Clustering in Hyper-Linked Document Database using Efficient Graph Algorithm

Win Lai Lai Naing, Thin Thin Htike
University of Computer Studies, Pathein
nainglay88@gmail.com

Abstract

Clustering is an essential data mining task with numerous applications. Clustering is the process of grouping the data into classes or clusters, so that objects within a cluster have high similarity in comparison to one another but are very dissimilar to objects in other clusters. This system uses efficient graph clustering algorithm to group online scientific literature. The pages and hyperlinks of the World-Wide Web may be viewed as nodes and edges in a directed graph. Our approach to clustering uses the citation patterns of the CiteSeer database to form previously established clusters (soft clusters). The soft clusters, in turn, can be compared to one another in terms of the papers that they have in common. Similar soft clusters are merged by Ward's agglomerative hierarchical clustering method. In the end we find the collections of documents that are all related to one another by their citation patterns. By approaching in this manner, we can rapidly calculate clusters for datasets with tens of thousands of documents.

1. Introduction

Over the past decade, the World Wide Web has become an increasingly popular medium for publishing scientific literature. While the web is rich with information about the progress of science, gathering and making sense of this data is difficult because publications on the web are largely unorganized.

One of the key issues in information retrieval of data in large and voluminous database is the design and implementation of an efficient and effective indexing structure for the data objects in the database. Without a properly designed indexing structure, the retrieval of information may be reduced to a linear exhaustive search. On the other hand, a good indexing structure will make the retrieval accurate and computationally efficient.

Clustering is a division of data into groups of similar objects. It has proved to be a particularly important tool in *information retrieval* for constructing taxonomy of a corpus of documents by forming groups of closely-related documents.

The idea of this paper is to use the citation graph that can be built from CiteSeer to find a collection of the entries by topics. Clustering that kind of data can be done in essentially two ways. One can either take a semantic-based approach or a graph-theory-based approach. In this thesis, we use a graph-theory-based approach.

2. Related Works

There are several ways to approach the automatic clustering of hypertext documents. Text-based clustering typically involves computing inter-document similarities based on content-word frequency statistics. Their clustering was then computed from inter-document similarities. Our clustering technique is based on the analysis of hypertext link topology. Unlike earlier link-topology techniques, co-citation analysis builds upon the notion that when a WWW document D contains links referring to documents A and B, then A and B are related in some manner in the mind of the person who produced the document. In this example, documents A and B are said to be co-cited.

In this section, we brief present some of the research literature related to clustering scientific literature. Chen and Carr [2] used ACM publication data to analyze the structure of hypertext literature, filtering out authors that were cited less than five times during the period of 1987 – 1998, resulting in 367 authors. Pitkow and Pirolli [4] used the concept of scientific publication citations to hypertext links on the web. 5,582 HTML and 15,139 non-HTML documents were considered and clustered using complete linkage hierarchical clustering at different citation frequency thresholds.

In comparison with these studies, the method we introduce here facilitates the analysis of much larger CiteSeer datasets (the dataset we analyze has close to an order of magnitude more documents than

the largest dataset from these studies), and addresses the issue of discriminating against newer publications.

3. CiteSeer Data

Once a clustering algorithm is developed, how it works should be tested by various data sets. In this sense, testing data sets play an important role in the process of algorithm development. Finding a good similarity function depends strongly on the dataset and it is not an easy task to define a "good" one.

We use the database of scientific literature created by CiteSeer [7, 8], which is available at http://csindex.com/. It is an automatically-built, freely available, online database of publications in the field of computer science. CiteSeer checks to verify that the document is a research document by testing for the existence of a reference or bibliography section.

```
<record>
<header>
<identifier>oai:CiteSeerPSU:4
</identifier>
</header>
<dc:title>Exploration Bonuses and Dual Control
</dc:title>
<oai_citeseer:author name="Peter Dayan">
<dc:date>1995-06-05</dc:date>
<dc:format>ps</dc:format>
<dc:identifier>
http://citeseer.ist.psu.edu/4.html
</dc:identifier>
<dc:language>en</dc:language>
<oai_citeseer:relation type="References">
<oai_citeseer:uri>oai:CiteSeerPSU:110654
</oai_citeseer:uri>
</oai_citeseer:relation>
<oai_citeseer:relation type="References">
<oai_citeseer:uri>oai:CiteSeerPSU:124093
</oai_citeseer:uri>
</oai_citeseer:relation>
</record>
```

Figure 1. Example of an Original Data Record from CiteSeer

CiteSeer provides three different sources of information:

- The HTML-based web site tailored for human use.
- The XML-based OAI interface tailored for harvesting.
- An archive of a snapshot of the OAI records.

The different data sources differ on their reliability and completeness. As aforementioned, the citation graph on which we are working is the one built from the archive of records from OAI as shown in Figure 1.

3.1. Citation Indexing from CiteSeer

A citation index is an index of citations between publications, allowing the user to easily establish which later documents cite which earlier documents.

It is originally designed mainly for literature search for researchers to find subsequent articles that cite a given article. There is a distinction between citation and reference as shown in Figure 2. If Paper R contains a bibliographic footnote using and describing Paper C, then

- R contains a *reference* to C,
- C has a *citation* from R

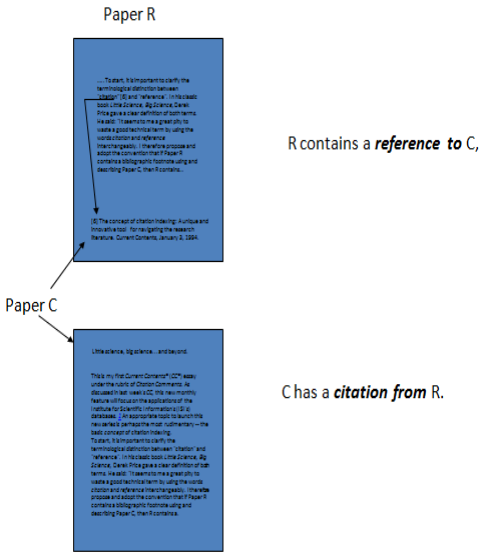

Figure 2. Distinction between Citation and Reference

The internal CiteSeer database represents documents and contexts (a list of documents in which they are referenced.) separately. Each document has a documentID and each context has a contextID. A *References* link from document A to document B means that document A cites document B. An *IsReferencedBy* link (citation) from document A to document B means that document B cites a context C such that the contextID of C is equal to the documentID of A.

Table 1. Example Database Table for Citation Indexing

(documentID)	(contextID)
oai:CiteSeerPSU:111	oai:CiteSeerPSU:222
oai:CiteSeerPSU:111	oai:CiteSeerPSU:333
oai:CiteSeerPSU:666	oai:CiteSeerPSU:111
oai:CiteSeerPSU:555	oai:CiteSeerPSU:111

For *References* link example, first define documentID oai:CiteSeerPSU:111 as document A, 222 as document B and 333 as document C shown

in Table 1. Thus document A contains references to document B and C. For *IsReferencedBy* link example, define 666 as document X and 555 as document Y. In Table 1, document X and Y's contextID is equal to document A, so that document A has citations from document X and Y.

4. Efficient Graph Clustering

The pages and hyperlinks of the World-Wide-Web may be viewed as nodes and edges in a directed graph. This graph is a fascinating object of study: it has several hundred million nodes today, over a billion links, and appears to grow exponentially with time. Ignoring the semantic information present in various HTML tags, a hypertext document has three different features: (1) the words contained in the document, (2) out-link, that is, the list of hypertext documents that are pointed to or cited by the document, and (3) the in-link, that is, the list of hypertext documents that point to or cite the document. If two documents share one or more links or in-links, then we consider them to be similar as well. This simple observation is the key to the present paper. The efficient graph clustering algorithm [1] is as shown in Figure 3.

```
1  procedure EFFICIENT-GRAPH-CLUSTER(graph: G = (V, E))
2     { Find most highly cited papers normalized by publications for year. }
3     C ← {}
4     for all {v ∈ V} do
5        norm ← number of papers published after v
6        norm-cite(v) ← |{(u, v) ∈ E}|/norm
7        if norm-cite(v) > threshold then
8           C ← C ∪ {v}
9        end if
10    end for
11
12    { Assign papers to soft cluster if they are co−cited. }
13    for all {v ∈ C} do
14       S_v ← {x : ∃_y (y, x) ∈ E ∧ (y, v) ∈ E}
15    end for
16
17    { Calculate similarity measure for all pairs c ∈ C . }
18    for all {x ∈ C} do
19       for all {y ∈ C} do
20          M_xy ← |S_x ∩ S_y|/(|S_x| + |S_y| − |S_x ∩ S_y|)
21       end for
22    end for
23
24    { Cluster reduced similarity matrix. }
25    CLUSTER(M)
26 end procedure
```

Figure 3. Efficient Graph Clustering Algorithm

The steps of the efficient graph clustering algorithm are:

- Calculate normalized citation counts for all vertices (documents) and identify key papers that are cited above some threshold.
- Create a soft cluster around each key paper by analyzing co-citation.
- Calculate similarity measure between soft clusters by ward's hierarchical clustering method.
- Determine cut-off level and merge soft clusters by similarity value.

The overview of the system flow diagram is as shown in Figure 4.

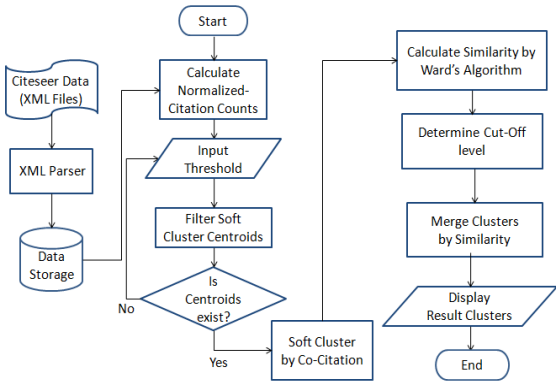

Figure 4. System Flow Diagram

4.1. Normalized Citation Counts

In the general case, graph clustering is a time consuming process because of the temptation to perform the clustering in a way that requires similarities to be calculated for all vertices in a graph. The citations in scientific literature are, however, far from random and very non-uniform. Thus, our approach is to reduce the dimensionally of the problem by first identifying key papers that are cited above some threshold.

Our graph clustering approach is based on the citation counts between papers. One problem with using the raw citation count as a measure of importance is that the older a paper is, the heavier the bias for the paper simply because there has been more opportunity for the paper to have been cited. For example, the older paper A has more citation counts than other papers as shown in Figure 5. To solve this problem we first calculate normalized citation count for each paper. Calculation of normalized citation count included in algorithm is:

$$\text{norm} \leftarrow \text{number of papers published after } v$$
$$\text{norm-cite}(v) \leftarrow |\{(u, v) \in E\}|/\text{norm}$$

In this way, newer papers can be upwardly adjusted while older papers with fewer citations over time are downwardly adjusted. Then, we filter updated key papers (centroid papers) that have greater normalized citation counts than threshold.

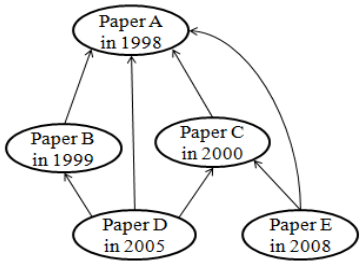

Figure 5. Example of Older Paper with Greater Citations

4.2. Co-citation Coupling

According to the second step of the algorithm, we form soft clusters by co-citation coupling between centroid papers and all other papers. The similarity between documents to form soft clusters is measured by the co-citation counts between these documents as shown in Figure 6.

If papers A and B are both cited by paper C, they may be said to be related to one another, even though they don't directly cite each other. If papers A and B are both cited by many other papers, they have a stronger relationship. The more papers they are cited by, the stronger their relationship is. In Co-citation coupling, the similarity of paper A and paper B is measured by the number of pages cite both A and B. Co-citation information becomes richer over time as more papers are published that cite given documents.

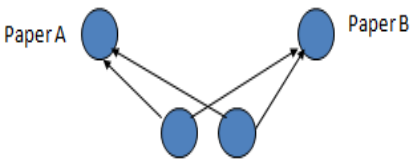

Figure 6. Co-citation Coupling

In algorithm, forming soft cluster by co-citation coupling for each centroid paper is calculated with

$$S_v \leftarrow \{x : \exists_y (y, x) \in E \wedge (y, v) \in E\}.$$

This means that for a node υ (which is a centroid paper) if there exists paper y cite both x and υ, we assume paper x and υ are similar by co-citation coupling. Then we add paper x to soft cluster of paper υ. In this way, we form soft cluster around each centroid paper. So the number of soft clusters is equal to the number of centroid papers.

4.3. Similarity between Soft Clusters

The third step in algorithm is the calculation of similarity between soft clusters to form the more compact final result clusters by Ward's hierarchical clustering method. There are many different ways of defining distance (or similarity) between clusters. Because of our method is based on co-citation coupling, the paper contains in one soft cluster can contain in other soft cluster. Thus the similarity between soft clusters is based on these common papers. For example, the calculation of similarity between two soft clusters, A and B is,

$$\frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

Thus, the similarity is defined by the number of elements in common divided by the number of disjoint elements as shown in Figure 7.

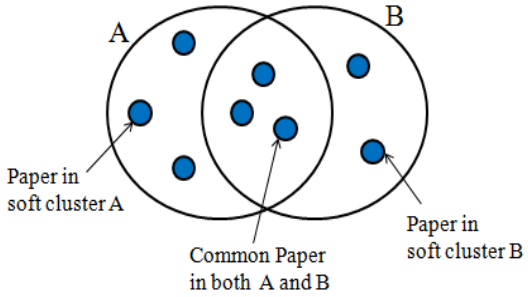

Figure 7. Similarity between Two Soft Clusters

We calculate the similarity measure between all soft clusters n, which results in n × n symmetric matrix as shown in Table 2. If all papers in the two soft clusters are identical, the similarity between these clusters is 1.

Table 2. Example Symmetric Similarity Matrix of Soft Clusters

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	1	0.11	0.25	0.21
Cluster 2	0.11	1	0.34	0.15
Cluster 3	0.25	0.34	1	0.28
Cluster 4	0.21	0.15	0.28	1

4.4. Ward's Hierarchical Clustering Method

Hierarchical clustering builds a cluster hierarchy or, in other words, a tree of clusters, also known as a dendrogram as shown in Figure 8. Hierarchical clustering methods are categorized into agglomerative (bottom-up) and divisive (top-down). A hierarchical agglomerative clustering (HAC) starts with one-point (singleton) clusters and

recursively merges two most appropriate clusters as shown in Figure 9. A divisive clustering starts with one cluster of all data points and recursively splits the most appropriate cluster.

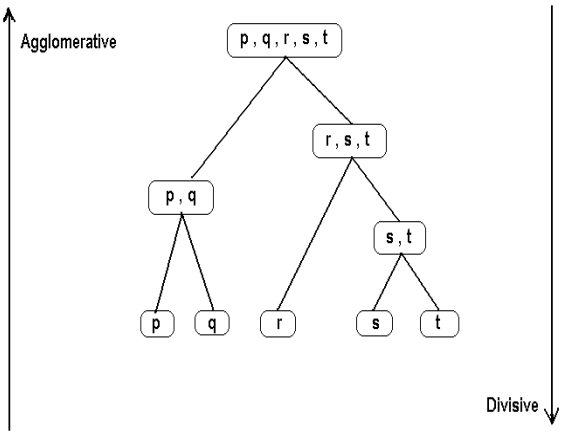

Figure 8. Hierarchical Clustering Dendrogram

In the final step of the algorithm, any standard clustering algorithm can be used. There are several agglomerative hierarchical clustering techniques, such as single linkage clustering (nearest neighbor technique), complete linkage clustering (farthest neighbor technique), average linkage clustering, average group linkage and Ward's hierarchical clustering method.

The HAC algorithm is summarized as follows. Let's assume that we want to cluster n data items, and we have n*(n-1)/2 similarity (or distance) values between every possible pair of n data items:

- Initially, each data item occupies a cluster by itself. So there are n cluster at the beginning.
- Find one pair of clusters whose similarity value is the highest, and make the pair a new cluster.
- Update the similarity values between the new cluster and the remaining clusters.

Ward's hierarchical clustering [5, 6] is an agglomerative hierarchical clustering technique that tends to locate compact and spherical clusters. At each step in Ward's clustering, the union of every possible cluster pair is considered and the two clusters whose fusion results in minimum increase in 'information loss' is combined. Information loss is considered by similarity between soft clusters as mentioned above. Maximum similarity between two soft clusters means minimum information loss. We use Ward's hierarchical clustering with individual soft clusters n and an n × n symmetric matrix of similarities as shown in Table 2. For example, in Table 2, we first merge Cluster 2 and 3 into new cluster named Cluster 2 because their merging similarity is highest. Second, we merge Cluster 1 and 4 into new cluster named Cluster 1. Then Cluster 3 and 4 are excluded and we update the similarity matrix. Then we find the two most appropriate soft clusters in updated similarity matrix according to their similarity. Ward's clustering executes these steps recursively.

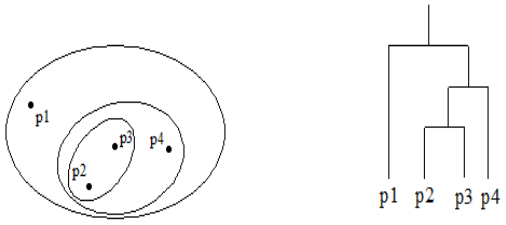

Figure 9. Agglomerative Hierarchical Clustering

4.4.1. Determining Cut-Off-Level

Hierarchical clustering does not require a pre-specified number of clusters. Ward's hierarchical clustering is one of the variance minimization techniques, such as k-means [3]. While k-means requires the desired number of clusters to be specified in advance, Ward's technique allows the posterior choice of the desired level of cluster generality.

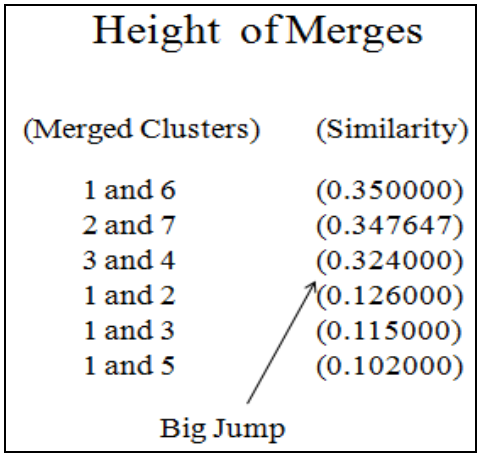

Figure 10. Heights of Merges

We propose a new way to naturally decide the cut level for the hierarchy; the cut-off level is determined by examining the "height of merges" where the gap between two successive combination similarities is largest. We identify the cut-off level where the similarity makes a "Big Jump" to a smaller point as shown in Figure 10.

In Figure 10, cluster 1 and 6 is merged into new cluster 1, cluster 2 and 7 is merged into new cluster 2 and cluster 3 and 4 is merged into new cluster 3 by their similarity. Next, new cluster 1 is merged with new cluster 2. At that point, the gap between their similarity (0.126000) and their above similarity

(0.324000) is too big. So we cut the merging hierarchy at that point. By determining the cut-off level at that point, four final clusters are produced as shown in Figure 11.

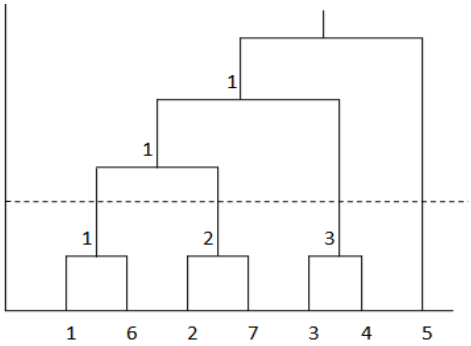

Figure 11. Determining Cut-Off-Level

5. Performance Evaluation

The performance of our system depends on various thresholds as shown in Figure 12. We can see that the smaller the thresholds, the larger the number of final clusters.

Threshold	Soft Cluster	Hieght of Merge	Final Cluster	Final Total Documents
0.001	232	107	125	2080
0.002	96	17	79	724
0.003	58	5	53	415
0.004	44	5	39	296
0.005	35	3	32	195
0.008	20	2	18	142
0.01	15	2	13	115
0.03	5	1	4	35
0.2	2	0	2	23

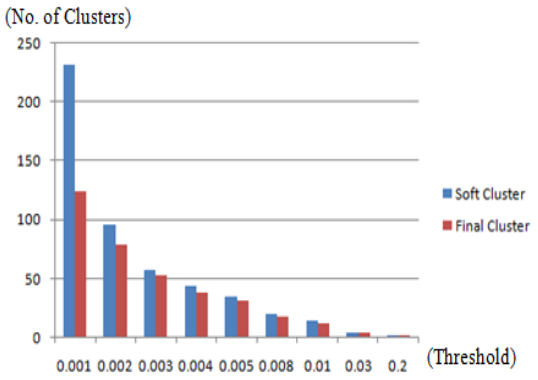

Figure 12. Performance of the Proposed System

6. Conclusion

This system allows for the identification of clusters in a database of scientific literature. Clustering scientific literature is an important problem because it enables tasks such as estimating the amount of activity, growth, and decay in different scientific areas, identifying the fragmentation or merging of disciplines, and assisting a user in navigating through a database. This system is applied to a database of computer science papers from the CiteSeer database. This system can save wasted time, effort and funds in finding the collections of documents that are all related to one another and can avoid unwitting duplication of research for researchers.

7. References

[1] A.Popescul, G. W. Flake, S. Lawrence, L. H. Ungar, and C. L. Giles. Clustering and Identifying Temporal Trends in Document Databases. In IEEE Advances in Digital Libraries ADL-2000, pages 173-182, Washington, DC, 2000.

[2] C.Chen and L Carr. Trailblazing the literature of hypertext: author co-citation analysis (1989-1998). In *Proceedings of the 10th ACM Conference on Hypertext and hypermedia: returning to our diverse roots*, pages 51-60, 1999.

[3] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. LeCam and J. Neyman, editors, *Proceedings Fifth Berkeley Symposium on Math. Stat. and Prob.*, pages 281–297. University of California Press, 1967.

[4] J. Pitkow and P. Pirolli. Life, death, and lawfulness on the electronic frontier. *In Proceedings of Human Factors in Computing Systems*, pages 383-390, 1997.

[5] J. Ward Jr. Hierarchical grouping to optimize an objective function. *J. Amer. Statist. Assoc.*, 58:236–244, 1963.

[6] L. Kaufman and P. J. Rousseeuw. *Finding Groups In Data: An Introduction to Cluster Analysis*. Wiley-Interscience, 1990.

[7] S. Lawrence, C.L. Giles, and K. Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6):67-71, 1999.

[8] S. Lawrence, K. Bollacker, and C. L. Giles. Indexing and retrieval of scientific literature. In *Eighth International Conference on Information and Knowledge Management, CIKM 99*, pages 139–146, Kansas City, Missouri, November 1999.