Extended Star Clustering Algorithm

Reynaldo J. Gil-García[1], José M. Badía-Contelles[2], and Aurora Pons-Porrata[1]

[1] Universidad de Oriente, Santiago de Cuba, Cuba
{gil,aurora}@app.uo.edu.cu
[2] Universitat Jaume I, Castellón, Spain
badia@icc.uji.es

Abstract. In this paper we propose the extended star clustering algorithm and compare it with the original star clustering algorithm. We introduce a new concept of star and as a consequence, we obtain different star-shaped clusters. The evaluation experiments on TREC data, show that the proposed algorithm outperforms the original algorithm. Our algorithm is independent of the data order and obtains a smaller number of clusters.

1 Introduction

Clustering algorithms are widely used for document classification, clustering of genes and proteins with similar functions, event detection and tracking on a stream of news, image segmentation and so on. For a good overview see [1,2]. Given a collection of n objects characterized by m features, clustering algorithms try to construct partitions or covers of this collection. The similarity among the objects in the same cluster should be maximum, whereas the similarity among objects in different clusters should be minimum.

One of the most important problems in recent years is the enormous increase in the amount of unorganized data. Consider, for example, the web or the flow of news in newspapers. We need methods for organizing information in order to highlight the topic content of a collection, detect new topics and track them. The star clustering algorithm [3] was proposed for these tasks and three scalable extensions of this algorithm are presented in [4]. The star method outperforms existing clustering algorithms such as single link [5], average link [6] and k-means [7] in the organizing information task as it can be seen in [3]. However, the clusters obtained by this algorithm depend on the data order and it could obtain "illogical" clusters.

In this paper we propose a new clustering method that solves some of its drawbacks. We define a new concept of star and as a consequence, we obtain different star-shaped clusters. Both algorithms were compared using TREC data and the experiments show that our algorithm outperforms the original star clustering algorithm.

The rest of the paper is organized as follows. Section 2 describes the star clustering algorithm and shows its drawbacks. Section 3 describes the proposed algorithm and the experimental results are shown in Section 4. Finally, conclusions are presented in Section 5.

A. Sanfeliu and J. Ruiz-Shulcloper (Eds.): CIARP 2003, LNCS 2905, pp. 480–487, 2003.

2 Star Clustering Algorithm

The star algorithm is different to the Scatter-Gather [8] and Charikar algorithm [9], because it does not impose a fixed number of clusters as a constraint on the solution. Besides, it guarantees a lower bound on the similarity between the objects in each cluster if the space of representation has metric properties. The clusters created by the algorithm can be overlapped. This is a desirable feature in the organization information problems, since documents can have multiple topics.

Two objects are β_0-similar if their similarity is greater or equal to β_0, where β_0 is a user-defined parameter. We call β_0-similarity graph the undirected graph whose vertices are the objects to cluster and there is an edge from vertex o_i to vertex o_j, if o_j is β_0-similar to o_i. Finding the minimum vertex cover of a graph is a NP complete problem. This algorithm is based on a greedy cover of the β_0-similarity graph by star-shaped subgraphs. A star-shaped subgraph of $l+1$ vertices consists of a single *star* and l *satellite vertices*, where there exist edges between the star and each of the satellite vertices. The stars are the objects with highest connectivity. The isolated objects in the β_0-similarity graph are also stars. The algorithm guarantees a pairwise similarity of at least β_0 between the star and each of the satellite vertices, but such similarity is not guaranteed between satellite vertices. Another characteristic of this algorithm is that two stars are never adjacent.

The star algorithm stores the neighbors of each object in the β_0-similarity graph. Each object is marked as star or as satellite. The main steps of the algorithm are shown in Algorithm 1.

Algorithm 1 Star clustering algorithm.

```
Calculate all similarities between each pair of objects to
construct the $\beta_0$-similarity graph
Let $N(o)$ be the neighbors of each object $o$ in the $\beta_0$-similarity
graph
Let each object $o$ initially be unmarked
Sort the objects by degree $|N(o)|$
While an unmarked object exists:
    Take the highest degree unmarked object $o$
    Mark $o$ as star
    For $o'$ in $N(o)$:
        Mark $o'$ as satellite
For each object $o$ marked as star:
    Add a new cluster $\{o\} \cup N(o)$
```

The complexity time of the algorithm is $O(n^2m)$, since it must calculate the similarities between all objects, each with m features.

The star clustering algorithm has some drawbacks. First, the obtained clusters depend on the order of the objects. If two or more neighbor objects with the same degree exist, only the first of them in the arrangement is a star. This problem is illustrated with the help of Figure 1, where the dark circles are the obtained stars and the clusters are outlined. In the figure on the left, the star algorithm takes first the object B, thus B is star and A, C are satellites. However, if object A (or C) is the first in the arrangement, the algorithm obtains the clusters shown in the figure on the right. As we can see, the obtained clusters are different.

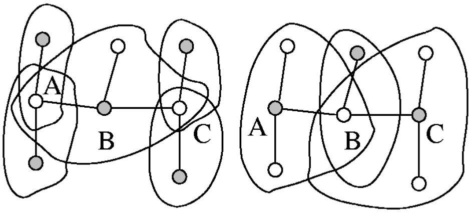

Fig. 1. Dependency of the data order.

The second main drawback of the star algorithm is that it can produce illogical clusters. Since two stars are never neighbors, the illogical clusters could be obtained. Figure 2 shows this problem. Object B should be a star and its neighbors with less degree should not be stars.

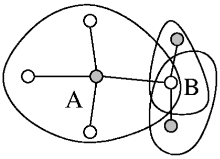

Fig. 2. Illogical clusters.

3 Extended Star Clustering Algorithm

In our algorithm we make two main changes with respect to the star clustering algorithm mentioned above. The complement degree of an object o is the degree of o taking into account its neighbors not included yet in any cluster, namely:

$$CD(o) = |N(o) \setminus Clu|$$

where Clu is the set of objects already clustered. As we can see, the complement degree of an object decreases during the clustering process as more objects are included in clusters.

Besides, an object o is considered a star if it has at least a neighbor o' with less or equal degree than o that satisfies one of the following conditions:

- o' has not a star neighbor.
- The highest degree of the stars that are neighbors of o' is not greater than the degree of o.

It is worth mentioning that these conditions are necessary but not sufficient. That is, some objects that satisfy the previous conditions could not be selected as stars by the algorithm.

The main steps of our algorithm are shown in Algorithm 2.

Algorithm 2 Extended star clustering algorithm.

```
Calculate all similarities between each pair of objects to
construct the $\beta_0$-similarity graph
Let $N(o)$ be the neighbors of each object $o$ in the $\beta_0$-similarity
graph
For each isolated object $o$ ($|N(o)| = 0$):
    Create the singleton cluster $\{o\}$
Let $L$ be the set of non-isolated objects
Calculate the complement degree of each object in $L$
While a non-clustered object exists: (*)
    Let $M_0$ be the subset of objects of $L$ with maximum
    complement degree
    Let $M$ be the subset of objects of $M_0$ with maximum degree
    For each object $o$ in $M$:
        If $o$ satisfies the condition to be a star, then
            If $\{o\} \cup N(o)$ does not exist:
               Create a cluster $\{o\} \cup N(o)$
    Delete the processed objects from $L$ (**)
    Update the complement degree of the objects in L
```

In the step (**) we can delete from L the objects in M or all the objects already clustered. We named these variations the unrestricted and restricted versions of the algorithm. In the restricted version, only the objects not yet clustered, can be star. Each version has advantages and disadvantages. The unrestricted version has more possibilities to find the best stars. The restricted version is faster but can obtain illogical clusters.

The complexity time of the algorithm is $O(n^2m)$ and is determined by its first step: the calculation of the similarities between objects. The complexitiy of the cycle (*) is $O(n^2)$. In the worst case, this cycle is repeated $\log_2 n$ times. The most expensive step in it, is the update of the complement degree of the objects, which is $O(\frac{n^2}{\log_2 n})$ if we have a table containing the differences between all combination of blocks of bits with a size $\log_2 n$. This table is not needed if

$\log_2 n$ is less than or equal to the word of the computer processor, because in this case the difference operation between blocks of bits is $O(1)$.

The proposed algorithm creates overlapped clusters and guarantees a pairwise similarity β_0 between the star and its neighbors. Unlike the original star clustering algorithm, the obtained clusters are independent of the data order. If two or more objects with the highest connectivity exist, our algorithm select as stars all of them. Besides, the selection of stars using the complement degree allows to cover quickly the data and it reduces the overlapping among the clusters.

The extended star clustering algorithm solves the problems of the original star clustering algorithm cited on section 2. An object is considered a star even if it has a star neighbor. Moreover, the second condition in the new star concept guarantees that if an object is a neighbor of two possible stars, both will be stars. Thus, it is independent of the order of objects.

Figure 3a) shows the stars obtained by our algorithm in the same case that in Figure 1.

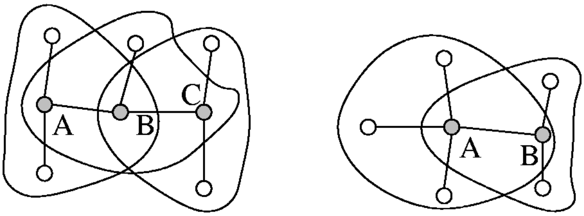

Fig. 3. Solutions: a) Order problem. b) Illogical clusters.

The unrestricted version of our algorithm does not form illogical clusters because it allows neighbor stars. Figure 3b) shows the stars obtained in the same case that in Figure 2 for the unrestricted version.

4 Experimental Results

In order to evaluate the performance of our algorithm, we compared it with the original star clustering algorithm. We used data (in Spanish) from the TREC-4 and TREC-5 conferences as our testing medium [11]. The TREC-4 collection contains a set of "El Norte" newspaper articles in 1994. This collection has 5828 articles classified in 50 topics. The TREC-5 consists of articles from AFP agency in 1994-1996 years, classified in 25 topics. We have only the data from 1994, for a total of 695 classified articles.

The documents are represented using the traditional vectorial model. The terms of documents represent the lemmas of the words appearing in the texts. Stop words, such as articles, prepositions and adverbs are disregarded from the document vectors. Terms are statistically weighted using the normalized term frequency (TF). Moreover, we use the traditional cosine measure to compare the documents.

To evaluate the quality of our algorithm, we partitioned the "El Norte" collection in five sub-collections. Each sub-collection is composed of articles related to 10 distinct topics. So, if we add the AFP collection we have a total of 6 collections. The general characteristics of these collections are summarized in Table 1.

Table 1. Description of collections

Collection	# of documents	Topics	Collection	# of documents	Topics
eln-1	1534	SP1-SP10	eln-4	811	SP31-SP40
eln-2	1715	SP11-SP20	eln-5	829	SP41-SP50
eln-3	1732	SP21-SP30	afp	695	SP51-SP75

To evaluate the clustering results, we use the F1-measure [12]. This measure compares the system-generated clusters with the manually labeled topics. It is widely applied in Information Retrieval Systems, and it combines the precision and recall factors. The F1-measure of the cluster number j with respect to the topic number i can be evaluated as follows:

$$F1(i,j) = 2\frac{n_{ij}}{n_i + n_j}$$

where n_{ij} is the number of common members in the topic i and the cluster j, n_i is the cardinality of the topic i, and n_j is the cardinality of the cluster j.

To define a global measure, first each topic must be mapped to the cluster that produces the maximum F1-measure:

$$\sigma(i) = \max_{j} \{F1(i,j)\}$$

Hence, the overall F1-measure is calculated as follows:

$$F1 = \frac{1}{S}\sum_{i=1}^{N} n_i F1(i,\sigma(i)), \;\; S = \sum_{i=1}^{N} n_i$$

where N is the number of topics.

In our experiments we compare the original star clustering algorithms with the unrestricted and restricted versions of the proposed algorithm. Table 2 shows the best F1-measure obtained by the algorithms for optimized values of β_0 in the 6 collections. As we can see, both versions of our algorithm outperform the original star algorithm in all of these collections except in eln-5. Besides, in most collections our algorithm obtain less clusters than the original star algorithm.

This is another important result, because our algorithm achieves a greater precision.

If we compare the results obtained by the original star algorithm with the restricted version of our algorithm, we can see the effect of using the complement degree and the new star concept. The restricted version always has better or equal F1 values with smaller quantity of clusters. The unrestricted version outperforms the restricted one in three cases whereas it has smaller performance in one case. We can expect that the unrestricted version has the best performance in most cases, hence if the main goal is to obtain the best clusters, we recommend the unrestricted version of the algorithm. On the other hand, if we want to reduce the number of clusters or the execution time, we should use the restricted version of the algorithm.

Table 2. Experimental results

Algorithm	Original Star		Extended Star (restricted)		Extended Star (unrestricted)	
Collection	F1	# of clusters	F1	# of clusters	F1	# of clusters
afp	0.76	136	0.77	99	0.78	105
eln-1	0.61	139	0.63	59	0.63	81
eln-2	0.66	62	0.67	59	0.72	61
eln-3	0.53	109	0.53	52	0.59	67
eln-4	0.55	62	0.58	68	0.58	43
eln-5	0.74	57	0.74	48	0.72	59

5 Conclusions

In this paper we presented a new clustering algorithm, named the extended star clustering algorithm. We use the complement degree of an object and we define a new concept of star. As a consequence, we obtain different star-shaped clusters. Our algorithm solves the problems of dependency of data order and illogical clusters of the original star algorithm.

We compare the proposed algorithm with the original star algorithm in several collections of TREC data. Our algorithm obtains a better performance in these collections and produces less clusters.

The new algorithm can be used in tasks such as information organization, browsing, topic tracking and new topic detection. Besides, our algorithm can be useful in other areas of Pattern Recognition.

As a future work we will construct a parallel version of our algorithm to process very large data sets.

References

1. Jain, A.K.; Murty, M.N. and Flynn, P.J.: Data Clustering: A Review. *ACM Computing Surveys*, Vol. 31, No. 3, September 1999.

2. Berkhin, P.: Survey of Clustering Data Mining Techniques, Technical Report, Accrue Software, 2002.
3. Aslam, J.; Pelekhov, K. and Rus, D.: Static and Dynamic Information Organization with Star Clusters. In *Proceedings of the 1998 Conference on Information Knowledge Management*, Baltimore, MD, 1998.
4. Aslam, J.; Pelekhov, K. and Rus, D.: Scalable Information Organization. In *Proceedings of RIAO*, 2000.
5. Croft, W. B.: Clustering large files of documents using the single-link method. *Journal of the American Society for Information Science*, pp. 189-195, November 1977.
6. Voorhees, E. M.: Implementing agglomerative hierarchical clustering algorithms for use in document retrieval. *Information Processing and Management*, 22:465—476, 1986.
7. Mc Queen, J.: Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.
8. Cutting, D.; Karger, D. and Pedersen, J.: Constant interaction-time Scatter/Gather browsing of very large document collections. In *Proceedings of the 16th SIGIR*, 1993.
9. Charikar, M.; Chekuri, C.; Feder, T. and Motwani, R.: Incremental clustering and dynamic information retrieval. In *Proceedings of the 29th Symposium on Theory of Computing*, 1997.
10. Cormer, T. H.; Leiserson, C. E.; Rivest, R. L. Introduction to Algorithms. McGraw-Hill, 1993.
11. `http://trec.nist.gov`
12. Larsen, B. and Aone, C.: Fast and Effective Text Mining Using Linear-time Document Clustering. In *KDD'99*, San Diego, California, pp. 16–22, 1999.