approximation of the document structure.

The second experiment has performed into four distinct document collections, where a modularity value of 0.6485 has been found through the partition of the document network into 5 clusters. The figures 1 and 2 show the network affinity matrix of the experiments #1 and #2, respectively.

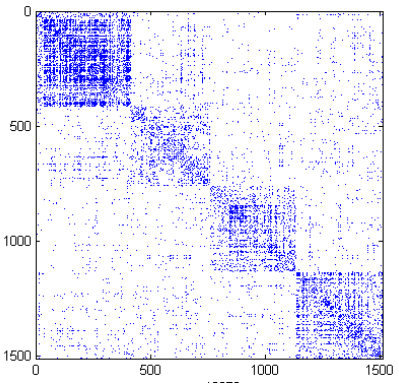

Figure 1: Network affinity matrix – problem #1.

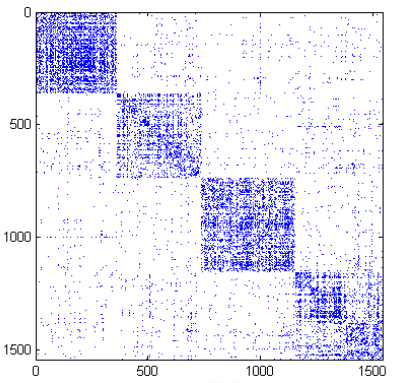

Figure 2: Network affinity matrix – problem #2.

In the case of the network affinity matrix of the problem #2, we can see the group at right bottom subdivided into two clusters, not too well pronounced, but visible. This could to justify the five groups found by algorithm.

In the last experiment the modularity value found was 0.6575, and the number of groups discovered 8, into a document collection with seven different classes. In this case the network affinity matrix (Fig. 3) show six groups well pronounced, but with some subgroups overlapping inside them.

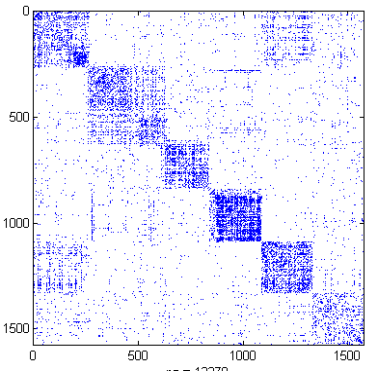

Figure 3: Network affinity matrix – problem #3.

Although in all cases the number of classes has been different from the clusters number discovered by the algorithm, the experiments achieved good results of the modularity value for real life networks, about 0.6 where values considered good by the bibliography is at maximum about 0.7 in the real life situations.

Another explanation to the differences between the classes number and the clusters number is the fact that documents are hierarchically organized generally, and the network modularity managed to capture the deeper network structure, not exposed by classes, but identified by the algorithm.

5 Conclusions

This work has presented an original approach for document clustering based on community structure detection in complex networks. The representation of a document corpus as a network allow the analysis and visualization of the document set as a graph, such that results issued from graph theory and complex network analysis can be used as text mining tools.

The results presented in this paper are preliminary but motivating. The graph representation of document corpus can deal with the high dimensionality of the vector space representation. The algorithms are robust against the scarcity of the matrix, but when the groups are not well separated the results may not identify the good number of clusters. Nevertheless, the visualization of the affinity matrix provides an important value from the point of view of human-computer interaction.

The future direction of this work is to further investigate the community structure detection algorithms and also identify the connections of spectral and kernel clustering algorithms in order to enhance the results.

Acknowledgements

The authors are grateful to the Brazilian Research Agencies, CNPq, FINEP and FAPERJ, for the financial support for this research.

References

[1] W.B. Michael, D. Zlatko, and R.J. Elizabeth, "Matrices, Vector Spaces, and Information Retrieval," SIAM Rev.1999, pp. 335–362.
[2] M. Berry, Survey of Text Mining : Clustering, Classification, and Retrieval, Springer, 2003.
[3] A.-L. Barabási, Linked : how everything is connected to everything else and what it means for business, science, and everyday life, Plume, 2003.
[4] M.E.J. Newman, "The Structure and Function of Complex Networks," SIAM Review, pp. 167–256, 2003.
[5] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.U. Hwang, "Complex networks: Structure and dynamics," Physics Reports, pp. 175–308, 2006.
[6] M.E.J. Newman, "Finding community structure in networks using the eigenvectors of matrices," doi: 10.1103/PhysRevE.74.036104, 2006.
[7] M.E.J. Newman, "Modularity and community structure in networks," PNAS 0601602103, 2006.
[8] M.E.J. Newman, and M. Girvan, "Finding and evaluating community structure in networks," Phys. Rev. E 69, 026113, 2004.
[9] M.E.J. Newman, "Fast algorithm for detecting community structure in networks," Phys. Rev. E 69, 066133, 2004.
[10] E.A. Leicht, and M.E.J. Newman, "Community structure in directed networks," arXiv:0709.4500v1 2007.
[11] B. Karrer, E. Levina, and M.E.J. Newman, "Robustness of community structure in networks," arXiv:0709.2108v1, 2007.
[12] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," Nature 435, pp. 814–818, 2005.
[13] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," Proc. Natl. Acad. Sci. USA 101, 2658–2663, arXiv:cond-mat/0309488v2, 2004.
[14] U. von Luxburg, "A tutorial on spectral clustering," Statistics and Computing 17(4), arXiv:0711.0189v1, pp. 395–416, 2007.
[15] S. White, and P. Smyth, "A spectral clustering approach to finding communities in graphs," SIAM International Conference on Data Mining, pp. 76–84, 2005.