Characterizing and Predicting Community Members from Evolutionary and Heterogeneous Networks*

Qiankun Zhao
AOL Lab, Beijing
China
qiankun.zhao@corp.aol.com

Sourav S. Bhowmick
Nanyang Technological University
Singapore
assourav@ntu.edu.sg

Xin Zheng
Tsinghua University
China
zhengxin99@tsinghua.edu.cn

Kai Yi
Peking University
China
yikai@ebusiness.pku.edu.cn

ABSTRACT

Mining different types of communities from web data have attracted a lot of research efforts in recent years. However, none of the existing community mining techniques has taken into account both the *dynamic* as well as *heterogeneous nature* of web data. In this paper, we propose to characterize and predict community members from the evolution of heterogeneous web data. We first propose a general framework for analyzing the evolution of heterogeneous networks. Then, the academic network, which is extracted from 1 million computer science papers, is used as an example to illustrate the framework. Finally, two example applications of the academic network are presented. Experimental results with a real and very large heterogeneous academic network show that our proposed framework can produce good results in terms of community member recommendation. Also, novel knowledge and insights can be gained by analyzing the community evolution pattern.

Categories and Subject Descriptors: I.6.5 [Simulation and Modeling]: Model Development.

General Terms: Algorithm, Design, Experimentation.

Keywords: evolutionary web community, heterogeneous network, member characterization, member prediction.

1. INTRODUCTION

With the availability of massive amount of data on the web, recently, many web-based communities such as web-based social communities, web page communities, and web user communities, have emerged. As a result, there have been increasing research efforts on extracting communities from the web [1, 6, 8, 10, 12, 13, 17, 21]. The basic idea is to model the web data as a graph/network, where the vertexes represent objects such as web pages or web sites and the edges represent the relationship between web pages or web sites. Then, the problem of *community mining* is to extract subgraphs satisfying certain properties such as objects within the same community are more similar/close to each other than objects outside the community. For example, Flake *et al.* [6] defined a community on the web as a set of sites that has more links to members of the community than to non-members. Then, they proposed an efficient *maximum flow (minimum cut)* approach to identify subgraphs that satisfy the definition. In the literature, there are different definitions of web communities and web community extraction has been proved useful in many applications such as focused crawler, search engines, web page categorization, and improved filtering mechanism [6, 8, 13, 23].

1.1 Motivation

Most of the community mining efforts focus on defining web communities and proposing corresponding community identification algorithms. Our investigation revealed that these efforts suffer from some combination of the following limitations.

Heterogeneous objects and relationships: In existing web community mining approaches, web data are modeled as graphs/networks with the assumption that all the objects are of a single type and the relationship between objects are *homogeneous*. Consequently, web data is represented as a homogeneous network such as the hyperlink-based web page graph in the HITS algorithm [11]. However, in reality, web data and corresponding relationships are *heterogeneous* in nature. Different types of web objects can be found in a network. For example, in a web-based academic network we can find `paper`, `researcher`, `conference`, and `journal` objects . At the same time, there are different types of relationships between these objects such as a paper `"is in proceeding of"` a conference, a researcher `"is the author"` of a paper, and two researchers are `"co-author"` for some papers. As a result, the homogeneous graph/network representation cannot accurately distinguish the heterogeneous web objects and their corresponding relationships.

Dynamic nature of the data sources: As web data is dynamic, the corresponding representation may evolve over

*This work was done when Qiankun Zhao was a Ph.D. candidate in Nanyang Technological University, Singapore.

CIKM'08, October 26–30, 2008, Napa Valley, California, USA.

time as well. For example, with the creation of new hyperlinked web pages and web sites, the web graph structure may change over time. As a result, the web communities may evolve as well. For instance, a specific community may split into several communities or a collection of communities may be merged into one. Furthermore, members in the communities may change from one community to another over time. For example, in the research community, when data mining research emerged in early 1990s, it was considered as a part of the database community. But due to its increasing popularity, data mining has evolved into an individual community. However, in majority of existing web community mining approaches, often web data is considered as snapshot data. Consequently, these efforts have not considered in the web community extraction process the evolutionary nature of web communities as well as the individual members of the communities. Only very recently, there are growing efforts to study evolution of web communities in the context of social network [2, 3, 5, 12, 14, 16, 17].

Beyond clustering of individual members: Most of the existing work considers the community mining as a clustering problem. The objective is to construct a model that can categorize an object into a specific community. However as we shall see later, in several real life applications, community mining is not simply a clustering problem. Specifically, there may exist *community-wide constraints* while the existing clustering problem only considers individual members in the community. For example, in the academic network, there are communities such as conference program committees that need to satisfy certain community-wide constraints. That is, not only should each member in the program committee satisfy certain properties, the community itself as a whole should satisfy some *global constraints*. For instance, members of the conference program committee community as a whole should cover all the related topics in a specific conference and the geographical locations where related research is active.

1.2 Overview

In this paper, ***we propose a novel framework of web community mining by combining the evolutionary and heterogeneous properties of network data***. A key goal of this framework is to characterize and predict members of a community. In our approach, we first model web data as a *heterogeneous network* where the vertexes are different types of objects and the edges represent different types of relationships. Such representation allows us to clearly differentiate the types of objects and corresponding relationships. Note that the reason we differentiate them is because, as we shall see later, different types of objects and relationships often play different roles in different community mining applications. Next, based on a user-defined time granularity, objects and relationships within the same time interval in the heterogeneous network are merged together. Then, a novel structure called *vector-based heterogeneous network* is proposed to represent the relationships for the sequence of time intervals. Note that the edges in the vector-based heterogeneous network not only represent the relationships between objects but also the evolution pattern of the relationships.

After we have represented the heterogeneous and evolutionary properties of the web data using the above structure, we extract features of the community from the vector-based heterogeneous network. We adapt the *PopRank* algorithm [19] to rank the objects and use the rank values as part of the features. Finally, a set of community models is constructed based on the set of extracted features. We propose a two level community model that consists of a *regression* phase and a *multi-class classification* phase. As there often exist hierarchical relationships between communities in real life, we construct the first level of the hierarchy between communities with the *regression* model and use the *multi-class classification* model to further distinguish the communities that cannot be separated by regression. In summary, the main contributions of this paper are as follows.

- We propose a novel framework of web community mining that combines the evolutionary and heterogeneous properties of web data. We illustrate the features and practicality of the framework with a real world example based on the academic network. While there has been several recent effort in studying evolution of social networks [3, 5, 14], to the best of our knowledge, these approaches do not take into account the heterogeneity of web data in the community extraction process.
- We propose a novel structure called *vector-based heterogeneous network* to model the heterogeneity and evolutionary features of web objects and associated relationships.
- We propose an approach based on the *PopRank* algorithm [19] to extract features related to a particular community. Based on the extracted features, we present a two-level community model construction technique based on regression and multi-class classification.
- The academic network data is used to illustrate how the proposed framework work with two representative applications: conference program committee recommendation and researcher evolution tracking.
- We present extensive experimental results with the real academic network data and illustrate that our proposed approach can produce high quality community models and provide insights about the evolution of the communities as well.

The rest of this paper is organized as follows. Related research is presented in Section 2. Section 3 describes the framework of our community mining technique. In Section 4, a real example of the academic network data is used to illustrate the framework. The experimental results based on academic community network data are presented in Section 5. The last section concludes this paper.

2. RELATED WORK

Modeling of massive graphs: There has been several work on developing models for massive graphs such as *configuration model* [18], *generative model* [4], Kleinberg's model for the small-world phenomenon [11], *forest-fire graph model* [15], and *biased preferential attachment model*[14] for social networks. In contrast, we focus on modeling heterogeneous and evolutionary community network by using novel vector-based heterogenous network.

Community extraction from static graphs: In [8], Kleinberg *et al.* defined a web community as a set of representative authority web pages linked by important hub pages that share a common topic. The HITS has been applied to find such web communities in [8, 11]. In [13], Kumar *et al.* defined a web community as a dense directed bipartite

subgraph which contains a complete bipartite subgraph of certain size. They expanded on hubs and authorities by using co-citation as a way to extract all communities on the web and used graph theory algorithms to identify all instances of graph structures that indicate community. In [6], Flake *et al.* defined a community as a vertex subset in which each member vertex has at least as many edges connecting to member vertices as it does to non-member vertices. They proposed to identify such communities using maximum flow and minimum cut algorithms.

Community mining works have also been done in the bibliometrics and document citation research [1, 20, 23]. In [20], various types of citation mining and bibliometrics techniques were discussed in the context of measuring the impact of papers, authors, and journals. In [1], graph clustering algorithm has been applied to cluster papers based on the citation relationships. In [23], a frequent itemset-based algorithm is proposed to generate the core sets of the communities and merging them with affiliated objects.

The above studies typically extract communities from a static (aggregated) graph and miss the details on the dynamic behavior about the communities. In contrast, we analyze the dynamic features for community extraction.

Community extraction from dynamic networks: Recently, there is a large body of work on community extraction from online dynamic networks [12, 17, 21]. In [12], Kumar *et al.* applied Kleinberg's bursty algorithm to identify communities as bursts of hyperlinks between blogs where the bursts are obtained from the *time graph* extracted from the blog graph as a result of crawling the blogs. Lin *et al.* [17] proposed a mutual awareness-based model for blog community formation. Note that these approaches consider only *dynamic* nature of web data whereas our approach is the first to integrate both the *dynamic* and *heterogeneous* properties for web community mining.

Community evolution and dynamics: More recently, evolution of large online communities have been studied in [2, 3, 5, 12, 14, 16, 17]. Leskovec *et al.* [15, 16] have studied the properties of the time evolution of graphs. The results give insights into the evolution of graph properties (such as average vertex degree, distance between pairs of nodes, conductance, *network community profile plot*) over time and statements about trends can be made. Kumar *et al.* [12] studied the evolution of the blogosphere as a graph in terms of the change of characteristics, (such as in-degree, out-degree, strongly connected components), the change of communities, as well as the burstiness in blog community. They [14] also classified the social network graph into three groups: the *singletons*, the *giant component*, and the *middle region*, and studied the evolutionary characteristics of these groups. Backstrom *et al.*[3] provided insights on the structural features that influence individuals to join communities, which communities will grow rapidly, and evolutionary characteristics of overlapping community pairs. Toyoda *et al.* [22] studied the evolution of web communities from a series of web archives by defining different types of community changes, such as emerge, dissolve, grow, and shrink, as well as a set of metrics to quantify such changes for community evolution analysis. Similar works have been done for citation network [9]. Asur *et al.* [2] introduced a family of events on both communities and individuals to characterize evolution of communities. They introduced metrics to measure stability, sociability, influence and popularity for communities and individuals. Falkowski *et al.* [5] proposed to observe the temporal changes to social networks at the subgroup level instead of vertex and edge level. Lin *et al.* [17] developed an *interaction space*-based representation to quantify community dynamics. They established community evolution by maximizing the *interaction correlation* between communities across two time slices.

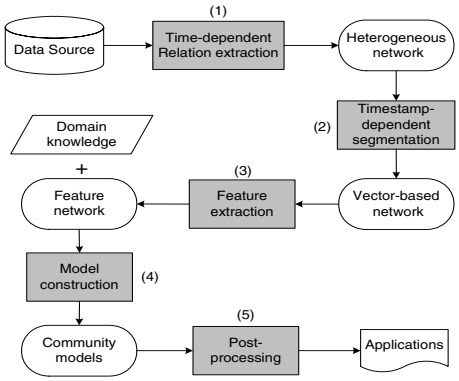

Figure 1: The framework of community mining.

In contrast, our work is complementary in nature. Rather than studying the structural properties of a specific type of network (social network or citation network), our approach aims to study general web community network and integrates features extracted from the dynamics of network data to enhance the community mining process. In our approach not only the evolution of community itself is considered, but also evolutions of members within each community are incorporated to make the community mining results more accurate. Furthermore, our investigation also includes predicting potential community members (*e.g.*, program committee) as well tracking evolutionary features of members.

3. THE FRAMEWORK

In this section, we present the framework of community mining based on the evolution of heterogeneous network. As shown in Figure 1, the framework consists of five major components: the *time-dependent relation extraction* module, the *timestamp-dependent segmentation* module, the *feature extraction* module, the *model construction* module, and the *post-processing* module. Here, we present the overview of the framework. We shall elaborate on each module in the subsequent sections in the context of academic network. The input to this framework is a set of data sources, domain knowledge, and the targeted applications. The objective of this system is to extract community models that can be used in specific applications.

3.1 Time-Dependent Relation Extraction

Given the data sources, the *time-dependent relation extraction* module extracts various types of objects and relationships between them. Different from existing relation extraction approaches such as hyperlink extraction, we extract the types of objects and their relationships along with the corresponding timestamps. For example, for the conference program committee application in the academic network, different objects such as `author`, `conference`, `paper` are extracted together with different relationships such as a paper `"is published in"` a conference, someone `"is the author of"` a paper, someone `"is a co-author of"` someone else, and someone `"is in the program committee of"`

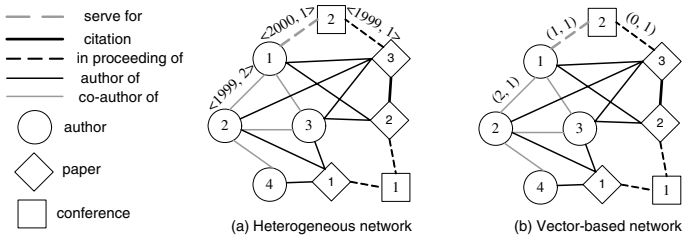

Figure 2: Heterogeneous and vector-based network.

a conference. At the same time, the corresponding time period during which the relationships are valid are recorded as well. For instance, a paper is published in 1999 in a conference. Using the extracted information, a *heterogeneous network* can be constructed. Formally, a heterogeneous network is defined as follows.

DEFINITION 1. [**Heterogeneous Network**] *A heterogeneous network $\mathcal{H}$ is a 8-tuple $\mathcal{H} = (\sum_V, \sum_A, V, A, s, t, \ell_V, \ell_A)$, where 1) V is a set of nodes and A is a multiset of edges; 2) $\sum_V$ and $\sum_A$ are finite alphabets of the available node and edge labels; 3) s: $A \longrightarrow V$ and t: $A \longrightarrow V$ are two maps indicating the source and target nodes of an edge; 4) ℓ_V: $V \longrightarrow \sum_V$ and ℓ_A: $A \longrightarrow \sum_A$ are two maps describing the labels of the nodes and edges.* □

Note that here each node in the network represents an object and each edge represents the relationship between the two connected objects. In this case, there may be multiple edges between two objects and the labels of the edges are the timestamps, types of relationship, and the *weight* of the relationship. Here *weight* of the relationship is measured based on the co-occurrence of the two connected objects. For example, a heterogeneous network is shown in Figure 2(a), where the labels of the edges are shown in the top left; the timestamp and *weight* of the edge are affiliated to each edge. Here *weight* of the edge is the number of times the relationship occurs. For instance, the *weight* of the edge between two authors represents the number of times two authors `co-author` papers. Also, each node has its own label that is listed in the bottom left. Note that there may be more than one edge between any two objects.

3.2 Timestamp-Dependent Segmentation

As mentioned above, each edge in the heterogeneous network has a timestamp. In order to monitor the evolution pattern of the objects and their relationships, we need to differentiate the relationships in the temporal dimension. However, in many real life applications, knowing the exact time of the relationships between objects may not be necessary. For example, in the conference program committee application, it is not necessary to know the exact time that someone is in the conference program committee of a conference. Rather, it is sufficient to know the year of the conference. Hence, for different applications, users can define any time granularity that is important to the application (such as day, month, year, etc). Based on the time granularity, objects and relationships within the same time interval are merged together. Then, the network data is represented as a new type of heterogeneous network called *vector-based heterogeneous network*, where each edge is a vector $w_i = [e_1, e_2, \cdots, e_k]$ such that e_i represents the *weight* of the relationship between the connected objects during time interval t_i. Formally, it is defined as follows.

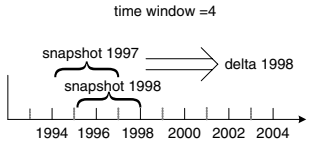

Figure 3: Snapshot and delta-based features.

DEFINITION 2. [**Vector-based Heterogeneous Network**] *A vector-based heterogeneous network $\mathcal{N}$ is a heterogeneous network denoted as $\mathcal{N} = (\sum_V, \sum_A, V, A, s, t, \ell_V, \ell_A)$, where the label of each edge is $\ell_{a_i} = (r, w_i)$, $\ell_{a_i} \in \ell_A$, r is the relationship between two vertexes and w_i is a vector that represents the weights of the relationships in a sequence of time intervals.* □

For example, given the yearly-based time granularity in the conference program committee application, vertexes represent objects such as `paper`, `conference`, and `author`, while the edges represent the weights of the relationships on a yearly basis. For instance, in Figure 2(b), the edge between two authors $w_i = (2, 1)$ represents that the two authors have co-authored two papers in the first year and one in the second year. Here we use the vector-based network representation for two reasons. Firstly, the storage space for the vector-based network is substantially smaller than a sequence of network graphs, as the network graphs can be very huge in many applications. Secondly, the vector-based representation is more flexible compared to a sequence of network graph. Particularly as we shall see later, in the feature extraction phase, different time windows can be used (Figure 3).

3.3 Feature Extraction

The feature extraction module extracts features from the vector-based heterogeneous network. This is the major step where the evolution and heterogeneous properties of the network are taking into account. As for the heterogeneous property, we adapt the *PopRank* algorithm [19] to rank the objects and use the rank values as part of the features. The rank values are obtained based on similarity propagations between different types of objects. At the same time, there are features that can be directly extracted from the graph using graph properties such as degree and distance. Basically, to represent the evolution of the heterogeneous network, we extract two groups of features: the *snapshot-based features* and the *delta-based features*. The *snapshot-based features* refer to features that are extracted from the vector-based heterogeneous network by taking the elements in the same time window in all the vectors. On the other hand, the *delta-based features* represent how the snapshot-based features change over time. For example, given the academic network data from `1994` to `2004`, for each year there is a snapshot-based feature for each object in the network; for every two consecutive years, there is a delta-based feature for each object. Note that the snapshot-based features can be defined using a time-window. For example, we can take the data from `1994` to `1997` together to get the snapshot-based feature for year `1997` with a time window of size *4* as shown in Figure 3. Here the delta-based features for each object are actually the percentages of change to the corresponding feature values in two consecutive snapshots.

3.4 Model Construction

A set of community models can now be constructed based on the set of features extracted using the above mentioned

extraction techniques. In this paper, we propose a two level community model that consists of a *regression* phase and a *multi-class classification* phase. The underlying intuition is that in many real life applications, there are hierarchical relationships between communities. For example, for the conference program committee community, we have top conferences, second tier conferences, and other conferences. Moreover, for conferences at the same level, there are communities with different characteristics. Some conferences focus more on theory while others focus more on application and engineering, even in the same rank. The basic idea is that we can get the first level of the hierarchy between communities with the regression model and use the multi-class classification model to further distinguish the communities that cannot be separated by regression.

3.5 Post-Processing

There may exist constraints that are application dependent. The last component, *post-processing*, is proposed to handle such constraints. For example, for the conference program committee community application, there are not only *local constraints* such as the properties of individual candidate program committee members, but also *community-wide constraints* to the community as a whole. Here *local constraints* refer to individual features such as the research expertise of the candidate; while *community-wide constraints* refer to features of the entire community such as the number of PC members, the area of coverage of all members, and the location of the conference. The local constraints can be modeled in the community model while the community-wide constraints need to be handled via post-processing.

In summary, we propose a framework to model the communities taking into account the dynamic and heterogeneous nature of the network. Given the data sources, relations and timestamps are extracted and modeled as a heterogeneous network. Based on a user-defined time granularity, the heterogeneous network is transformed into a vector-based network representation. Then, community models are constructed based on the snapshot-based and delta-based features that are extracted using the object rank algorithm over the vector-based network.

4. THE ACADEMIC NETWORK

In this section, the academic network is used as an example to illustrate the above framework in detail. Note that the techniques and models discussed in this section can be extended to other types of network as well. Firstly, we explain the reasons for choosing the academic network as an example. Then, characteristics of the academic network data are described. Next, details of feature extraction and community model construction will be presented. Lastly, two applications of the community models are presented to illustrate the usefulness of the model as well as the importance of post-processing.

4.1 Why Academic Network?

The reasons for choosing the academic network as an example to illustrate the above mentioned framework can be summarized as follows.

Firstly, academic network is one typical example of the evolutionary heterogeneous network. The academic network contains various types of objects such as `papers`, `journals`, `conferences`, and `authors`. Moreover, there are many types

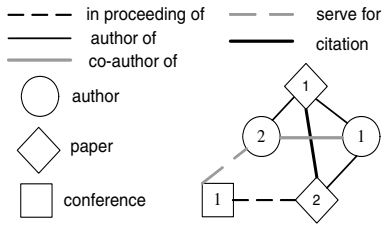

Figure 4: Academic network objects and relations.

of relationships between objects. For example, the relationships involving authors can be of multiple types such as `"co-author of"`, `"colleague of"`, `"co-serving of a conference"`, or `"member-chair"` relationships. Also, the relationships between authors and conferences are definitely different from the relationships between authors and authors. At the same time, the academic network evolves over time. For instance, every year there will be new papers published that contain new authors and new citations to existing papers.

Secondly, there are massive amount of high quality academic network data available on the web such as the academic publication portals: ACM *digital library*, IEEE *explorer*, DBLP, *CiteSeer*, etc. The timestamps of all papers, conferences, journals, and their relationships are available as well. Moreover, there are sets of community data available in the academic network. For example, there are conference program committees, journal editorial boards, special interest groups such as SIGMOD, SIGGRAPH, SIGIR, etc. The historical information related to these communities is also available from the web. Such large source of data enriched with temporal and heterogeneous features provide ideal platform to build our framework.

Thirdly, there exist rich domain-specific constraints in the academic network besides the local constraints. For instance, considering the conference program committee community, beside the constraints on individual candidate committee members, there are *community-wide constraints* such as *diversity* and *coverage*. Here *diversity* refers to the fact that all the members of the program committee should have limited overlaps in terms of physical locations (affiliations) and expertise. *Coverage* refers to the constraint that all the members of the program committee as a whole should cover all the topics listed for the conferences as well as all the targeted geographic locations.

4.2 The Academic Network Data

In this section, we describe the characteristics of the academic network data that will be used in the rest of this paper. The data is extracted from the Libra[1] dataset, which contains more than *1* million research papers in the computer science area `1989` to `2004`, together with *650,000* authors, *1700* conferences, and *480* journals. The types of relationships between these objects are shown in Figure 4. In this paper, mainly five types of relationships are considered. They are `"co-author-of"` relation between authors, `"author-of"` relation between authors and papers, `"in proceeding of"` relation between papers and conferences/journals, `"citation"` relation between papers, and `"serve for"` relation between researchers/authors and conferences/journals. In total, there are more than *7* million object relationships in this collection.

[1] *Libra* is an object-level web search prototype for research materials in Microsoft Research Asia[19]

Conference	CFH	NDA	NOP	NDP	YPC
SIGMOD	1970	2268	3289	186	1998
VLDB	1975	1981	2145	509	1996
ICDE	1984	1672	2656	293	1999
KDD	1994	991	1717	472	1995
ICDM	2001	355	765	275	2001
PKDD	1997	387	678	231	1999
PAKDD	1998	440	843	231	1999

Table 1: Conference statistics and PC information.

Name	Description
CFH	when the conference was first held.
NDA	the number of distinct authors that have published in that conference till 2005.
NOP	the number of papers published in the conference till 2005.
NDP	the number of distinct program committee members that have served in that conference.
YPC	the year from which we start to collect the program committee members of the conference.

Table 2: Column description.

Note that there is no program committee information (the `"serve for"` relationship) in the current version of Libra. So we extract such kind of information and add them into our dataset. Table 1 shows information related to list of conferences as extracted from the Libra system. Semantics of each column in Table 1 is shown in Table 2. Observe that these conferences are leading conferences in the database and data mining areas. Note that we fail to collect all the historical conference program committee members from the web as some of the web pages are not available any more.

As time-dependent relation extraction have already been described in [19] in the context of Libra system, we will focus our attention on the *feature extraction*, the *community model construction*, and the *post-processing* modules.

4.3 Feature Extraction

The objective of our framework is to construct the community models from academic data. Specifically, we focus on constructing conference program committee community models. As a result, the objective is to extract a group of *researchers* (authors in the Libra database) to form a community. In this section, we focus on the feature extraction for authors. Basically, there are two types of features for the authors: *snapshot-based features* and *delta-based features*.

4.3.1 Snapshot-based Features

As mentioned in the preceding sections, some of the features (such as distance between objects) can be directly extracted from the vector-based network using graph theory while other features may need propagation among different types of objects and relationships. Here, we first review the *PopRank* algorithm [19] that will be used to extract the propagation-based features. Then, the list of extracted snapshot-based features will be discussed.

The *PopRank* algorithm was proposed to rank web objects in the heterogeneous relation network. Basically, the popularities of web objects are propagated using different types of relationships, where different *propagation factors* are assigned automatically for different types of relationships. For example, to get the popularity of a paper, not only the collection of papers is considered, the relations with other objects such as conferences and authors are also taken into account.

To compute the popularity score of an object, the *PopRank* model takes into account both the popularity of the object and its relations with other objects. We use the following formula to compute the *PopRank* scores R_X of the objects of type X:

$$R_X = \varepsilon R_{EX} + (1-\varepsilon)\sum_{\forall Y} \Gamma_{YX} M_{YX}^T R_Y$$

where R_{EX} is the popularity of object X, which is the probability that the "random object finder" find this object using only relationship within this type of objects; while R_X is the probability that the "random object finder" find this object using all relationships with other types of objects. ε is the damping factor, Γ_{YX} is the propagation factor of the relationship from an object of type Y to an object of type X, and M_{YX}^T is the adjacent matrices. For details of the algorithm, please refer to [19].

Algorithm 1 Snapshot-based Feature Extraction

Input: The vector-based academic network: N_v, query-based feature set Q, PopRank-based feature set P
Output: A set of snapshot-based features F_s

Description
  Let A be a set of objects
  for all $a_i \in A$ **do**
    for all $q_j \in Q$ **do**
      for all valid timepoint t in the *time-window* **do**
        $s_q = N_v$.**Query**(q_j, *relation*, t, a_i)
        \ * * For example, N_v.**Query**(*NumPaper*, author-paper, 2000,100243) can be used to extract *NumPaper* for author *100243* for year 2000 * * \
        Insert (a_i, q_j, s_q, t) in the sequence F_s
      end for
    end for
    for all $p_j \in P$ **do**
      for all valid timepoint t in the *time-window* **do**
        $s_p = N_v$.**PopRank**(p_j, *relation*, t, a_i)
        \ * * For example, N_v.PopRank(*ConfRank*, author-paper, 2000, 123) can be used to extract *ConfRank* for conference *123* for year 2000 * * \
        Insert (a_i, p_j, s_p, t) in the sequence F_s
      end for
    end for
  end for
  Return F_s

Table 3 shows the list of sample snapshot-based features for individual authors, where the first five features can be extracted directly using queries against the database. These features are called *query-based features*. The last three features are extracted using *PopRank* and are called *PopRank-based features*. Note that these features are extracted from the vector-based network with a timestamp and time window. For example, given a time window of size *4* years, the timestamp of `1999`, the corresponding snapshot-based features are extracted using objects and relationships that exist between `1996` and `1999` as shown in Figure 3. For instance, if an author has published *20* papers between `1996` and `1999`, then the values for the *NumPaper* feature is set to *20*. Note that the values for certain features that are calculated using *PopRank*, such as *BSConf*, *AuthorRank*, and *ExpertRank*, are normalized. As a result, there will be a sequence of values for each author for each snapshot-based feature. The snapshot-based feature extraction algorithm is shown in Algorithm 1.

4.3.2 Delta-based Features

To reflect the evolution of the heterogeneous network, we propose to use the delta-based features. The intuition be-

Feature name	Description	Extraction Method	Class
NumPaper	Total number of papers the author has published.	Query	*Publishing*
AreaPaper	Number of papers the author has published in a specific area.	Query	*Publishing*
NumCoAuthor	Number of co-authors he/she has.	Query	*Social*
D2PCChair	The co-author distance between the author and the conference chair.	Query	*Social*
PCAge	Number of times the author has been a PC member.	Query	*Experience*
BSConf	The PopRank of the best conference he/she has served.	PopRank	*Combined*
AuthorRank	The PopRank of the author as a researcher.	PopRank	*Publishing*
ExpertRank	The PopRank of the author as an expert in a specific area.	PopRank	*Publishing*

Table 3: A list of sample features for authors extracted from the heterogeneous network.

hind is that, to be a conference program committee member, often the author should not only have been active in the area before but also be active at that time point. The snapshot-based features can reflect how active the author is at a particular time point, while the delta-based features are expected to reflect how active the author is during a certain time period. The delta-based feature extraction algorithm is shown in Algorithm 2.

Given two sets of most recent snapshot-based features of the same author at years t_i and t_{i+1}, denoted as $F_{t_i} = \{f^1_{t_i}, f^2_{t_i}, \cdots, f^k_{t_i}\}$ and $F_{t_{i+1}} = \{f^1_{t_{i+1}}, f^2_{t_{i+1}}, \cdots, f^k_{t_{i+1}}\}$, the delta-based features $\delta_{t_{i+1}} = \{\ \delta^1_{t_{i+1}},\ \delta^2_{t_{i+1}},\ \cdots,\ \delta^k_{t_{i+1}}\ \}$ is defined as follows.

$$\delta^j_{t_{i+1}} = \frac{f^j_{t_{i+1}} - f^j_{t_i}}{\max\{f^j_{t_{i+1}}, f^j_{t_i}\}} \quad (1)$$

Example 1: To extract the delta-based features for an author in `1999`, the two sets of most recent snapshot-based features at `1998` and `1999` are used as shown in Figure 3. With the corresponding values of *NumPaper*, the value of the feature *delta-based NumPaper* can be calculated accordingly. Similarly, the values for other delta-based features can be extracted. Finally, there will be a sequence of values for the delta-based features for each author.

By looking into the properties of the extracted features, the snapshot-based features and delta-based features can be categorized into three classes: `publishing`, `social`, and `experience`, as shown in Table 3. Here, `publishing` features are those features that can reflect the author's ability to publish papers such as *NumPaper* and *AreaPaper*. `Social` features refer to features that represent how active the author is in terms of research collaborations, while the `experience` features reflect the experiences of the authors in terms of organizing a conference or serving as a program committee member or chair. Note that here *BSConf* is taken as a combined feature of `publishing`, `social`, and `experience`.

4.4 Community Model Construction

In this section, we present a two-level community model in the context of conference program committee community. Basically, the model construction process is a learning process. That is, given a list of historical conferences and corresponding program committee members, the objective is to build a model to characterize the program committees in terms of the features of their members. As a result, given a conference and a specific timestamp, we can recommend a list of program committee members based on the constructed model. Note that in this model construction process, we use historical conference program committee members as positive examples to train the models. The reason we do not use negative examples is that it is often inaccurate to treat any author who is not selected as a program committee member as a negative example. This is because

Algorithm 2 Delta-based Feature Extraction

Input: The vector-based academic network: N_v, a set of snapshot-based features F_s
Output: A set of delta-based features F_δ

```
1: Description
2:   Let A be a set of objects
3:   for all a_i ∈ A do
4:     for all s_j ∈ F_s do
5:       δ_t^ij is calculated using Equation 1
     \* Refer to Example 1 *\
6:       Insert (a_i, s_j, δ_t^ij, t) in the sequence F_δ
7:     end for
8:   end for
9:   Return F_δ
```

the program committee community has certain community-wide constraints besides constraints to individual authors as mentioned in Section 3. The two-level community model consists of a *regression model* and a *multi-class classification model*. We elaborate on them in turn.

4.4.1 Regression Model

The goal of the regression model is to assign each author a *score* for a specific year to measure the quality of the author. Based on this score we can decides whether he/she can serve as a conference program committee member and which conferences he/she can serve. The intuition is that the score here represents the *PopRank* of the best conference the author is qualified to serve as a program committee member, which is denoted as the *BSConf* feature in Table 3. Note that the historical *BSConf* value can be extracted as the *PopRank* of the best conference he has served till then using queries. Then, we can use the historical *BSConf* values to predict the *BSConf* for the next year.

To get such score value for each author, we propose to build a regression model based on the historical instances of the *BSConf* values for the conference program committee members and corresponding features. As in different areas, the conference *PopRank* values may vary. In this paper, we build a general regression model for them by normalizing the values within areas. For instance, we assign rank values of *1* to the best database and the data mining conferences, respectively. The algorithm we used is the regression version of *SVM Light*[2].

The basic idea of the training process is to use the normalized *BSConf* feature as the label of each author. That is, the regression model is to assign each author a *BSConf* value based on all the other features denoted as F'_{t_1}, F'_{t_1-1}, and δ'_{t_1} for a specific time point t_1. An example training instance is $\{\ F'_{t_1-1},\ F'_{t_1},\ \delta'_{t_1},\ BSConf_{t_1}\ \}$, where the last value is taken as the *label*. Note that the regression model is time-dependent. If we want to build a regression model for

[2] http://svmlight.joachims.org/

1999, then all the conference program committee members before 1999 are used to construct the model.

Once the model is constructed, given any author with features F'_{t_1}, F'_{t_1+1}, and δ'_{t_1+1}, we can assign a score to him/her. The score is then compared with the conference *PopRank* values at the time t_{1+1} and necessary decisions can be made.

4.4.2 Multi-Class Classification

The regression model may generate more than two conferences that match with the author in terms of the assigned score and conference *PopRank* values. The multi-class classification model is then used to verify which conferences he/she may be able to serve as a program committee member, as different conferences have their own characteristics in choosing program committee members. In this paper, we use the *multi-class SVM light*[2].

Similar to the regression model, each program committee member in the historical conferences is taken as a training instance. The two sets of snapshot features and the delta features F'_{t_1}, F'_{t_1+1}, and δ'_{t_1+1} are used. However, the *label* is not the *BSConf* but the corresponding conference name. Note that the multi-class classification model is built from the list of conference whose *PopRank* values are very close. That is, models are constructed to distinguish these conferences where the program committee members may have very similar *BSConf* values.

For example, based on the *Libra* data, we build multi-class classification model for a list of database conferences such as SIGMOD, VLDB, and ICDE, where the *PopRank* are very close to each other. Similarly, another multi-class classification model is built for data mining conferences PKDD, PAKDD, and ICDM. By doing so, we can successfully distinguish conferences with very close *PopRank* values.

Once the classification model is constructed, given an author with all the required feature values and a list of candidate conferences (obtained using the regression model), we can decide which conferences the author is qualified to serve as a program committee member.

4.5 Applications and Post-Processing

In this section, we present two applications of the conference program committee models. The focus of this section is to illustrate the usefulness of the community model as well as necessary post-processing.

4.5.1 Conference PC Recommendation

This is a tool designed for experts who are expected to organize academic conferences. It provides the basic function of automatically recommending the list of program committee members and advanced functions for interactive refinement of the program committee.

Given the name of a conference, the corresponding area, the level of the conference, the program committee chair, and the number of expected members, the basic recommendation function works as follows. Firstly, the regression model is applied to the features of the researchers and only those whose output scores are within the specific range of the conference are selected. From the selected researchers, the multi-class classifier is used to select the set of researchers that best match the specific conference. Lastly, the global constraint-based pruning techniques are applied. Note that, usually, the number of researchers satisfying the above criteria is much more than the number of program committee members specified by the chair. Hence, we introduce the following two objectives and a constraint for further pruning (post-processing).

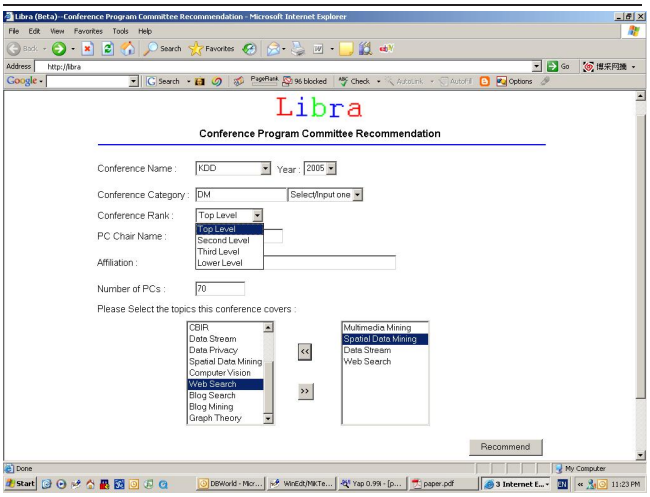

Figure 5: Screenshot of PC recommendation.

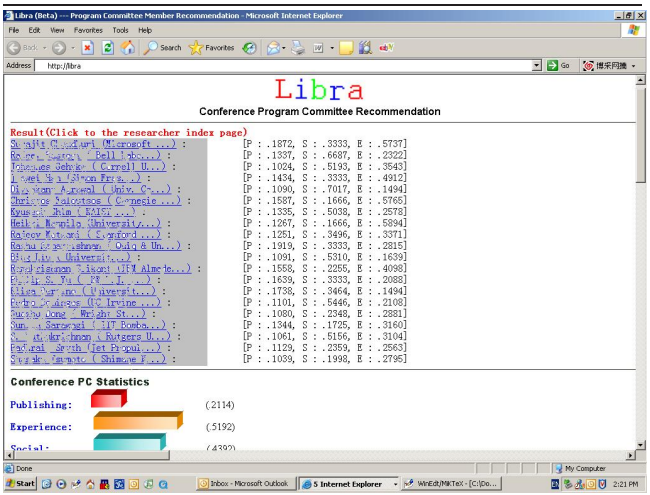

Figure 6: Screenshot of PC recommendation result.

- Given the author graph, where each vertex is an author, there is an edge between any two authors if they have been co-authors or they have co-served a conference/journal or they have the same affiliation. The *diversity* of a set of selected program committee members $A = \{a_1, a_2, \ldots, a_k\}$, denoted as $Div(A)$, is defined as: $Div(A) = \sum_{\forall a_i \in A, a_j \in A, a_i \neq a_j} MinDis(a_i, a_j)$, where $MinDis(ai, aj)$ is the minimum distance between any two selected program committee members. Then, the objective is to extract a fixed-size subset of the program committee member candidates that have the maximum diversity.
- Given the list of targeted topics for a conference, the second objective is to select a list of program committee members that cover all the topics. The *coverage* of the recommended committee members $A = \{a_1, a_2, \ldots, a_k\}$, denoted as $Cov(A)$, is defined as: $Cov(A) = \frac{\sum_{i=1}^{|topic|} MaxTopic_i(a_j)}{|topic|}$, where $1 \leq j \leq k$, $|Conf|$ denotes the total number of targeted topics for the con-

ference, $MaxTopic_i(a_j)$ is the maximal topic coverage of the selected candidates. Then, the objective is to extract a fixed-size subset of the program committee member candidates that have the maximum coverage.

- The constraint is the issue of incorporating *rising stars* as PC members. Every year there may be some rising stars in a particular area. These young researchers may not have much experience but have high publishing ability in quality conferences and journals. In our approach, we reserve some slots of the PC committee for such rising stars. In this paper, we use the following simple approach to identify rising stars. An author who has an average delta-based feature value larger than a user-defined threshold is considered to be rising star. Note that we acknowledge that rising star detection is a complex problem. Hence, the refinement of the above constraint is earmarked as future work.

Based on the above objectives and constraint, the selection process is realized using the multi-objective optimization genetic algorithm [7]. As we shall see in the experimental results, this approach can produce satisfactory recommendation results. Note that the program committee chair(s) can assign anyone he/she thinks is qualified but are not selected by the system. He/she can also remove any of the researchers from the recommended list. After that, the system will generate a list of program committee members that satisfies all the above constraints. Figures 5 and 6 depict the screenshots of the input and output of our recommendation system, respectively.

4.5.2 Researcher Evolution Tracking

This is a tool designed for the academic committee in research institutes and universities to evaluate the research performance of researchers and faculty members. For instance, it can be used as one of the "tools" to evaluate whether or not to promote a faculty member. The *researcher index* monitors the performance of a researcher in terms of his/her publishing ability, social activities, and experience of organizing research conferences. Moreover, the researcher's research interests and expertise areas can be tracked. Another important function of this tool is that the evolution patterns of the performances of well-known researchers in relevant areas can be used as examples to new and junior researchers in order to guide them to be successful researchers in the future.

5. EXPERIMENTAL RESULTS

We now present the experimental results to illustrate the performance of the proposed framework in the above mentioned applications in the context of academic network. We use the Libra dataset as our test bed. The version of Libra data used in this paper contains three major types of objects: `papers`, `authors`, and `conferences` or `journals`. Details of the dataset have been described in Section 4.2.

5.1 Program Committee Recommendation

To measure the quality of the conference program committee recommendation application, we use part of the data available as source to construct the community model and use rest of the data to evaluate the recommendation quality. Similar to the traditional classification quality measure, hereafter we use *precision* and *recall* as performance metrics:

Conference area	Training	Testing	Recall	Precision
Database	1994-2000	2001	0.891	0.913
	1994-2001	2002	0.908	0.916
	1994-2002	2003	0.914	0.922
	1994-2003	2004	0.920	0.923
	1994-2004	2005	0.921	0.925
Data Mining	1994-2000	2001	0.873	0.898
	1994-2001	2002	0.911	0.918
	1994-2002	2003	0.915	0.923
	1994-2003	2004	0.921	0.925
	1994-2004	2005	0.922	0.927

Table 4: Quality of PC recommendation result.

$$Precision = \frac{\text{avg. no. of correctly predicted PC members}}{\textit{no.} \text{ of unique PC members in the predicted list}}$$

$$Recall = \frac{\text{avg. no. of correctly predicted PC members}}{\textit{no.} \text{ of PC members}}$$

Note that the above quality measures are based on the average of different prediction results. The reason is that, given a conference and related constraints, by running our algorithms repeatedly, we may get a set of different prediction results. In real life scenarios, there may be more than one group of program committee members satisfying all the constraints as well.

Recommendation quality: Table 4 shows the precision and recall of the automatic recommendation results. The results are summarized based on the area of the conferences, which is shown in the first column of the table. For instance, the database area includes three conferences VLDB, SIGMOD, and ICDE. Also, the datasets that are used for training and testing are recorded in this table. In this set of experiments, firstly, previous program committee members are used to build community models. Then, the community model are tested with the immediate subsequent data. For example, in the first row of Table 4, we use all the PC members in the database area from `1994` to `2000` as training data to build the prediction model and predict the list of PC members for year `2001`. Note that the precision and recall are the average values for all the conferences in the specific area. From this table, it can be observed that the proposed community model can produce high quality results.

Effect of Distance: In Table 5, the *distance* between the training data and the testing data is varied from *1* year to *5* years. Here, the *distance* refers to the difference between the timestamps of the testing data and the latest training data. For example, all the experimental results that are shown in Table 4 have a distance of *1* year. If we take the data collection from `1994-1999` as training data and use the constructed model to recommend program committee members for years `2001` and `2002`, then the distances are *2* and *3* years, respectively. Note that for a specific distance value, the precision and recall in this table are computed by taking the mean of the precision and recall values of all the seven conferences listed in Table 1 for the specified distance. It is obvious that when the distance between the training data and testing data increases, the quality of the recommendation may decrease slightly. This observation shows that the conference program communities are evolving over time. That is, if the distance between the training and testing data is large enough, the community model cannot accurately reflect the current characteristics of the community. As a result, the quality of recommendation quality decreases.

Recommendation quality of a new conference: To

Distance	Recall	Precision
1	0.909	0.918
2	0.901	0.907
3	0.897	0.903
4	0.878	0.894
5	0.862	0.885

Table 5: Effects of distance.

Model No.	Description	Precision	Recall
1	All Conferences	0.683	0.579
2	DM Conferences	0.895	0.863
3	KDD conferences	0.816	0.794
4	Other DM conferences	0.912	0.904

Table 6: Recommendation with general models.

evaluate the recommendation quality of new conferences, general models are used for specific conferences. For instance, Table 6 shows the quality of recommendation using four different models to generate the list of program committee members for a specific data mining conference "PAKDD" for 2005. Note that the PAKDD conference program committee information is not used in this process, as we are assuming PAKDD as a new conference. The four models are constructed as follows. (a) **Model 1:** All conferences listed in Table 1 except PAKDD; (b) **Model 2:** All data mining conferences in Table 1 except PAKDD; (c) **Model 3:** KDD conference; (d) **Model 4:** Data mining conferences except KDD and PAKDD.

It can be observed that the last three models can produce satisfactory recommendation results, while the general model built from all the conference (except PAKDD) can only provide recommendation with limited quality.

5.2 Researcher Evolution Tracking

We now present different types of evolution patterns of several researchers over time. Here, for each researcher the *BSConf* feature value is used as an overall measure that combines `publishing`, `social`, and `experience` features. Note that due to privacy issue we have not revealed the names of the researchers. Figure 7 shows different types of evolution patterns. For instance, researcher *1* becomes very active from 1996 and his/her performance increases till 2001. After that he/she has maintained a stable performance. Researcher *2*, on the other hand, has consistently maintained a stable performance since 1994. Performance of researcher *3* is getting better every year. On contrary, the performance of researcher *4* has gone down since 2001. To identify the "rising star", we simply use the percentage of changes. If it is larger than a threshold, then the researcher is a rising star. For instance, *researcher 1* is identified as a rising star in 1996 as his *BSConf* value increase dramatically from 1994 to 1996. Note that the *BSConf* values (Figure 7) have been normalized to values between 0 and 10.

6. CONCLUSIONS

In this paper, we proposed a novel framework of web community mining that combines the evolutionary and heterogeneous properties of web data along with their community-wide constraints. We illustrated the usefulness of our framework with a real world example based on the academic network. In our approach, we proposed a novel structure called vector-based heterogeneous network to model the heterogeneity and evolutionary features of web objects and associated relationships. Then, a set of features of a particular community is extracted from this network using the *PopRank* algorithm [19]. After that, we proposed a two-level community model. Experiments with real academic network data reveal that the proposed framework can produce high quality results and interesting insights about the evolution pattern of the network.

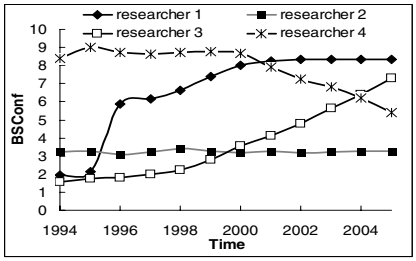

Figure 7: Researcher evolution tracking.

7. REFERENCES

[1] L. Adamic and E. Adar. Friends and neighbors on the web. In *TR HP*, 2001.

[2] S. Asur, S. Parthasarathy, and D. Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *SIGKDD*, 2007.

[3] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *SIGKDD*, 2006.

[4] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[5] T. Falkowski, J. Bartelheimer, and M. Spiliopoulou. Mining and visualizing the evolution of subgroups in social networks. In *IEEE/WIC/ACM WI*, 2006.

[6] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *SIGKDD*, 2000.

[7] C. M. Fonseca and P. J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part I: A unified formulation. In *IEEE TSMC*, 28(1):26–37, 1998.

[8] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *ACM HT*, 1998.

[9] J. Hopcroft, O. Khan, B. Kulis, and B. Selman. Natural communities in large linked networks. In *SIGKDD*, 2003.

[10] H. Ino, M. Kudo, and A. Nakamura. Partitioning of web graphs by community topology. In *WWW*, 2005.

[11] J. M. Kleinberg. Hubs, authorities, and communities. In *ACM CSUR*, 31(4es):5, 1999.

[12] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursy evolution of blogspace. In *ACM WWW*, 2003.

[13] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for emerging cyber-communities. *Computer Networks*, 31(11–16):1481–1493, 1999.

[14] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *SIGKDD*, 2006.

[15] J. Leskovec, J. Kleinberg, and C. Faloutos. Graphs over time: Desification laws, shrinking diameters and possible explanations. In *SIGKDD*, 2005.

[16] J. Leskovec, K.J. Lang, A. Dasgupta, and M.W. Mahoney. Statistical properties of community structure in large social and information networks. In *ACM WWW*, 2008.

[17] Y. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. L. Tseng. Blog community discovery and evolution based on mutual awareness expansion. In *ACM/IEEE WI*, 2007.

[18] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 1995.

[19] Z. Nie, Y. Zhang, J.-R. Wen, and W.-Y. Ma. Object-level ranking: bringing order to web objects. In *WWW*, 2005.

[20] F. Osareh. Bibliometrics, citation analysis and co-citation analysis: A review of literature I. Libri 46 (September 1996), 149-158, 1996.

[21] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe. A framework for community identification in dynamic social networks. In *SIGKDD*, 2007.

[22] M. Toyoda and M. Kitsuregawa. Extract evolution of web communities from a series of web archives. In *ACM HT*, 2003.

[23] W.-J. Zhou, J.-R. Wen, W.-Y. Ma, and H.-J. Zhang. A concentric model for community mining in graph structures. TR, Microsoft Research, 2002.