Comparison of Algorithms for Web Document Clustering Using Graph Representations of Data

Adam Schenker[1], Mark Last[2], Horst Bunke[3], and Abraham Kandel[1,4]

[1] University of South Florida, Tampa FL 33620, USA
[2] Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel
[3] University of Bern, CH-3012 Bern, Switzerland
[4] Tel-Aviv University, Tel-Aviv 69978, Israel

Abstract. In this paper we compare the performance of several popular clustering algorithms, including *k*-means, fuzzy *c*-means, hierarchical agglomerative, and graph partitioning. The novelty of this work is that the objects to be clustered are represented by graphs rather than the usual case of numeric feature vectors. We apply these techniques to web documents, which are represented by graphs instead of vectors, in order to perform web document clustering. Web documents are structured information sources and thus appropriate for modeling by graphs. We will examine the performance of each clustering algorithm when the web documents are represented as both graphs and vectors. This will allow us to investigate the applicability of each algorithm to the problem of web document clustering.

1 Introduction

The topic of clustering has long been studied in the pattern recognition community. The goal of clustering is to create groups of data items in an unsupervised fashion such that items in the same cluster are similar to each other yet dissimilar to items in other clusters. Many algorithms are presented in the literature [1], such as *k*-means [2], hierarchical agglomerative clustering [3], fuzzy *c*-means [4], and graph partitioning [5]. A more recent method is the global *k*-means method, which attempts to find a "good" initialization state for the *k*-means algorithm [6].

Common to many clustering algorithms is that they are expected to work on data (usually numeric) which is represented by vectors (i.e. sets of attribute values). However, using such vector representations may lead to the loss of the inherent structural information in the original data. Consider, for example, the case of web document clustering. With the increasingly large amount of Internet-based content, it is difficult and costly to categorize and cluster every document manually. In order to deal with this problem, automated clustering of web documents, which allows them be more easily browsed, organized, and cataloged with minimal human intervention, is an important research area [7][8]. Under the vector space model of document representation [3], which is often applied to web documents, each term which may appear on a document is represented by a vector component (or dimension). The values associated with each dimension

A. Fred et al. (Eds.): SSPR&SPR 2004, LNCS 3138, pp. 190–197, 2004.

indicate either the frequency of the term or its relative importance according to some weighting scheme. A problem with representing web documents in this manner is that certain information, such as the order of term appearance, term proximity, term location within the document, and any web specific information, is lost under the vector model. Graphs are a more robust data structure, capable of capturing and maintaining this additional information.

Until recently, there have been no mathematical frameworks available for dealing with graphs in the same fashion that we can deal with vectors. Clustering algorithms require the computation of similarity (or distance) between two objects. This is easily accomplished with vectors in a Euclidean feature space, but until recently it has not been possible with graphs [9][10][11]. Further, a representative of a cluster (such as a centroid) is sometimes required for clustering; again, we have not had such tools available for graphs until lately [12].

Given these new graph-theoretic foundations, a version of the k-means algorithm which can cluster data that is represented by graphs rather than by vectors has been proposed [13]. The experimental results, which compared clustering performance with the traditional vector-based k-means, showed that the performance when representing documents by graphs usually exceeds that of the corresponding vector-based approach.

In this paper we will compare the clustering performance of several different classical clustering algorithms when using data represented by graphs. As mentioned above, it has been shown that the graph representation scheme under the k-means algorithm compares favorably with the vector approach [13] in terms of clustering performance. We have already investigated the effects of various graph distance measures [14] and graph representations [15] on clustering performance. However, the impact of changing the underlying clustering algorithms when clustering data represented by graphs has not been examined. Given graph-theoretic distance and centroid definitions, we can adapt many different clustering algorithms to work with graph-based data in addition to k-means.

Clustering data which is represented by graphs is a novel approach, and it is important for the reader to realize the difference between this method and the well known graph partitioning clustering procedure [5]. In our approach, the data items themselves (web documents for the application presented here) are represented by graphs which are then used in a classical clustering algorithm; by contrast, in graph partitioning clustering, each data item is represented by a single node in a graph representing the clustering problem, with edge weights indicating similarity between nodes (data items). Clustering with graph representations is also discussed in [16], where structural similarity is determined by the subgraph relation and graphs are restricted to having numerical-valued attributes. Our approach uses the size of the maximum common subgraph to calculate real-valued distances between pairs of graphs and does not place any restriction on the attributes or labels associated with the nodes and edges of graphs.

The remainder of the paper is organized as follows. In Sect. 2 we will briefly explain how clustering algorithms can utilize data that is represented by graphs.

We will also present one of our novel graph representations for web documents. The experimental results comparing each clustering algorithm will be given in Sect. 3. Sect. 4 contains our concluding remarks.

2 Clustering Algorithms for Graph Representations

In order to deal with data represented by graphs instead of the traditional case of vectors when using a clustering algorithm, we need two mathematical definitions: distance between graphs and a representative of a set of graphs. A distance measure for graphs, based on the maximum common subgraph (the largest graph shared in common by two graphs) has been proposed [9]:

$$dist(G_1, G_2) = 1 - \frac{|mcs(G_1, G_2)|}{\max(|G_1|, |G_2|)} \tag{1}$$

where G_1 and G_2 are graphs, $mcs(G_1, G_2)$ is their maximum common subgraph, $\max(\ldots)$ is the standard numerical maximum operation, and $|\ldots|$ denotes the size of the graph. The size of a graph is taken in the current work to be the number of nodes and edges contained in the graph. In the general case the computation of the maximum common subgraph is NP-Complete [17], but for our graph representation of web documents (described below) the computation of the maximum common subgraph can be accomplished in polynomial time due to the existence of unique node labels [18].

In order to create a representative of a set of graphs, we can use the median of set of graphs [12]:

$$g = \arg\min_{\forall s \in S} \left(\frac{1}{n} \sum_{i=1}^{n} dist(s, g_i) \right) \tag{2}$$

In basic terms, the median is the graph g in the set of graphs S (where $S = \{g_1, g_2, \ldots, g_n\}$) which has the minimum average distance to all other graphs in the set.

By using Eqs. 1 and 2 instead of vector distance calculations or centroid calculations, respectively, we can arrive at a version of a classical clustering algorithm which can utilize data represented by graphs. In order to represent web documents using graphs and maintain the information that is usually lost in a vector model representation, we use the following method. First, each term (word) appearing in the web document, except for stop words such as “the”, “of”, and “and” which convey little information, becomes a node in the graph representing that document. This is accomplished by labeling each node with the term it represents. Note that we create only a single node for each word even if a word appears more than once in the text. Thus each node in the graph represents a unique word and is labeled with a unique term not used to label any other node. Second, if word a immediately precedes word b somewhere in a “section” s of the web document, then there is a directed edge from the node corresponding to a to the node corresponding to b with an edge label s. We take into account

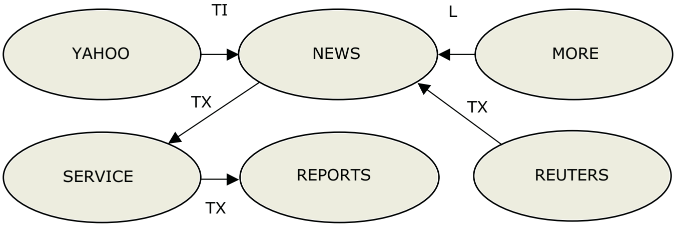

Fig. 1. Example of a graph representation of a document.

certain punctuation (such as a period) and do not create an edge when these are present between two words. Sections we have defined are: *title*, which contains the text related to the web document's title and any provided keywords; *link*, which is text appearing in clickable hyperlinks on the web document; and *text*, which comprises any of the readable text in the web document (this includes link text but not title and keyword text). Next, we remove the most infrequently occurring words for each document by deleting their corresponding nodes, leaving at most m nodes per graph (m being a user provided parameter). This is similar to the dimensionality reduction process for vector representations but with our method the term set can be different for each document. Finally, we perform a simple stemming method and conflate terms to the most frequently occurring form by re-labeling nodes and updating edges as needed. An example of this type of graph representation is given in Fig. 1. The ovals indicate nodes and their corresponding term labels. The edges are labeled according to title (TI), link (L), or text (TX). The document represented by the example has the title "YAHOO NEWS", a link whose text reads "MORE NEWS", and text containing "REUTERS NEWS SERVICE REPORTS". Note also there is no restriction on the form of the graph and that cycles are allowed. While this appears superficially similar to the bigram, trigram, or N-gram methods [19], those are statistically-oriented approaches based on word occurrence probability models. Our method does not require or use the computation of term probabilities.

3 Experimental Results

In order to evaluate the performance of our proposed method of using graphs with the various clustering algorithms, we performed several experiments on two different collections of web documents, called the F-series and the J-series[1]. Each collection contains web documents in HTML format. The F-series originally contained 98 documents assigned to one or more of 17 sub-categories of four major category areas. Since there are multiple sub-category classifications from the same category area for many of these documents, we have reduced the categories to just the four major categories in order to simplify the problem.

[1] The data sets are available under these names at: ftp://ftp.cs.umn.edu/dept/users/boley/PDDPdata/

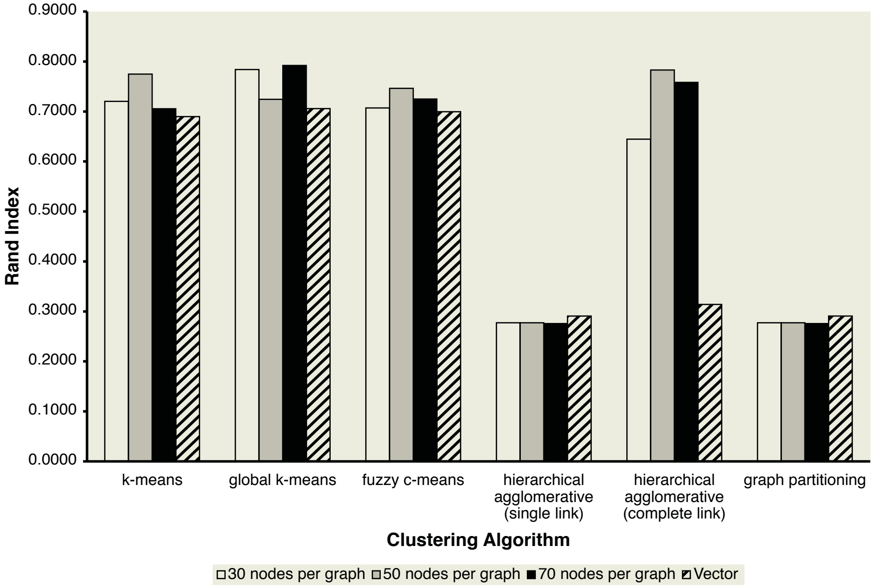

Fig. 2. Experimental results for the F-series data set.

There were five documents that had conflicting multiple classifications (i.e., they were classified to belong to two or more of the four major categories) which we removed, leaving 93 total documents. The J-series contains 185 documents and ten categories. We have not modified this data set. Clustering performance is calculated by the Rand Index [20], which is defined as the number of agreements (i.e. pairs of items which both appear together in a ground truth cluster and a cluster created by the clustering algorithm; *or* pairs of items which appear in different clusters in both ground truth and the created clusters) divided by the number agreements and disagreements (i.e. those cases that are not agreements). Thus the Rand Index is a measure of how closely the clustering created by some clustering algorithm matches ground truth (i.e. it is a measure of clustering accuracy). A value of 1.0 indicates a clustering that exactly matches ground truth.

The clustering performance results for the F-series and the J-series for the various graph-based clustering algorithms are given in Figs. 2 and 3, respectively. The charts show the performance of each algorithm as a group of four columns. From left to right the algorithms compared are: k-means, global k-means, fuzzy c-means, hierarchical agglomerative clustering (single link), hierarchical agglomerative clustering (complete link), and graph partitioning. For our experiments we varied the maximum number of nodes allowed per graph, which is the parameter m described in the previous section. Within each group of columns, the white (leftmost) bar indicates using 30 nodes per graph maximum. The grey bars correspond to using 50 nodes per graph maximum, while the black bars are the results when using 70 nodes per graph maximum. The rightmost (striped) bars represent the performance of the traditional vector-based approach using a

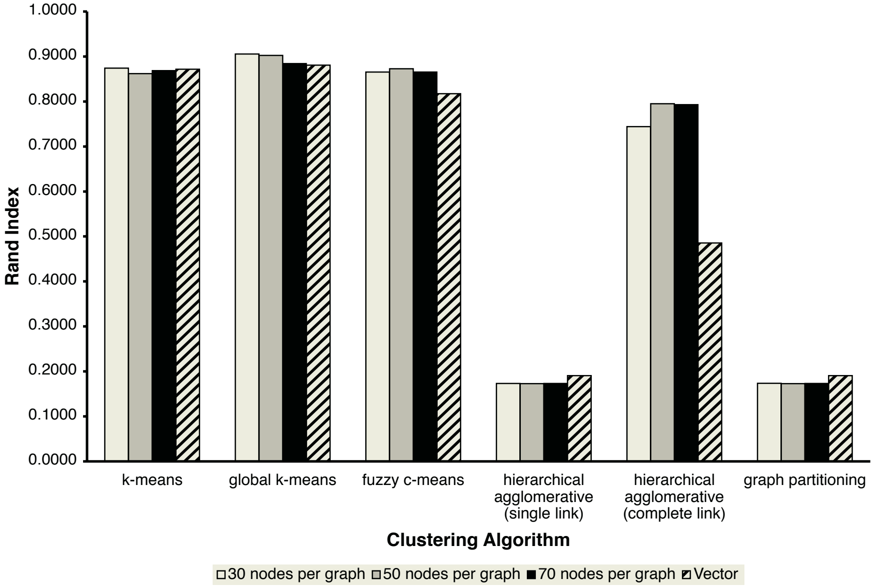

Fig. 3. Experimental results for the J-series data set.

distance measure based on Jaccard similarity [3], which was the best performing in our experiments. Nondeterministic clustering algorithms that use random initializations (k-means and fuzzy c-means) are represented by the average of ten experiments. The results indicate that the single link hierarchical clustering algorithm and the graph partitioning algorithm both performed poorly for all graph sizes and both data sets. Their similar performance is not surprising, as both methods take a similar approach of examining pairs of minimum distance objects; the hierarchical agglomerative algorithm can be seen as a bottom-up procedure while the graph partitioning method is its top-down counterpart. Both of these algorithms can lead to a "chaining effect" where most objects are placed in one large cluster with the other clusters containing only one or a few objects each. Complete link hierarchical agglomerative clustering does not suffer from this phenomenon, and thus its performance is considerably improved over the case of single link. Global k-means was the best performing algorithm overall; it only performed worse than other methods for the F-series data set when using 50 nodes per graph. The graph sizes we selected did not have a consistent influence across algorithms or data sets.

In comparing the performance of the graph-based methods to the traditional vector-based clustering, we see that in most cases clustering with data represented by graphs outperformed the clustering produced with a vector representation for the same clustering algorithm. Only for the single link hierarchical agglomerative clustering and graph partitioning algorithms did the vector approach perform better than all the graph-based experiments in the group; however, the margin of improvement was slight and clustering performance was still poor for all approaches for these two algorithms. The graph clustering approach strongly outperformed the vector model for the complete link hierarchical agglomerative clustering algorithm for both data sets.

4 Conclusions

In this paper we compared the performance of several popular clustering algorithms when using data represented by graphs rather than other conventional representation models, such as vectors. The novelty of this work is utilizing classical clustering algorithms, such as k-means or hierarchical agglomerative clustering, for clustering graph-based data. We compared six algorithms in all: k-means, global k-means, fuzzy c-means, hierarchical agglomerative clustering (single link and complete link), and graph partitioning. We used the Rand Index to measure how well the produced clusters matched ground truth when clustering two web document data sets. The results showed our graph approach outperformed the traditional vector representation methods for most clustering algorithms, strongly for the case of the complete link hierarchical agglomerative clustering algorithm.

Acknowledgments

This work was supported in part by the National Institute for Systems Test and Productivity at the University of South Florida under the U.S. Space and Naval Warfare Systems Command Contract No. N00039–02–C–3244 and by the Fulbright Foundation that has granted Prof. Kandel the Fulbright Research Award at Tel-Aviv University, College of Engineering during the academic year 2003–2004.

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys **31** (1999) 264–323
2. Mitchell, T.M.: Machine Learning. McGraw-Hill, Boston (1997)
3. Salton, G.: Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, Reading (1989)
4. Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall, Upper Saddle River (1995)
5. Zahn, C.T.: Graph-theoretical methods for detecting and describing gestalt structures. IEEE Transactions on Computers **C-20** (1971) 68–86
6. Likas, A., Vlassis, N., Verbeek, J.J.: The global k-means algorithm. Pattern Recognition **36** (2003) 451–461
7. Strehl, A., Ghosh, J., Mooney, R.: Impact of similarity measures on web-page clustering. In: AAAI-2000: Workshop of Artificial Intelligence for Web Search. (2000) 58–64

8. Zamir, O., Etzioni, O.: Web document clustering: A feasibility demonstration. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (1998) 46–54
9. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. Pattern Recognition Letters **19** (1998) 225–259
10. Fernández, M.L., Valiente, G.: A graph distance metric combining maximum common subgraph and minimum common supergraph. Pattern Recognition Letters **22** (2001) 753–758
11. Wallis, W.D., Shoubridge, P., Kraetz, M., Ray, D.: Graph distances using graph union. Pattern Recognition Letters **22** (2001) 701–704
12. Jiang, X., Muenger, A., Bunke, H.: On median graphs: properties, algorithms, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (2001) 1144–1151
13. Schenker, A., Last, M., Bunke, H., Kandel, A.: Clustering of web documents using a graph model. In Antonacopoulos, A., Hu, J., eds.: Web Document Analysis: Challenges and Opportunities. Volume 55 of Machine Perception and Artificial Intelligence. World Scientific Publishing Company (2003) 3–18
14. Schenker, A., Last, M., Bunke, H., Kandel, A.: Comparison of distance measures for graph-based clustering of documents. In: Proceedings of the 4th IAPR-TC15 International Workshop on Graph-Based Representations in Pattern Recognition. Volume 2726 of Lecture Notes in Computer Science., Springer-Verlag (2003) 202–213
15. Schenker, A., Last, M., Bunke, H., Kandel, A.: Graph representations for web document clustering. In: Proceedings of the 1st Iberian Conference on Pattern Recognition and Image Analysis. Volume 2652 of Lecture Notes in Computer Science., Springer-Verlag (2003) 935–942
16. Perner, P.: Data Mining on Multimedia Data. Volume 2558 of Lecture Notes in Computer Science. Springer-Verlag (2003)
17. Messmer, B.T., Bunke, H.: A new algorithm for error-tolerant subgraph isomorphism detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 493–504
18. Dickinson, P., Bunke, H., Dadej, A., Kretzl, M.: On graphs with unique node labels. In: Proceedings of the 4th IAPR-TC15 International Workshop on Graph-Based Representations in Pattern Recognition. Volume 2726 of Lecture Notes in Computer Science. Springer-Verlag (2003) 13–23
19. Tan, C.M., Wang, Y.F., Lee, C.D.: The use of bigrams to enhance text categorization. Information Processing and Management **38** (2002) 529–546
20. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association **66** (1971) 846–850