Document Clustering - an Optimization Problem

Ao Feng
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
aofeng@cs.umass.edu

ABSTRACT

Clustering algorithms have been widely used in information retrieval applications. However, it is difficult to define an objective "best" clustering result. This article analyzes some document clustering algorithms and illustrates that they are equivalent to the optimization problem of some global functions. Experiments show the good performance of these algorithms, but there are still counter-examples where they fail to return the global optimum. We argue that Monte-Carlo methods in the global optimization framework have the potential to find better solutions than traditional clustering, and they are able to handle more complex structures.

1. INTRODUCTION

For a text collection with a large number of unlabeled documents, the commonly-used analysis is to run a clustering process based on the proximity among these elements. Documents in a cluster are very likely to match the same information need [2]. There are many clustering algorithms [1], but evaluation of the cluster quality is still a major concern. Different applications have various performance measures, most of which involve manual annotation of "truth" data. How to define "optimal" clusters without any subjective relevance judgment is still an open problem.

In this article, we will show that clustering algorithms are actually the optimization of some global functions. Section 2 introduces a few algorithms and defines their corresponding functions. Section 3 compares the performance of a clustering algorithm and a Monte-Carlo optimization method. Analysis and further extension of the framework are introduced in Section 4.

2. CLUSTERING ALGORITHMS

There are mainly two types of clustering algorithms, hierarchical and partitional. In this section, we will select one representative method from each and show that it corresponds to a global optimization problem. We believe that other algorithms have their own global functions as well.

2.1 Hierarchical Clustering

With a pair-wise similarity matrix of n documents, hierarchical agglomerative clustering (HAC) starts from n singleton clusters (each containing exactly one document). In each round, the most similar cluster pair is merged, and the process goes on until the highest similarity falls below the preset clustering threshold. There are three common options for the similarity calculation between two clusters: complete link, average link or single link; average link usually performs better than the other two.

Since only the similarity matrix is available, the goal for clustering is to group together documents that have high similarities, and keep the non-similar documents apart. If we formulate a relation matrix R where $R_{ij} = 0$ when documents i and j are in the same cluster and -1 otherwise, a global score function can be defined as

$$S = \sum_{1 \le i,j \le n}^{i \ne j} score(i, j, R_{ij}) \tag{1}$$

$$score(i, j, R_{ij}) = \begin{cases} c & \text{if } R_{ij} = -1 \\ sim(d_i, d_j) & \text{if } R_{ij} = 0 \end{cases} \tag{2}$$

here d_i, d_j are documents, and c is a constant.

If c is the same as the clustering threshold, the global score S is always increased in the process of HAC. Suppose that C_k and C_l are the most similar clusters, and

$$sim(C_k, C_l) = \frac{\sum_{d_i \in C_k, d_j \in C_l} sim(d_i, d_j)}{|C_k||C_l|} > c \tag{3}$$

After merging them, the change in S is

$$S' - S = \sum_{d_i \in C_k, d_j \in C_l} sim(d_i, d_j) - c|C_k||C_l| > 0 \tag{4}$$

In a divisive clustering algorithm, the basic operation is to split a cluster from an edge with low across-edge similarity. If the average similarity is smaller than c, it also raises S.

2.2 Partitional Clustering

K-means is an algorithm that clusters objects in a vector space into k partitions. It starts with k initial cluster centroids and assigns each object to its closest one. In each round, the centroids are recalculated and objects are reassigned. It keeps running until the clusters converge.

The process of K-means does not indicate any global objective, but it is actually trying to minimize the intra-cluster

Submission for ACM Conference on Research and Development in Information Retrieval, 2007. Please do not cite or distribute this draft paper.

variance.

$$V = \sum_{i=1}^{k} \sum_{x_j \in C_i} |x_j - \mu_i|^2 \quad (5)$$

here x_j is an object, C_i is a cluster and μ_i is its centroid.

3. EXPERIMENTS

In the previous section, we have described two clustering algorithms and shown their corresponding global functions. A natural question is, are they guaranteed to return the best solution in the global optimization framework?

3.1 Evaluation

There are two ways to evaluate the clustering output. With the global functions defined above, we can calculate the objective: values closer to the optimum mean better results. The other is to compare system-generated clusters to some "truth" data, where a better match gets higher score. The latter often depends on the quality of the manual annotation, and annotators may have different opinions.

Assume that we randomly select two objects x_i and x_j, each of them will be assigned to some cluster in the system output and in the truth data, respectively. If they have the same membership status in both cases, it is regarded as a successful case for the system.

$$\begin{aligned} precision = &\quad P(C(x_i) = C(x_j) | C'(x_i) = C'(x_j)) \\ recall = &\quad P(C'(x_i) = C'(x_j) | C(x_i) = C(x_j)) \\ F1 = &\quad \frac{2 \times precision \times recall}{precision + recall} \end{aligned} \quad (6)$$

here $C(x_i)$ is the cluster x_i belongs to in the truth data, and $C'(x_i)$ is its cluster in system output.

3.2 Implementation of HAC

The collection used in the experiments is part of TDT-3[1]. Six topics are selected from the same scenario (science/discovery), with a total of 280 news stories. The similarity matrix is calculated with tf-idf, where the term vectors are built based on the body part of stories.

The only parameter in HAC is the clustering threshold, and we tune it to maximize $F1$ defined in Equation 6. The optimal threshold is 0.09 from the experiment.

Simulated annealing (SA) is implemented for the optimization problem in Equation 1. It also starts with a list of singleton clusters (R is all -1 except for the diagonal elements that are 0). In each round, a story d_i is randomly picked, and another story d_j is selected based on the probability distribution of R_{i*}. If $R_{ij} = 0$, the cluster they are in will be broken into two. If $R_{ij} = -1$, the clusters of d_i and d_j will be merged. Whenever S increases after a round, the change is kept. Otherwise, it is kept with some probability.

Since SA does not have deterministic output, we run it 20 times, and the best runs are shown in Table 1. HAC achieves better performance on both $F1$ and S, but SA runs are fairly close to it.

In the experiment, HAC gets higher S than all SA runs. Does that mean HAC will always find the optimal solution? Figure 1 shows a counter-example. HAC combines 1 and 2 first since they are the most similar pair, then no other nodes can be merged because the average similarity cannot exceed 0.4. However, the best clusters are {1,3} and {2,4}, which gets 2.6 instead of 2.55 for S. On the other hand, Monte-Carlo algorithms are more likely to find the ideal solution, since they have some chance to get out of local maxima.

[1]Available from the linguistic data consortium (LDC), catalog number LDC2001T58.

Algorithm	HAC	SA-best S	SA-best $F1$
precision	0.330	0.267	0.307
recall	0.540	0.535	0.555
$F1$	0.410	0.356	0.395
S	3787.6	3779.1	3773.1

Table 1: Performance of HAC and SA

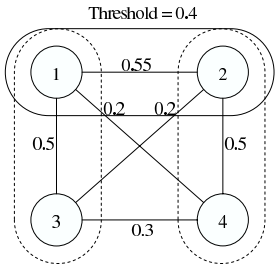

Figure 1: Counter-example: HAC does not get the optimal solution

3.3 Analysis of K-means

The result of K-means is greatly dependent on the initial selection of centroids. Even with deterministic process, different starting states lead to different results. Sometimes the solution after convergence can be much worse than the global optimum, indicating that there are local minima in the topology of the intra-cluster variance V.

4. CONCLUSIONS

With two candidate algorithms, we have shown that many clustering methods correspond to the optimization of some objective global functions, and they often fail to return the optimal solution. Experiments with HAC show that SA in the global optimization framework can achieve at least close performance to the deterministic algorithm, and it has the potential to find better results.

Another advantage of the global optimization framework is that it can model more complex relations. For example, news stories contain contextual information, which shows logical, temporal or spatial links among reports, in addition to the term similarity that is used to cluster documents. Optimizing multiple relations together is likely to yield better results than a traditional clustering algorithm for such applications.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of

the sponsor.

5. REFERENCES

[1] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

[2] C. J. van Rijsbergen and K. S. Jones. A test for the separation of relevant and non-relevant documents in experimental retrieval collections. *Journal of Documentation*, 29(3):251–257, 1973.