A Concentric-Circle Model for Community Mining in Graph Structures

Shailesh kumar*

ABSTRACT

Discovering communities from a graph structure such as the Web has become an interesting research problem recently. In this paper, comparing with the state-of-the-art authority detecting and graph partitioning methods, we propose a concentric-circle model to more accurately define communities. With this model, a community could be described as a set of concentric-circles. The most important objects representing the concept of a whole community lie in the center and are called core objects. Affiliated objects, which are related to the core objects, surround the core with different ranks. Base on the concentric-circle model, a novel algorithm is developed to discover communities conforming to this model. We also conducted a case study to automatically discover research interest groups in the computer science domain from the Web. Experiments show that our method is very effective to generate high-quality communities with more clear structure and more tunable granularity.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - *Data mining;* H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval.

General Terms

Algorithms, Performance

Keywords

Community, concentric-circle model, web mining

1. INTRODUCTION

Many communities, either in an explicit or implicit form, have existed in the Web today, and their number is growing at a very fast speed. Discovering communities from a network environment such as the Web has become an interesting research problem recently. Network structures like the Web can be abstracted into directional or non-directional graphs with nodes and links. It is usually rather difficult to understand a network's nature directly from its graph structure, particularly when it is a large scale complex graph. Data mining is a method to discover the hidden patterns and knowledge from a huge network. The mined knowledge could provide a higher logical view and more precise insight of the nature of a network, and will also dramatically decrease the dimensionality when trying to analyze the structure and evolution of the network.

Quite a lot of work has been done in mining the implicit communities of users, web pages or scientific literature from the Web or document citation database using content or link analysis [7, 8, 13, 17]. Several different definitions of

*Faculty, IMS Dehradun.

community were also raised in the literature. In [8], a web community is a number of representative authority web pages linked by important hub pages that share a common topic as shown in Figure 1(a). In [13], a web community is a highly linked bipartite sub-graph and has at least one core containing complete bipartite sub graph as shown in Figure 1(b). In [7], a set of web pages that linked more pages in the community than those outside of the community could be defined as a web community (see Figure 1(c)). Also, a research community could be based on a single most-cited paper and contain all papers that cite it [17] (see Figure 1(d)).

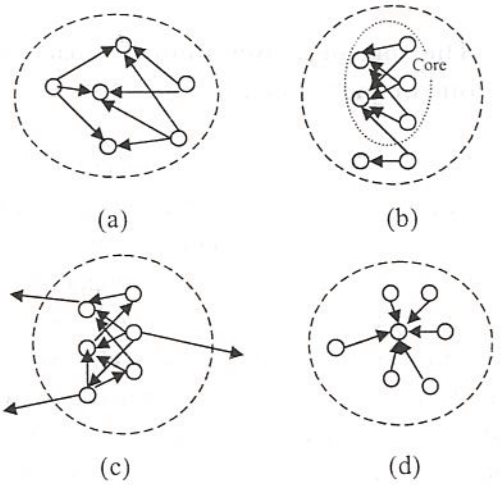


Figure 1: Several different definitions of community

While each of the above definition characterizes some essential properties of a community, it makes the community mining task rather difficult because of a lack of uniform definition. In this paper, we propose a novel concentric-circle model to describe a community. In this model, a community is represented as a set of concentric circles. The central circle is the core of the community and is made up of a set of authoritative objects that are simultaneously linked by other objects. The core, as a whole, could completely represent the concept of the community. Affiliated objects are distributed in outer concentric circles and ranked according to their importance.

In prior works, since a community is simply defined as a group of related objects, community mining is considered as a clustering problem. Many existing methods have been directly applied to detect communities. Among these methods, authoritative resources finding [8, 17, 5, 6] and graph structure partitioning [13, 7] are the two major clustering methods currently widely used in community mining. Using these clustering methods to identify communities has several shortages. First, objects in a cluster are not ranked. Secondly, clusters are not allowed to overlap. That is, one object generally can only belong to one cluster. Thirdly, the similarity between objects is required to be measured by some explicit functions, which are usually hard to define. In this paper, we develop a new method based on our proposed concentric-circle model to automatically discover communities in a complex graph structure. This method overcomes the above shortages and has proved to be effective from our experiments. This method could generate understandable communities with more clear structure and tunable granularity.

The rest of this paper is organized as follows: In Section 2, we describe our proposed concentric-circle model for community in detail. In Section 3, a mining algorithm for the concentric-circle communities is introduced. Granularity tuning options and parameter selection are discussed. Section 4 shows our experimental results. Related work is introduced in Section 5. And finally we conclude our work and discuss some future work in Section 6.

2. A CONCENTRIC-CIRCLE MODEL FOR COMMUNITY

In this section, we describe how to use a concentric-circle model to describe a community. With this model, a community could be described as a set of concentric-circles, as shown in Figure 2. The most important objects representing the concept of a whole community lie in the center and are called *core objects*. *Affiliated objects*, which are related to the core objects, surround the core with different ranks.

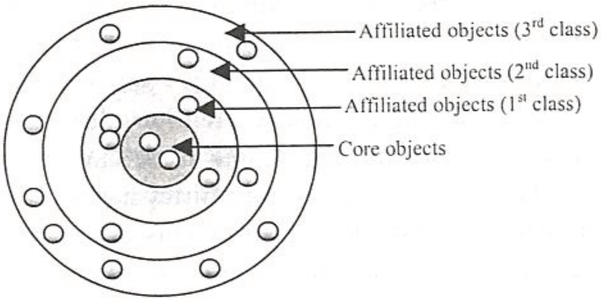

Affiliated objects (3rd class)
Affiliated objects (2nd class)
Affiliated objects (1st class)
Core objects

Figure 2: A concentric-circle model for community

With this model, a community is defined as a four-tuple $<C, A, F, Va>$. C denotes the core object set; A denotes the affiliated object sets; F is the affiliation definition function measuring two objects i and j, and will return a positive value if i is affiliated by j. Va is the importance vector for A to measure the affiliating degree for every object in A to the core C.

For example, F could be such a function that returns 1 if j has a direct link to i and returns 0 otherwise; or a function that returns the reciprocal of the sum of link weights if there is a path from j to i. For each $a \in A$, there exists at least one object $b \in C$, s.t. $F(a, b) > 0$.

Below we will explain the properties of this model in detail using research communities built on a paper citation database as an example.

✧ **Core objects and affiliated objects in a community are distinguished.**

It will be convenient for users to browse a community if the most representative objects are highlighted and objects are ranked according to their importance to the core topic. Take the research community as an example. Papers in a research community are not equally important. Some may give great contributions and pioneer the area, and others may be only follow-up or "delta" works. The classical papers, as a whole, naturally compose the core and define the topic or concept of this community. So in a graph structure, a community could be completely represented by several authoritative objects in the core that have many in-links. Other objects of the community are surrounding the core and affiliated to the core objects. Discriminating core and affiliated objects and ranking objects according to their importance will greatly help user understanding the structure and nature of a community.

✧ **The core of a community is made up of one or more objects.**

Using a single important object to form the community is convenient and ideal [17]. But in most cases, the core of a community is often a combination of several objects. For example, R*-tree is a very basic technique which is used as the foundation stone of several research areas such as high dimensional indexing, spatial database, clustering, etc. If we only use the authoritative paper "*The R*-tree: an efficient and robust access method for points and rectangles (by N. Beckmann et al)*" as the core to construct a community, we will most likely get a community mixed with papers from multiple areas. If we combine it with other papers, e.g. "*Efficient processing of spatial joins using R-trees (by B. Seeger et al)*", we will get a better core to define a coherent research community.

✧ **It is not required that objects in the core of a community should be tightly linked.**

In most of previous methods [7, 8, 13], it is required that objects in the core (or the whole community) are tightly correlated with explicit links. Although it is reasonable to assume that core objects have strong correlations among themselves, we argue that it is unnecessary to assume that objects in a core should be interlinked with many *explicit* links. Let us clarify this with an example. Protein molecule structure prediction is a research area in bioinformatics. Suppose some works may reference two classical papers, one in molecule computation and the other in sequential analyzing. Obviously they are quite independent and have no explicit link between them. For this new research community, if explicit links are required among core objects, then the core could not be formed until some papers referencing both classical papers become important enough to connect them to form the core. This example shows that emphasizing explicit links between core objects may delay the formation of a community. Thus, we argue that the linkages among core objects should be implicit and be built through other objects' co-citations. In other words, we focus more on the 'hidden links' deduced from the links of affiliated objects.

- ✧ **Hub objects are not included in the core.**

Authority and *hub* are used in the HITS algorithm [11] to describe two types of important objects. Loosely speaking, an object pointed to by many other objects is an authority and an object pointing to many other objects is a hub. In the previous works [7, 8, 13], the authority and hub objects are not distinguished in terms of community constructing. Hub objects may be useful in finding important authority objects, such as portal websites. But in most cases, hub objects should not be added into the core since they have no contribution to the concept of a community. In [4] a brief discussion about using hub values for constructing web communities is given. One may argue that a good survey paper (can be viewed as a "hub" paper) which references a number of important papers is also influential in the research community. If this survey is widely accepted and cited, it would also be added into the core because of its high authority value caused by citations, but not because of its high hub value. Another important reason for not including hub objects in the core is the "mixed-hub" phenomena, meaning that a hub object may be related to multiple different topics and communities. Adding "mixed-hub" objects into the core of a community may lead to the risk that authority objects from other communities would also be included into the core and thus result in a topic drift.

- ✧ **Affiliated objects are expanded gradually according to local hub value**

In our model, each affiliated object acts like a small hub to those core objects. The more core objects in a community an affiliated object links to, the better it matches the topic of this community. Thus we use the 'local hub' value as the ranking criteria for these affiliated objects An object referring most of the core objects is closer to the core, and those objects only having indirect and transitive links to the core are farther and more marginal.

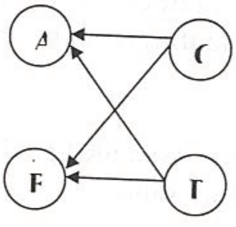
3. COMMUNITY MINING

Based on the above concentric-circle model, the goal of community mining is to discover object sets conforming to this model from a graph.

3.1 Basic Algorithm

Given a graph and its link topology, the basic algorithm of community mining contains two phases: generating core sets and expanding the core sets with affiliated objects. We use 'core set' here to emphasize that the core of a community is a set of objects.

A graph can be represented by an adjacent matrix as shown in Figure 3. Core sets can be found through analyzing the co-citations in the graph, which is equivalent to calculating frequent itemsets in the associate rule algorithm [2]. In other words, we need to find out all combinations of objects which meet certain link threshold (or support) in the graph. These frequent itemsets are candidate core sets of communities.



$$\Rightarrow \begin{array}{cc} & \begin{array}{cccc} A & B & C & \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

$\Rightarrow$ *Frequent Itemset* (A, B)(support $= 2$)

Figure 3: Adjacent matrix of a graph

The size of a core set may vary from ones to thousands. Finding all possible long itemsets is computationally expensive. A more efficient way is to find a number of short itemsets and then assemble them to longer ones. If two itemsets, say AB and ABC, meet the support threshold, we prefer the longer one because it could result in a more accurate community. In our approach, m-itemsets is generated from (m-1)-itemsets. Once an itemset is obtained, all of its subsets will be filtered out. An algorithm about how to use (m-1)-itemsets to generate m-itemsets orderly is described in [2]. Below is the pseudo code of finding frequent itemsets.

```
Generate 1-itemsets IS1 with minimal support S
k ← 2
while $k \le m$ do //generate up to m-itemsets, m is the length of the longest itemset
    Generate k-itemsets ISk using (k-1)-itemsets IS(k-1) with S
    Prun IS(k-1) using ISk
    k ← k +1
end
Put IS1 to ISm to itemsets set IS
```

In this algorithm, the support threshold S is used to denote the minimal support needed to put objects in an itemset.

Once core sets are found, they can be expanded to produce complete communities. The basic idea is to use these core sets as initial communities, and then get affiliated objects according to the core's in-links and add them to the communities. This process is performed iteratively until no more objects could be added to any communities. If we want to distribute affiliated objects into multiple outer circles, local hub values of these affiliated objects are calculated as the ranking and differentiating criteria. Below is the pseudo code of this expanding phase.

```
for every itemset I in IS
    Put objects in I to community C
    do
        Add objects not in C but having links to objects in C to C
        Calculate ranking value of new added objects
    until No more objects could be added
    Put a copy of C to communities set CS
    Clear C
end
```

3.2 Tuning Granularity

Using the basic algorithm alone will generate a large amount of communities. Many of them may have only negligible difference and it is more reasonable to combine them into one community. Suppose that we have two communities expanded from core itemsets ABCDE and ABCDF, it is quite hard to say there are significant differences between them. This is especially the case for the long itemsets with big parts overlapped. Therefore, some tuning mechanisms are needed to merge similar communities to get coarser but more reasonable results. The granularity tuning could be conducted at either the core set generating phrase or the expanding phrase. So there are two kinds of merging process - core set merging and community merging. We will discuss each of them in the following.

3.2.1 Core Set Merging

Core set merging means to combine two similar core sets. Suppose there are two core sets *ABCDE* and *ABCF* as shown in Figure 4. It is easy to see that quite a big part of these two itemsets, i.e. the subset *ABC*, are overlapped. As for the different parts, the itemset *DE* and *F*, if they get strong support to be included in a frequent itemset *DEF*, it is clear that these two core sets should be merged together, that is, to form a new itemset *ABCDEF*.

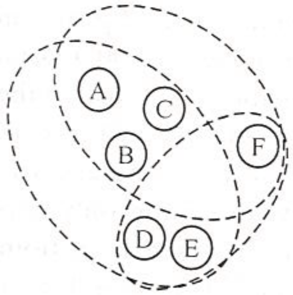


Figure 4: Core set merging

However, core set merging will cause to exceed the limit of the support threshold. In the above example, the support of the new itemset *ABCDEF* will not meet the support threshold anymore (otherwise, it would have been got in the core set generating phase). So we need to redefine some new constraints to control this merging process. Let *Si* be the object set of an itemset *i*, $\|Si\|$ be the number of objects in *Si*, *Support(T)* be the support value of itemset *T*. The constraints for core set merging (which means that only when all of them are met, could itemset *i* and itemset *j* be merged) are defined as follows:

(1) $Min(\|Si\|,\|Sj\|)/\|Si \cap Sj\| < 2$;

(2) $\exists T \subset \|Si \cup Sj - (Si \cap Sj)\|$, $\|Si \cup Sj - (Si \cap Sj)\| / \|T\| < 2$, support (T) $\geq$ S;

(3) $\|T\| \geq 2$, $\ni o_1 \in T$ and $o_1 \in (Si - (Si \cap Sj))$, $\ni o_2 \in T$ and $o_2 \in (Sj - (Si \cap Sj))$

3.2.2 Community Merging

As shown in Figure 5, in some cases, even if two communities have different core sets, quite a lot of their affiliated objects could be coincident (the coincident objects are represented as shadowed circles in Figure 5). Sometimes it would be better if they are merged into one community. Here the similarity between communities is determined by the overlapping condition of affiliated objects. So we have a simpler constraint than the core sets merging. Let *ESi* denote the affiliated object set expanded from the core set *Si*. Since every affiliated object in *ESi* has a ranking weight w_k, as we mentioned previously, we slightly modify the constraint rule (1) of core set merging by substituting object number with weight sum. Two communities originating from itemset *i* and *j* could be merged if and only if the following constraint is satisfied:

$$Min(\sum_{\|ESi\|} w_k, \sum_{\|ESj\|} w_k) / \sum_{\|ESi \cap ESj\|} w_k < 2$$

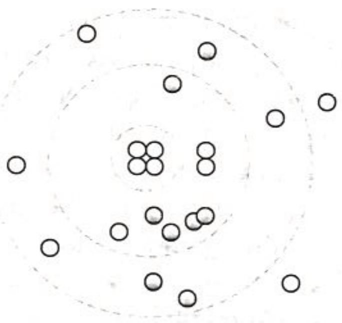


Figure 5: Community merging

These two kinds of merging can be used together or separately. Community merging has a potential risk to result in communities with too coarse granularity. The reason mainly lies in that it only considers the object overlapping and ignores the link support. In general, community merging is suitable only when the results of core set merging don't meet the granularity requirement.

3.3 Parameter Setting

In the core set generating algorithm, objects are scanned iteratively to generate the itemsets of different size orderly until no longer itemset is found. The length of the longest itemset meeting the support threshold, *m*, is used as an important termination condition. However, we do not know the number in advance. Moreover, directly computing all itemsets meeting the support threshold is very expansive. Since we use core set merging method to assemble long itemsets from short ones, it is needless to generate all itemsets. For example, we can use 2 itemsets to merge to an 8-itemset incrementally, or use 5-itemsets to approximate it, or directly calculate it. Experiments in Section 4 show that our algorithm is insensitive to the size of itemsets. Generating initial 2-itemsets and then merging them could result in very close result as that generated by the longest itemsets. It is clearly there is a tradeoff between the core set generating and merging phases. We will discuss this point in detail in Section 4.

Another important parameter is the support threshold *S*. This is a parameter needed to be decided according to experiences or experiments. Here we suggest a way to estimate this value. First, randomly pick a number of nodes (no less than 1% of total nodes) in the graph and calculate their in-links and out-links. Then the amplified average links of each node can be used as an initial estimation of *S*, i.e, $S = f \times \sum_R w_i / \|R\|$, where *R* is the random selected node set and $\sum_R w_i$ is the weight sum of all links related to *R*. *f* is the amplifying frequency factor and currently is set to 2 experientially. This estimation is particularly useful to find those potential emerging communities.

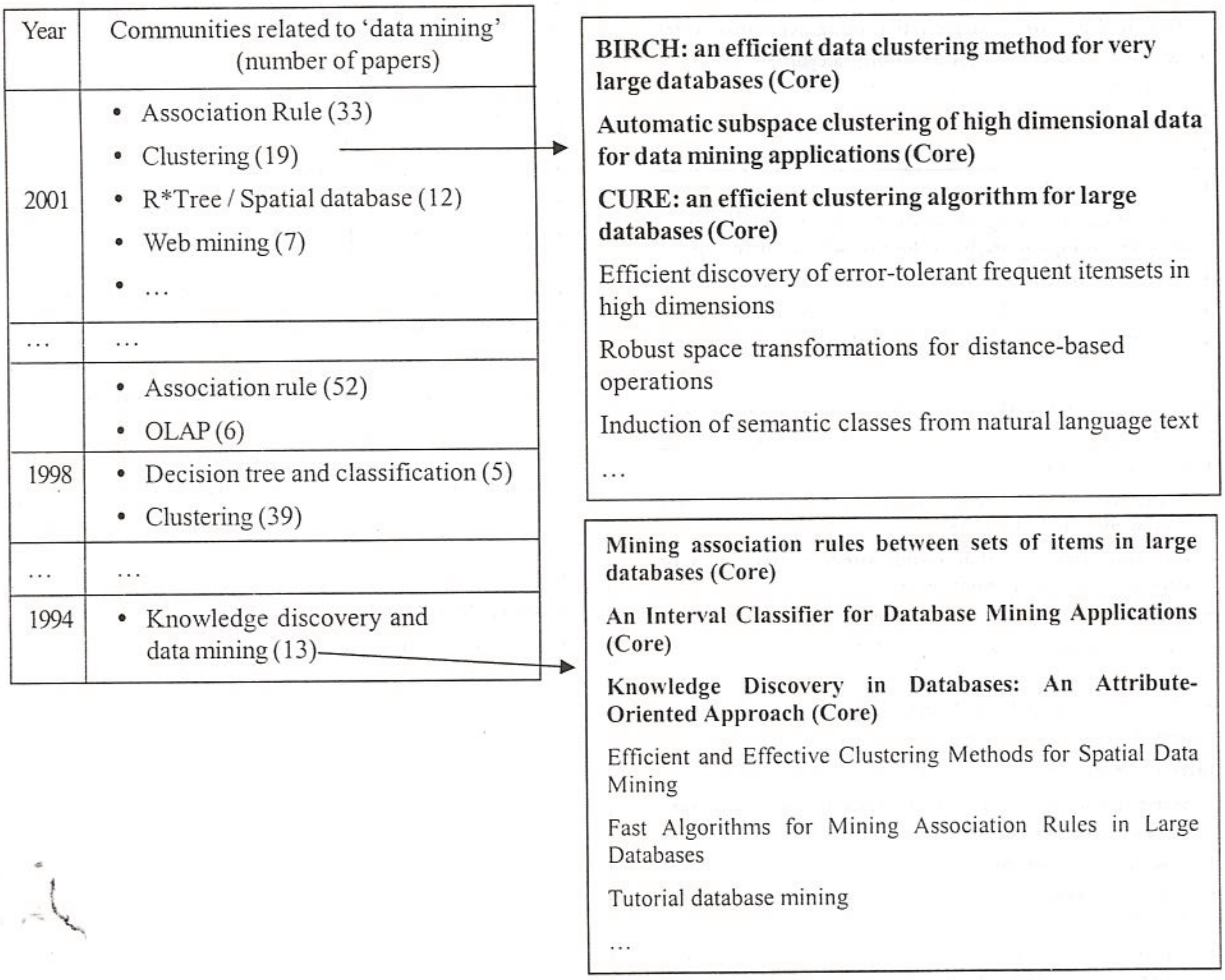
4. EXPERIMENTAL RESULTS

Based on the proposed concentric-circle model and community mining algorithm, we conducted a case study to automatically discover research interest groups in the computer science domain. We collected more than 60,000 papers from 168 conferences and journals by crawling the ACM digital library (http://www.acm.org/dl) and some important conference websites such as VLDB, ICDE, etc. Most of the collected papers are about database and data mining areas. So our investigation will mainly focus on finding the interest groups related to these areas. For each paper, we only collected and used its metadata such as authors, abstract, references and published year. We use the paper citation relationship to build links among papers. As we want to construct a closed mid-sized graph, only references to our collected papers are extracted. This results in a total 61,000 directional links in the citation graph. About half of the objects are isolated and have no in-links or out-links. Because it is meaningless to construct communities containing only one paper, the isolated nodes are filtered out from the graph. Support threshold is set to 4.0 according to our parameter estimation method.

4.1 Quality of Mined Communities

According to the time that papers were published, we built yearly graphs whose nodes were only made up of papers published in that year and their referencing papers published in the year before. For each yearly graph, we discovered communities using our concentric-circle model and mining method. Totally 981 communities were produced and, for each year, the number of communities ranged from 2 to 75. Figure 6 illustrates the communities related to the Data Mining area. The results are rather interesting. From this table, we can see that the first interest group about data mining emerged in 1994. There are totally 13 important seed papers (in our database) at that time and three papers are highlighted as core papers. Then this research area developed very rapidly and was split into multiple branches. For instance, in 1998, four interest groups related to Data Mining, that is, "Association Rule", "OLAP", "Decision Tree" and "Clustering", are discovered. In 2001, there are more than ten branched interest groups related to data mining. Some of them are newly emerging communities such as "web mining". At the same time, some traditional interest groups dispersed or merged yearly. The right side of Figure 6 shows the details of two mined communities. We can see that the core of a community is usually made up of several objects. Also, there are no explicit citation relationships among these core objects.

The mining results are also compared with the search results of the CiteSeer literature search engine (<http://citeseer.nj.nec.com/cs>). To retrieve communities, all of the communities are indexed using words from titles, keywords and abstracts of papers in each community, and each word is weighted simply using its frequency. Table 1 shows the results of using query 'web mining' to retrieve papers in our research interest groups and in CiteSeer website. Compared with



Year	Communities related to 'data mining' (number of papers)
2001	• Association Rule (33) • Clustering (19) • R*Tree / Spatial database (12) • Web mining (7) • …
…	…
1998	• Association rule (52) • OLAP (6)
	• Decision tree and classification (5) • Clustering (39)
…	…
1994	• Knowledge discovery and data mining (13)

BIRCH: an efficient data clustering method for very large databases (Core)

Automatic subspace clustering of high dimensional data for data mining applications (Core)

CURE: an efficient clustering algorithm for large databases (Core)

Efficient discovery of error-tolerant frequent itemsets in high dimensions

Robust space transformations for distance-based operations

Induction of semantic classes from natural language text

…

Mining association rules between sets of items in large databases (Core)

An Interval Classifier for Database Mining Applications (Core)

Knowledge Discovery in Databases: An Attribute-Oriented Approach (Core)

Efficient and Effective Clustering Methods for Spatial Data Mining

Fast Algorithms for Mining Association Rules in Large Databases

Tutorial database mining

…

Figure 6: Yearly communities related to Data Mining

CiteSeer's traditional list-style results, our results are organized in a more reasonable way: papers about different sub-topics are clustered into separate groups. In each group, most important papers are highlighted as core papers and other papers are ranked according to their importance. Such a kind of organization provides a clear global view of the structure of an area and is very helpful for users to more efficiently search for information. Compared with other result clustering methods, all clusters in our method are mined offline and needn’t to be generated on the fly. We also noticed that some important papers in CiteSeer's results are missed in our results. It is understandable since our datbasea is incomplete and contains only a small subset of that of the CiteSeer database.

4.2 Performance Evaluation

We also conducted some experiments to test the influences of using different initial itemsets. In our dataset, the longest itemset is 7-

Table 1: Comparison with CiteSeer

Using Concentric-circle clustering (Top 5×4)	Using CiteSeer (Top 20) (2002-11-13)
Interest Group #1: • HyPursuit: a hierarchical network search engine that exploits content-link hypertext clustering (core) • Enhanced hypertext categorization using hyperlinks (core) • WebACE: a Web agent for document categorization and exploration • The Web as a graph • Inferring Web communities from link topology	• Statistical Pattern Recognition: A Review • Web Mining Research: A Survey • The World Wide Web: Quagmire or Goldmine? • WebSIFT: The Web Site Information Filter System • A Study of Approaches to Hypertext Categorization • Using Data Mining Techniques on Web Access Logs to.. • On Clustering Validation Techniques • Data mining models as services on the internet • Discovery and Evaluation of Aggregate Usage Profiles.. • Discovery of Web Robot Sessions based on their Navigational.. • Measuring the Accuracy of Sessionizers for Web Usage.. • On Mining Web Access Logs • An efficient algorithm for Web usage mining • Putting the World Wide Web into a Data Warehouse: - Dwh-Based Approach To… • Blockmodeling Techniques for Web Mining • An approach to build a cyber-community hierarchy • A Machine Learning Based Approach for Table Detection on The Web • Clustering the Users of Large Web Sites into Communities • Mining Web Access Logs Using Relational Competitive .. • First-Order Learning for Web Mining
Interest Group #2: • Integration of heterogeneous databases without common domains using queries based on textual similarity (core) • Snowball: extracting relations from large plain-text collections • Query containment for data integration systems • Providing database-like access to the Web using queries based on textual similarity • Discovering unexpected information from your competitors' web sites	
Interest Group #3: • Scatter/Gather: a cluster-based approach to browsing large document collections (core) • Constant interaction-time scatter/gather browsing of very large document collections (core) • Web document clustering: a feasibility demonstration • Virtual reviewers for collaborative exploration of movie reviews • Using clustering and visualization for refining the results of a WWW image search engine	
Interest Group #4: • Scatter/gather browsing communicates the topic structure of a very large text (core) • Finding and visualizing inter-site clan graphs • Fast and effective text mining using linear-time document clustering • Exploring browser design trade-offs using a dynamical model of optimal information foraging • Deriving concept hierarchies from text	

itemset, which means that one paper at most could cite 7 other papers in the dataset. It is obvious this number can be increased if more papers are collected. We generate itemsets by setting the size of initial itemsets from 2 to 7 respectively. Then core set merging and community merging are used to construct the final communities.

The running time of each setting is shown in Figure 7. Using 2-itemsets as initial itemsets is most efficient and can save about 25% time compared with that using initial itemsets longer than 4. It will spend only half of the time to find all frequent 2-timesets than to find all frequent itemsets. But merging from 2-itemsets will need more time. In addition, there are no significant difference between running times when the size of the initial itemsets is larger than 4.

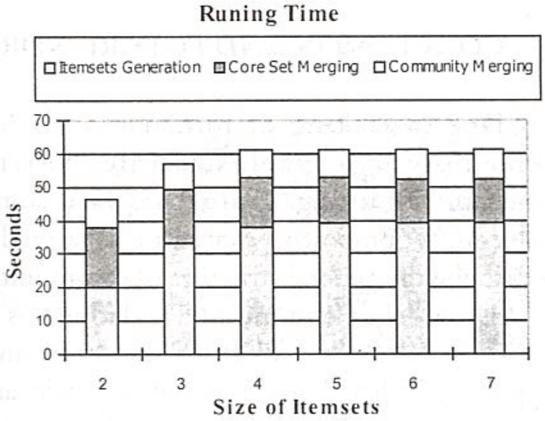


Figure 7: Performance comparison of using different initial itemsets

Although using short initial itemsets can benefit the running time, it will be unpractical if it heavily affects the quality of the generated communities. Therefore, we also measured the effects on the quality of communities when using different initial itemsets. Figure 8 and Figure 9 shows that no matter how variant the initial itemsets are, the final communities generated are very stable in all settings. In Figure 8, the number of communities has tiny perturbation (less than 1%) before using up to 4-itemsets and becomes stable thereafter. We also compare the structure difference of the communities obtained by using different initial itemsets. The baseline is the communities obtained by directly using the longest initial itemsets, that is, the 7-itemsets. Cosine similarity [18] is used to measure the similarity between communities. Figure 9 shows the average cosine degree and standard deviation between the baseline and the communities obtained with varying initial itemsets. Generally, the Cosine values are rather small, which denotes that the difference of the final communities is very little when using different initial itemsets. Even when we use 2-itemsets as the initial itemsets and only use core set merging to generate final results, the average Cosine value is still less than 16 degree.

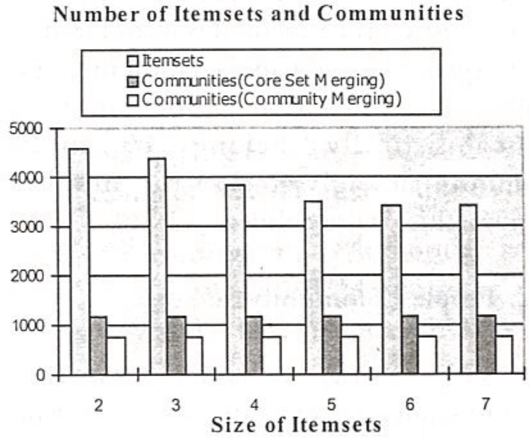


Figure 8: Number of communities generated using different initial itemsets

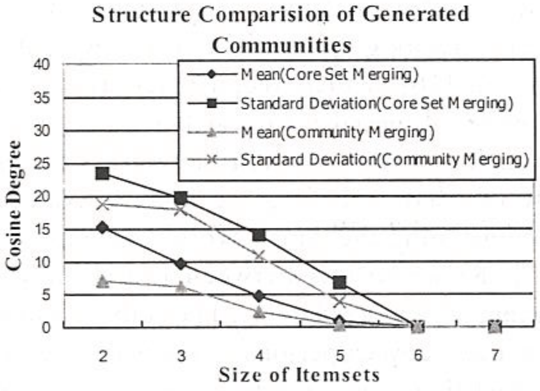


Figure 9: Structure comparison of communities generated using different initial itemsets

5. RELATED WORK

5.1 Web Community

Discovering web communities attracts a lot of attentions in recent years. [3] uses focused crawling to discover web resources related to a certain topic. "Hypertext-Induced Topic Selection" (HITS) algorithm [11] is applied to find web communities in [8] and [12]. [13] defines the core of community as a complete bipartite sub-graph and proposes an iterative pruning algorithm to fit the size of the whole web. In [7], web pages in a community are required to link to more pages inside than those outside. By combining HITS and graph partitioning, they use maximum flow and minimal cuts to separate sub-graphs. A Probabilistic-HITS method is introduced in [6] and experiments on paper citation in Cora and web pages show that a document could probabilistically belong to multiple communities, given that the number of communities is pre-defined.

5.2 People Community

Some works in social network and recommender systems are related to finding people communities. [19] aims to discover people sharing common interests based on email communications. An interest distance is defined under a graph structure. The Referral Web, a social network graph of field experts is built in [9] and [10], which reconstructs the social networks of specialized researcher communities through co-author relationship and focuses on referral chains. It is quite a good example demonstrating the 'Small World Phenomenon' [15]. Relationships between individuals in campus are extracted in [1]. The authors assume that links between persons' home pages indicate their relationships in the real world, and find out students' social communities by analyzing link topology.

5.3 Bibliometrics and Document Citation

As the citations among literatures could be regarded as hyperlinks in document space, similar community mining work also can be found in bibliometrics and document citation research. [5] collects papers published in ACM Hypertext Conference series from 1987 to 1998, picks out 367 important authors, uses PCA to extract factors which might imply study fields, and visualizes a periodic author co-citation map. Based on the CiteSeer scientific literature database, [17] raises a graph clustering algorithm to cluster papers and shows yearly growth of those clusters. They use each of the selected key papers which meet certain citation threshold as centroid and then expand them to cluters. Inter-cluster similarities are calculated and an agglomerative hierarchical clustering is used.

6. CONCLUSION AND FUTURE WORK

Due to lacking of formal models for community in a graph structure, current community mining approaches face some problems. We proposed a concentric-circle model to describe the general structure of community. In this model, a community is defined as a combination of core objects in the center and supporting affiliated objects in outside circles are around the core. We also introduced a mining algorithm to automatically generate communities from a graph. Experiments on a documents citation database showed that our algorithm is solid and effective. It could find communities of adaptive granularity. Tuning options and performance issues are also discussed.

Currently we only apply the concentric-circle model and mining algorithm to discover interest groups from paper citation database. Since this is a general method that could potentially be used in any application scenarios where the data can be abstracted to a graph structure, we prepare to test its usefulness in other environments such as human relationship network, newsgroups, communication network, etc. We also plan to analyze the evolution of communities. Since the objects and links in a network are usúally dynamic, we could observe the communities and their changes in a time series manner.

7. REFERENCES

[1] Lada A. Adamic and Eytan Adar, Friends and Neighbors on the Web, Technical report, Xerox Parc, 2002.

[2] Rakesh Agrawal, Tomasz Imielinski, Arun Swami, Mining Association Rules between Sets of Items in Large Databases, in Proceedings of the International Conference on Management of Data (ACM SIGMOD), 1993.

[3] Soumen Chakrabarti, Martin van den Berg, Byron Dom, Focused crawling: a new approach to topic-specific Web resource discovery, in Proceedings of The 8th International World Wide Web Conference, 1999.

[4] Soumen Chakrabarti, Byron. E. Dom, S. Ravi Kumar, Prabhakar. Raghavan, Sridhar. Rajagopalan, Andrew. Tomkins, David. Gibson, and Jon. Kleinberg, Mining the Web's link structure, in IEEE Computer (Vol. 32 No. 8: 60-67), 1999.

[5] Chaomei Chen, Les 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, 1999.

[6] David Cohn, Huan Chang, Learning to Probabilistically Identify Authoritative Documents, in Proceedings of the 17th International Conference on Machine Learning, 2000.

[7] Gary William Flake, Steve Lawrence, C. Lee Giles, Efficient Identification of Web Communities, in Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD-2000), 2000.

[8] David Gibson, Jon Kleinberg, Prabhakar Raghavan, Inferring Web Communities from Link Topology, in Proceedings of the 9th ACM Conference on Hypertext and Hypermedia, 1998.

[9] Henry Kautz, Bart Selman, Mehul Shah, Referral Web: Combining Social Networks and Collaborative Filtering, in Communications of the ACM, 1997.

[10] Henry Kautz, Bart Selman, Mehul Shah, The Hidden Web, in AI Magazine, 1997.

[11] Jon M. Kleinberg, Authoritative Sources in a Hyperlinked Environment, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, 1998.

[12] Jon M. Kleinberg, Hubs, Authorities, and Communities, in ACM Computing Surveys, Vol. 31, Number 4es, 1999.

[13] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, Trawling the web for emerging cyber-communities, in Proceedings of The 8th International World Wide Web Conference, 1999.

[14] Bing Liu, Chee Wee Chin, Searching People on the Web According to Their Interest, poster of The 11th International World Wide Web Conference, 2002.

[15] Stanley Milgram, The Small World Problem, Psychology Today, 1, 61, 1967.

[16] Lawrence Page, Sergey Brin, Rajeev, Motwani, Terry Winograd, The PageRank Citation Ranking: Bringing Order to the Web, Unpublished, 1998.

[17] Alexandrin Popescul, Gary William Flake, Steve Lawrence, Lyle H. Ungar, C. Lee Giles, Clustering and Identifying Temporal Trends in Document Databases, in IEEE Advances in Digital Libraries, ADL 2000.

[18] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill Book Company, 1983.

[19] Michael F. Schwarz, David C. M. Wood, Discovering Shared Interests Among People Using Graph Analysis of Global Electronic Mail Traffic, in Communications of the ACM, 1992.