A Comparison of Two Novel Algorithms for Clustering Web Documents

Adam Schenker
University of South Florida, Department of Computer Science and Engineering
E-mail: aschenke@csee.usf.edu

Mark Last
Ben-Gurion University of the Negev, Department of Information Systems Engineering
E-mail: mlast@bgumail.bgu.ac.il

Horst Bunke
University of Bern, Department of Computer Science
E-mail: bunke@iam.unibe.ch

Abraham Kandel
University of South Florida, Department of Computer Science and Engineering
E-mail: kandel@csee.usf.edu

Abstract

In this paper we investigate the clustering of web document collections using two variants of the popular k-means clustering algorithm. The first variant is the global k-means method, which computes "good" initial cluster centers deterministically rather than relying on random initialization. The second variant allows for the use of graphs as fundamental representations of data items instead of the simpler vector model. We perform experiments comparing global k-means with random initialization using both the graph-based and the vector-based representations. Experiments are carried out on two web document collections and performance is evaluated using two clustering performance measures.

1. Introduction

The goal of clustering, a class of techniques that fall under the category of machine learning, is to automatically segregate data into groups called clusters. Clusters are collections of similar data items, and they can be created without prior training on labeled examples (unsupervised learning). A wide variety of clustering algorithms have appeared in the literature over the years, including techniques based on function optimization and hierarchical methods [4]. Applying clustering procedures to web document collections is of particular interest for several reasons. First, it can eliminate the need for manual organization, which can be costly for a large number of documents. Second, it can improve retrieval performance by constraining searches to certain clusters. Third, it allows document collections to be more easily browsed by users.

The *k*-means algorithm [7] is a popular method in unsupervised clustering that has also been used to cluster web documents [11]. Often when documents are clustered they are represented by vectors whose components indicate term frequency or importance [9]. The vector model is simple and naturally allows for operations that are easily performed in a Euclidean space, such as distance computation and cluster center calculation. However, it discards information which is inherent in the original documents, such as the order in which terms appear, where in the document they appear, and so forth. Recently an extension of the *k*-means algorithm which allows for documents to be represented by more robust graphs instead of vectors has been introduced [10]. Experimental results, which compared the graph-based method to the traditional vector methods, showed that the graph-based approach can outperform the vector approach by including this additional information.

In this paper we are interested in continuing the experiments relating to the graph-based *k*-means clustering algorithm by combining it with the global *k*-means method of Likas *et al.* [5]. Global *k*-means allows for the deterministic computation of "good" initial cluster centers. In the previous experiments with graph representations a series of random initializations was used. However, *k*-means is subject to becoming trapped at local extrema, thus some initial random states may lead to poor clustering performance. By combining global *k*-means with the graph-based approach, we hope to create a hybrid method with even better performance than either of the original methods. To test this new mixture of methods, we perform experiments with each possible combination: random with graphs, global *k*-means with graphs, random with vector, and global *k*-means with vector. We measure clustering performance over two web document data sets using two performance measures; in [10] only one web data set was examined, using a single performance measure. This also marks the first time global *k*-means will be applied to a web document collection; the experiments of [5] were carried out on the Iris data set, an image segmentation data set, and an artificial data set. Further, each of these data sets utilized a vector representation, thus this will be the first time graph-based data will be used with global *k*-means.

The remainder of our paper is organized as follows. We recount the *k*-means approach and describe the global *k*-means method in Section 2. We describe the graph-based version of *k*-means and the method used to represent web documents with graphs in Section 3. In Section 4 we will give experimental results which compare clustering performance of each of the method combinations. Conclusions are presented in Section 5.

2. The *k*-means algorithm

The *k*-means algorithm is a straightforward method for clustering data [7]. The basic procedure of the *k*-means method typically begins with assigning each data item to a random cluster. The number of clusters, *k*, is provided *a priori* by the user. Next, the cluster centers are calculated by finding the centroid of the data items in each cluster. After that, a new assignment of data items to clusters is computed by assigning them to their closest cluster center according to some distance measure. This process of computing cluster centers and then updating the cluster assignments is repeated until there is no change in the centroids. Under the vector-space model Euclidean distance is typically used as the distance measure, however in document clustering other distance measures such as cosine similarity or Jaccard similarity are often used due to length invariance or other properties [9].

Likas *et al.* [5] have recently introduced what they call the global *k*-means method. This procedure provides a way of determining "good" initial cluster centers for the *k*-means algorithm without having to use random initializations. Their experimental results have shown clustering performance under global *k*-means to be as good or better than using random initializations. The basic procedure is an incremental computation of cluster centers. Starting at the case of one cluster (*k*=1), the cluster center is defined to be the centroid of the entire data set. For the general case of *k* clusters, the centers are determined by taking the centers from the *k*-1 clusters problem and then determining the optimum location of a new center. This is accomplished by considering each data item as the new cluster center and then executing the *k*-means algorithm with that particular set of initial cluster centers and determining which one minimizes the error as defined by:

$$E(m_1,\ldots,m_M) = \sum_{i=1}^{N}\sum_{k=1}^{M} I(x_i \in C_k)\left\|x_i - m_k\right\|^2 \quad (1)$$

where N is the number of data items, M is the number of clusters, x_i is data item i, m_k is cluster center k, and $I(X)$ = 1 if X is true and 0 otherwise. A problem with this approach is that it requires execution of the *k*-means algorithm $O(N \cdot M)$ times. For many applications this will be too time-consuming. With this in mind, the authors have also proposed a "fast" version of global *k*-means. Under this method, instead of running *k*-means when considering each data item as a new cluster center candidate we calculate the following:

$$b_n = \sum_{j=1}^{N} \max(d_{k-1}^{j} - \left\|x_n - x_j\right\|^2, 0) \quad (2)$$

where d_{k-1}^{j} is the distance between data item x_j and its closest cluster center for the *k*-1 clusters problem. We then select the new cluster center to be data item x_i where:

$$i = \arg\max_{n} b_n \quad (3)$$

It is this "fast" version of the global *k*-means method that we have implemented and that we will use for our experiments in this paper.

3. Graph-based *k*-means

The basic *k*-means algorithm, as described in the previous section, has been extended to work with graphs instead of vectors [10]. In brief, this is accomplished using two techniques. First, for a distance measure a graph-theoretical distance measure based on the maximum common subgraph is used [2]:

$$d(x,y) = 1 - \frac{|mcs(x,y)|}{\max(|x|,|y|)} \quad (4)$$

Here x and y are graphs (not vectors), $mcs(x,y)$ is their maximum common subgraph, max(...) is the usual maximum value operation, and |...| indicates the size of a graph as defined by the number of nodes and edges in the graph. This distance measure is used in the *k*-means algorithm to determine the assignments of data items to clusters, where each item is placed with the cluster center with the minimum distance. To compute cluster centers, which are also graphs, the median of a set of graphs is used [1]. The median of a set of graphs is the graph from the set with minimum average distance to all the other graphs in the set; here distance is defined by a graph-theoretical distance measure such as Eq. (4). Thus we take the data items in each cluster, as determined by the cluster assignment step, and compute their center to be the median. For global *k*-means, we simply use the graph-theoretical distance measure of Eq. (4) in Eq. (2) and apply it as usual.

In order to benefit from the additional modeling capability of graphs, we need a method of representing the original data items (web documents, in this case) as graphs instead of vectors. We do this using the following process [10]. First, terms that occur on each document, with the exception of frequently occurring stop words such as "the" and "of" which provide little information, are extracted. We then apply a simple stemming algorithm and remove the most infrequent terms on each document leaving some fixed number of the most frequent terms, which are the most informative after stop word removal. Each term becomes a node in the graph representing the document. Each node is unique and only appears at most once in a graph (*e.g.* if the term "computer" appears twenty times in a document, there is only one node in the graph representing this term). When two terms are adjacent in the text of the document, we insert a directed edge from the node representing the former term to the node representing the latter. This edge is labeled with the section of the web document in which the adjacency occurs. Three document sections are defined: text (all readable text on the page), link (text in clickable hypertext links), and title (the document's title and any meta-data such as keywords).

This graph representation has an interesting effect on the complexity of the distance computation, Eq. (4). Usually the determination of the maximum common subgraph is NP-complete [6]. However, with unique node labels the complexity becomes $O(n^2)$, where n is the number of nodes in the graph. This is due to only having to match specific node labels in each graph and no longer needing to examine all possible node combinations. Thus the distances and median graph can be computed in polynomial time.

4. Experimental evaluation

In this section, we compare global k-means to random initialization using the graph-based representation and the traditional vector-space representation of web documents. We will evaluate clustering performance in our experiments using the following two clustering performance measures. These indices measure the matching of clusters computed by each method to the "ground truth" clusters, meaning they measure how close each clustering method is to the "correct" clustering that would be produced manually by a human. Other clustering performance measures rely on distances between data items. This can give us an indication of cluster compactness and separation, which is useful when ground truth is not available. However, they do not tell us if the clustering is correct. The first performance index is the *Rand index* [8]. The Rand index is computed by examining all pairs of objects in the data set after clustering. If two objects are in the same cluster in both the ground truth clustering and the clustering we wish to measure, this counts as an agreement. If two objects are in different clusters in both the ground truth clustering and the clustering we wish to measure, this is also an agreement. Otherwise, there is a disagreement. The Rand index is computed by dividing the number of agreements by the sum of agreements and disagreements. Thus the Rand index is a measure of how closely the clustering created by some procedure matches ground truth (*i.e.* it is a measure of clustering accuracy). It produces a value in the interval [0,1], with 1 representing a clustering that perfectly matches ground truth.

The second performance criterion we use is *mutual information* [3][11], which is an information-theoretic measure that compares the overall degree of agreement between the clustering under consideration and ground truth, with a preference for clusters that have high purity (*i.e.* are homogeneous with respect to the classes of objects clustered as given by ground truth). We omit the details of this method for brevity. Higher values of mutual information indicate better performance.

Table 1. Results for F-series (Rand index)

	Global k-means		Random	
Graph Size	Vector	Graphs	Vector	Graphs
10	0.7057	0.7281	0.6899	0.6730
20	0.7057	0.7976	0.6899	0.7192
30	0.7057	0.7838	0.6899	0.7394

Table 2. Results for F-series (mutual information)

	Global k-means		Random	
Graph Size	Vector	Graphs	Vector	Graphs
10	0.1914	0.1653	0.102	0.1498
20	0.1914	0.2274	0.102	0.1638
30	0.1914	0.2336	0.102	0.1793

Table 3. Results for J-series (Rand index)

	Global k-means		Random	
Graph Size	Vector	Graphs	Vector	Graphs
10	0.8809	0.9049	0.8717	0.8689
20	0.8809	0.9065	0.8717	0.8819
30	0.8809	0.9056	0.8717	0.8758

Table 4. Results for J-series (mutual information)

	Global k-means		Random	
Graph Size	Vector	Graphs	Vector	Graphs
10	0.2787	0.3048	0.2316	0.2393
20	0.2787	0.3135	0.2316	0.2597
30	0.2787	0.3188	0.2316	0.2447

We performed our clustering experiments on two web data sets, called the F-series and J-series (available at ftp://ftp.cs.umn.edu/dept/users/boley/PDDPdata/). The F-series consists of 93 web documents from four classes, while the J-series contains 185 documents and ten classes. We slightly altered the F-series due to the fact that there were conflicting multiple classifications for some documents; the original data set contained 98 documents and 17 sub-classes of four major classes. We use these two collections for several reasons. First, the original HTML documents are available, which is necessary for constructing the graph representations. Some document collections provide only a term–document matrix (vector representation). Second, ground truth assignments are provided for the documents; other web data sets are created with other tasks in mind, such as rule induction or prediction, and may not include this information. Finally, these data sets are of a manageable size in terms of both the number of documents and the number of clusters.

In our experiments we look at clustering documents that are represented by graphs that have 10, 20 or 30 maximum nodes per graph. The selection of the number of nodes comes from prior experimental results with these data sets; the optimum graph size generally depends on the size and nature of the data set. Graphs are created from the original web documents using the procedure described above in Section 3. For the vector-space experiments we use pre-created term–document matrices which are supplied at the site where the web document collections are hosted; the vectors of the F-series data set have 332 dimensions while those of the J-series have 474. We use a distance based on the Jaccard similarity [9] for the vector model, which was the best performing of the vector distance measures we have worked with.

The results of our experiments are presented in Tables 1 to 4 for values of k equal to the number of clusters present in ground truth (k=4 for the F-series, k=10 for the J-series). Here random denotes the average of ten experiments each using a random initialization. The results show that in all cases, whether graph or vector related, the global k-means method consistently outperformed the corresponding random method. The results also show that, in five out of the eight cases when using the minimum number of nodes per graph (10), the graph-based method outperformed the vector method for both random and global k-means for both data sets.

Table 5. Execution times using random initialization (in seconds)

	Random (average of 10 experiments)			
	Graphs – 10	Graphs – 20	Graphs – 30	Vector
F-series	84.4	126.1	205.3	24.5
J-series	173.1	396.4	550.2	214.9

Table 6. Execution times using global *k*-means (in minutes)

	Global *k*-means			
	Graphs – 10	Graphs – 20	Graphs – 30	Vector
F-series	11.87	24.88	38.68	14.57
J-series	239.55	545.92	818.47	507.55

The execution times for the experiments are also given in Tables 5 and 6. All experiments were carried out on the same system under the same operating conditions: an unloaded 296 MHz Sun UltraSPARC-II with 1,024 megabytes of memory. As expected, the execution time for global k-means is much greater than random, due to the need to compute the initial cluster centers. However, the initial cluster centers for a data set (or a subset of one), once computed, can be re-used for incremental clustering without incurring an additional performance penalty. This can be useful for data sets that are dynamic or need to be re-clustered frequently. We see the potential for a time savings over the vector case when using small graphs. For the J-series using global k-means, the graph-based method with a maximum graph size of 10 nodes was not only better performing than the vector case, it was faster by nearly four and a half hours.

5. Conclusions

In this paper we have examined the global k-means method by combining it with our extension of the k-means algorithm that allows for web documents to be represented by graphs, which are more robust than vectors. We performed experiments on two web document data sets and measured agreement with ground truth using the Rand index and mutual information. The results consistently show a clear improvement when using global k-means over random initialization for both the graph-based approach and the traditional vector-space approach for both data sets and both performance measures. Combining global k-means with our graph-based algorithm, which had already shown an improvement over the usual vector approach, yielded the best results in all but one case (out of 12). This case occurred only for mutual information of one data set, when we allowed the maximum dimensionality reduction used in our experiments of 10 nodes per graph.

6. Acknowledgments

This work was supported in part by the National Institute for Systems Test and Productivity at the University of South Florida under U.S. Space and Naval Warfare Systems Command grant number N00039–01–1–2248.

References

[1] X. Jiang, A. Muenger, and H. Bunke, "On median graphs: properties, algorithms, and applications", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 10, 2001, pp. 1144–1151.

[2] H. Bunke and K. Shearer, "A graph distance metric based on the maximal common subgraph", *Pattern Recognition Letters*, Vol. 19, 1998, pp. 255–259.

[3] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley, 1991.

[4] A. K. Jain, M. N. Murty and P. J. Flynn, "Data clustering: a review", A*CM Computing Surveys*, Vol. 31, No. 3, 1999, pp. 264–323.

[5] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means algorithm", *Pattern Recognition*, Vol. 36, 2003, pp. 451–461.

[6] B. T. Messmer and H. Bunke, "A new algorithm for error-tolerant subgraph isomorphism detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 5, 1998, pp. 493–504.

[7] T. M. Mitchell, *Machine Learning*, McGraw–Hill, Boston, 1997.

[8] M. Rand, "Objective criteria for the evaluation of clustering methods", *Journal of the American Statistical Association*, Vol. 66, 1971, pp. 846–850.

[9] G. Salton, *Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer*, Addison–Wesley, Reading, 1989.

[10] A. Schenker, M. Last, H. Bunke, and A. Kandel, "Clustering of Web Documents Using a Graph Model", *Web Document Analysis: Challenges and Opportunities*, eds. A. Antonacopoulos and J. Hu, to appear.

[11] A. Strehl, J. Ghosh, and R. Mooney, "Impact of similarity measures on web-page clustering", *AAAI–2000: Workshop of Artificial Intelligence for Web Search,* 2000, pp. 58–64.