Deduced Social Networks for Educational Portal

Monika Akbar
Dept. of Computer Science
Virginia Tech, Blacksburg, VA
amonika@vt.edu

Clifford A. Shaffer
Dept. of Computer Science
Virginia Tech, Blacksburg, VA
shaffer@vt.edu

Edward A. Fox
Dept. of Computer Science
Virginia Tech, Blacksburg, VA
fox@vt.edu

ABSTRACT

By analyzing the behavior of previous users, educational portals can be made to provide new users with more support to find the best information. The AlgoViz Portal collects metadata on algorithm visualizations and associated research literature. We show how logs can be used to discover latent relationships between users, deducing an implicit social network. By clustering the log data, we find different page-viewing patterns, which provide practical information about the different groups of users.

1. INTRODUCTION

Educational digital libraries (DLs) or portals provide a gateway to educational resources. An abundance of resources provides opportunities but also creates a problem for users when searching for high quality material. While users of educational portals can play a critical role by providing their feedback and ratings on the content, not many choose to do so. Lack of active participation is a key problem for building online community [3]. Even when DLs provide community space, willingness to participate can decide the success of a portal.

One example of a project that combines an educational portal with online community is the AlgoViz Portal (`http://algoviz.org`). In the absence of adequate explicit user feedback, AlgoViz usage data has helped us to generate networks and find common usage patterns. A portal built for a specific user community (in our case, the educators) may support disparate groups of users. An example of such groups results from linking users via shared resources (e.g., a co-author network). Analysis of these networks and their contextual information can reveal interesting user behavior, different user roles, and communities with similar interests. In this paper, we present a methodology for using log data to deduce connections between users and identify user interests. Applications of such information include refining existing services and providing recommendations on content. While most of the current recommendation systems rely on active user data (e.g., feedback, review, ratings, buying history), we depend solely on passive user data (e.g., clicks, pageviews, times in pages, etc.). Unfortunately, our target audience is mostly anonymous users. Yet, we are able to identify what these users are interested in. We believe our approach can be used in other educational portals that have little active user participation but abundant passive user activities.

The rest of the paper is organized as follows: Section 2 provides background and related work. We describe the concept of Deduced social network and methodology of finding and analyzing these networks in Section 3 along with a case study in Section 4. Section 5 presents two prototype applications developed based on the results of these analyses. We conclude the paper in Section 6.

2. RELATED WORK

Many existing educational portals are built upon a DL framework. Social aspects of DLs were described in [2]. Early DL research pointed out the importance of understanding the needs of the target audience and of building online communities [2, 11]. Researchers have documented the various types of participation, and discovered factors that motivate users to actively participate in those communities [15]. Social navigation methods are used to guide users in an unfamiliar information space, but these methods largely depend on previous user feedback/rating [14]. In cases where user feedback is scarce, rating-based systems can prove insufficient to derive useful usage information. While Amazon has a successful recommendation system [12], it is targeted for e-commerce and depends heavily on user feedback. For educational sites, domain-based recommendation systems for e-learning were explored in [4]. In cases where user activity is less, recommendation systems based on social patterns were proposed in [6].

3. DETECTING COMMUNITIES AND INTERESTS IN AN EDUCATIONAL PORTAL

In this paper, we use the term 'group' to refer to some users with a special interest towards specific content. Figure 1 shows the architecture of the system used in this paper. This figure portrays the approach taken in the case study section and consists of four segments: Filtering Module, Network Generation, Finding Groups, and Topic Modeling for all User Groups. We will briefly describe each segment next. Further details can be found in Section 4 where we present a case study.

Filtering Log Data: Many online systems log their user activity for various purposes. These data, when coupled with user account information, provide useful insight on various factors. Unfortunately, logs also describe activity by crawlers, spammers, and bots along with that of legitimate users. Thus the first step for further analysis is to filter the logs to remove anomalous data.

Generating Networks within an Educational Portal: Once the log has been filtered, we cluster users based

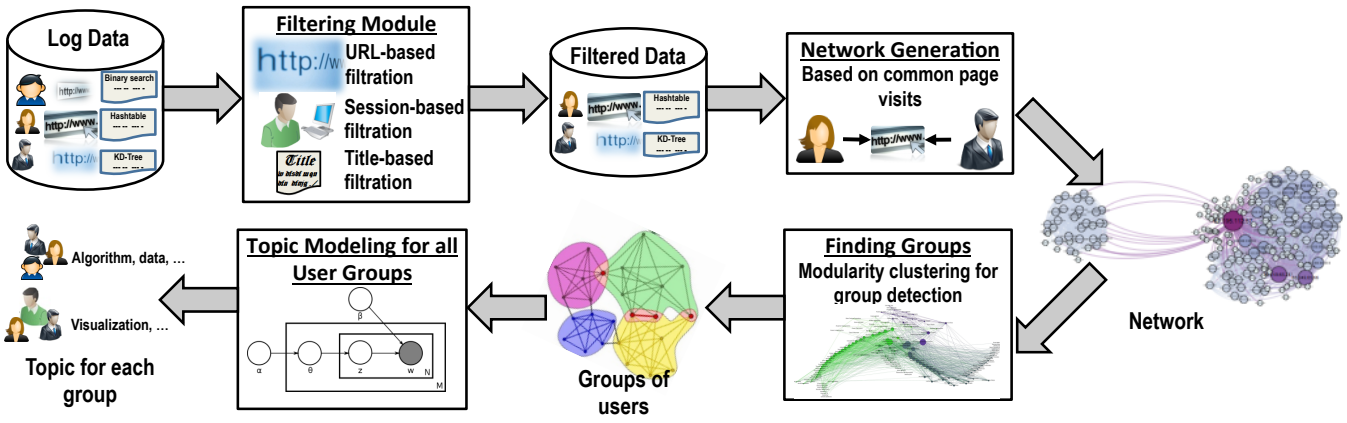


Figure 1: Architecture for group detection within an educational portal using passive user data.

on their passive activities within the portal. A network built using passive data such as pageviews can be called a passive social network. In contrast to active social networks, passive social networks are not created explicitly by the users but are generated based on user activities. Passive social networks inherit more object-centric approach rather than the relationship-centric approach of active social networks. A wide range of objects can be used in a portal to define a passive social network. For example, we can define a Deduced Social Network (DSN) where users will be the nodes, and an edge links two users who viewed the same page. Our case study in Section 4 contains details on constructing such DSNs. Formally, we define a DSN as:

Definition 1. A **Deduced Social Network (DSN) is a Graph with tuple** G=($Entity$, $Connection$, $Object$, k), **where:**

- $Entity$ **is a node of the network, and** $Connection$ **is an edge between two** $Entities$**,**
- $Object$ **is an attribute of** $Entity$**, where one** $Entity$ **can have multiple** $Object(s)$**; and**
- k **is a function that returns the minimum number of** $Object(s)$ **that must be common between two** $Entities$ **to create a** $Connection$ **between them.**

Finding Groups within a DSN: Based on DSN characteristics, we may need further analysis to identify groups of users. Depending on connection factors, the DSN can be dense or sparse, thus having different network characteristics (e.g., degree distribution, betweenness centrality).

Identifying Group Interest: Once groups of users are identified, the next step is to find their areas of interest. Characterization of each group can be done by exploring the contents of the pages visited by the members of the group. Instead of such mundane exploration of pages for each group, it is also possible to characterize groups using cluster enrichment techniques (e.g., hypergeometric distribution [9]), classification techniques (e.g., Naive Bayes text classification [13] and Support Vector Machine-based classification), or topic modeling (e.g., Latent Dirichlet Allocation (LDA) [1]).

4. CASE STUDY: THE ALGOVIZ PORTAL

The AlgoViz portal collects within its database several aspects of user history. The Accesslog table is given in Table 1, showing data on the session, user IDs, IPs, timestamps for when the page was visited, etc. AlgoViz content is open for public viewing, hence it is possible for users to not register and so not receive a user ID. These users are referred to as Anonymous users and have a default user ID of 0. Despite a small number of registered users, AlgoViz project leaders are interested in understanding the trends of its overall user base. So instead of IDs, we rely on other methods to identify users whether registered or unregistered.

We used IP addresses to denote users. For any given IP, we are able to view which pages were viewed in that session and the time when the page loaded. We used these data to deduce a behavioral social network. The Accesslog table uses access-id (AID) as the primary key. It also stores session information. Each page viewed in a session generates a new AID row in the table.

We selected the log data of two months (September and October from 2010) for processing. An average month generates 100,000 rows in the table. Much of the data are generated by spammers, crawlers, bots, etc. We followed a three-step process to filter the log data of such outliers. The process involves filtering data based on page title, internal path of the page, and session information.

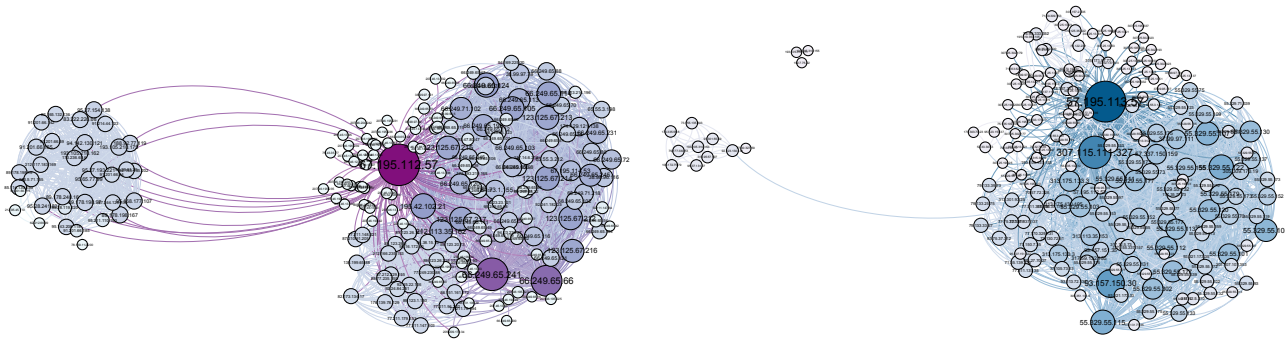
4.1 Deduced Social Networks

The filtered data were then used to connect pairs of users based on their common pageviews. These connections created a DSN. Nodes represent a user and edges indicate that the users have viewed similar pages. We used the *connection threshold* parameter to vary the network strength. A *connection threshold* of size k for an edge indicates that two users have viewed at least k common pages. Figure 2 shows DSNs based on AlgoViz log data for the months of September and October 2010 with a connection threshold of 10. Two users were connected only if they viewed at least 10 similar pages within a month. We varied k from 1 to 20. With lower k, the network starts to get dense and it becomes difficult to effectively identify interesting user groups. At k=8, we start observing special segments within the network which remain similar until k reaches 15. Then the network starts to lose some smaller segments. So, we selected k=10 for our analysis. Also, from a user's point of view, it is possible

Table 1: Sample entries of AlgoViz log data.

Session ID	Page Title	Internal Path/Page URL	IP Address‡	User ID	Timestamp
ievav83	Lifting the hood of the computer...	node/1413	9x.1y8.111.25	0	1276272047
t5fuuba	biblio/export/tagged/118/popup	research.cs.vt.edu/algoviz/biblio	2xy.2z.2a8.192	5	1276260935
ivuks8s	Has an AV helped you learn a topic in computer science?	research.cs.vt.edu/algoviz/poll/	1x1.yz.145.90	0	1276260943

‡ IPs are masked to protect user identity.



(i) Deduced social network for AlgoViz (September 2010) (ii) Deduced social network for AlgoViz (October 2010)

Figure 2: Using log to find user groups with similar interest (connection threshold k=10). Node size is proportional to the degree of each node. IPs are masked to protect user identity.

and likely that s/he may view ten or more pages in a month (or within a session). Once we selected k, we used a force-directed graph layout algorithm [5] to draw these networks. The network shown in Figure 2 (i) has 195 nodes and 2255 edges, and (ii) has 130 nodes with 1180 edges.

4.2 Network Analysis

Network analysis can help us to understand hidden trends and user preferences. Depending on the network characteristics, a number of approaches can be used to analyze the network. For example, for dense graphs, graph partitioning can help us to find smaller sub-graphs that might reveal interesting information. We describe below some approaches that might be useful for analyzing the passive networks discussed in the previous section.

Graph partitioning — Detecting Groups: There are a number of ways in which graphs can be analyzed, graph partitioning being one. Graph partitioning breaks the graph into disjoint subsets such that the number of connections within the subsets is high but the number of connections between the subsets is low. Relevant graph partitioning techniques have been studied in areas such as web science [10], epidemiology[8], sensor networks [16], etc.

Modularity, introduced by Girvan and Newman [7], is a quality measure for clustering that has been successfully adopted in many areas. Modularity clustering is dependent on edge betweenness — a measure that assigns weights on an edge as the number of shortest paths between pairs of vertices containing this edge. If a network contains multiple communities then the number of edges connecting the communities will be less than the number of edges within the community, and all shortest paths between those communities will contain one of those edges that connect the communities. Thus, the edges that connect the communities will have relatively higher edge betweenness values.

As we see from Figure 2, the graphs are very dense, making it difficult to analyze user trends. Graph-partitioning methods are used to find sub-graphs or groups within such dense graphs. We used Modularity clustering which has been successfully used to find communities within large networks in other domains. The result of the clustering is given in Figure 3. While most of the clusters of September 2010 DSN (Figure 3 (left)) are closely situated, clusters 2 and 4 are clearly separable from the rest of the clusters. Similarly, in October 2010 DSN (Figure 3 (right)), clusters 4 and 6 are distinguishable from the rest of the clusters. This indicates that while some clusters are closely related, others address different topics.

Topic modeling — Identifying Group Interests: Finding clusters is useful for detecting groups and their sizes

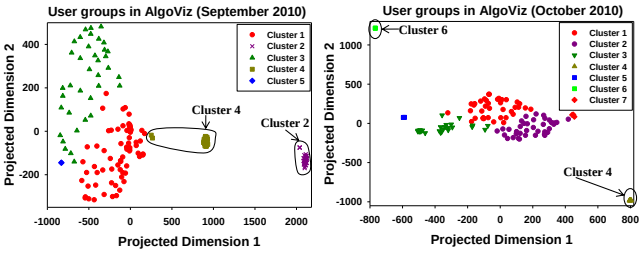


Figure 3: Clusters found in the DSNs of Figure 2.

Table 2: Topic distribution of Sept-10 DSN, T=5.

Clust. #	Top Topic (1)	Contr - ibution	Top Topic (2)	Contr - ibution	Top Topic (3)	Contr - ibution
1	3	0.667	5	0.25		
2	3	0.312	1	0.212	5	0.209
3	1	0.539	2	0.193	4	0.119
4	3	0.75	5	0.156		
5	2	0.254	1	0.216	4	0.213

Table 3: Topic distribution of Oct-10 DSN, T=7.

Clust. #	Top Topic (1)	Contr - ibution	Top Topic (2)	Contr - ibution	Top Topic (3)	Contr - ibution
1	3	0.987				
2	1	0.828	2	0.138		
3	6	0.421	5	0.145	3	0.12
4	1	0.815				
5	5	0.253	4	0.217	6	0.137
6	3	0.264	6	0.161	4	0.144
7	6	0.306	3	0.221	7	0.119

but it is not sufficient to understand patterns within the groups. Topic modeling allows us to get an overview of the subjects addressed in the clusters. Probabilistic models such as Latent Dirichlet Allocation (LDA) [1] have been used extensively to detect topics for a document corpus.

The clustering results in the previous subsection were used to identify topics within the clusters. We used the Mallet API[1] which utilizes the LDA approach to find topics appearing in the page titles of that cluster. While building models, we set the number of topics similar to the number of clusters that were generated for each month. For example, in September 2010, there were five clusters and we opted for five topics for those clusters. The number of sampling iterations for the topic model of each month was 200.

Two of the top-most topics in the September 2010 DSN include words related to AlgoViz bibliography entries (i.e., biblio). One of the prominent collections of AlgoViz is the bibliography of publications related to algorithm visualizations. The collection can be sorted by author, title, publication type, or publication year. Also, these entries can be exported in RTF, bibtex, and XML format. The last three topics are less related to bibliography entries. Two are related to sorting, tree, programming, etc.

Table 2 shows the topic distribution within the clusters in the September 2010 DSN. The first column shows the cluster ID, while the subsequent pairs of columns show the topic ID and its proportion in the cluster. We show the top three topics for each cluster that has at least 10% topic proportion value within the cluster. For example, in cluster 1, topics 3 and 5 are most dominant. Cluster 4 has a similar topic distribution. Cluster 5 has three dominant topics (i.e., topics 2, 1, and 4).

The topic distribution for the October 2010 DSN is shown in Table 3. Cluster 1 consists of biblio entries (e.g., biblio, export, rtf), and other content pages (e.g., linked, functional). Topics 3, 6, and 7 are mostly related to bibliography entries and these topics are dominant in cluster 7. Topics 1, 2, 4, and 5 are mostly related to AlgoViz catalog entries, These topics are prominent in clusters 2, 4 and 5.

5. APPLICATIONS

The clustering results along with the topics highlighted in the previous section indicate that AlgoViz users have clusters of interests when it comes to using online resources related to algorithm visualizations. There are groups of people who are solely interested in bibliography entries. Other users are more inclined to catalog entries; of these, interests clusters around sorting or graph algorithms, animations, and demos. Along with their interests, we are able to detect the size of each cluster. Knowing the groups, their interests, and size gives us leverage on better serving the target audience.

In AlgoViz we used the results in two ways: within the content recommendation blocks that suggests a list of entries and within the ranking function that lists entries for users during browse and search operations. Details on both approaches are given next.

5.1 Approach 1: Recommend Content

AlgoViz has a number of different collections: forum posts, bibliography lists, and catalog entries. Each collection has a different content type. For each collection we developed a separate recommendation block that shows a list of highly accessed entries of that collection (see Figure 4). Following the cluster results found in Section 4.2, we mapped the collection in AlgoViz with the most similar cluster(s) and selected top entries of those cluster(s) for showing in the blocks. For example, entries from cluster 1 in the September 2010 DSN are more related to catalog entries (i.e., topics 3 and 5). Thus top entries from cluster 1 were used to generate the content of the recommendation block (titled "People also viewed") in Figure 4 (top). Similarly, cluster 2 demonstrated a high volume of forum posts, hence it was used for the forum post recommendation block in Figure 4 (bottom).

5.2 Approach 2: Refine Search and Browse for Catalog Entries

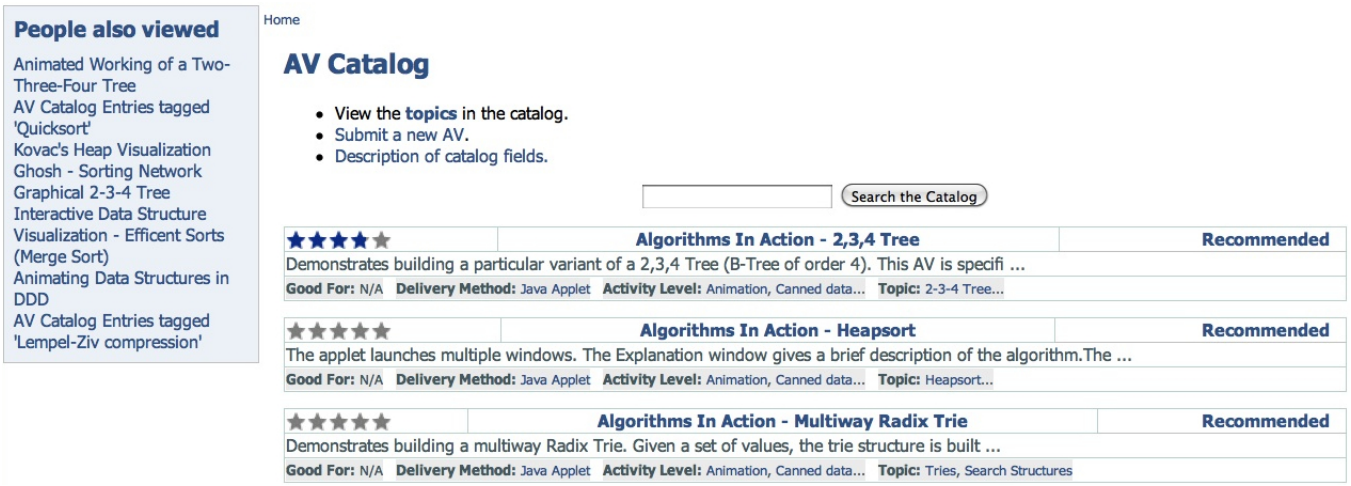
For the second approach we modified the ranking function used to list the entries in AlgoViz (for both browsing and searching). AlgoViz is built on the Drupal[2] infrastructure. While Drupal has a native ranking function, we opted to use Apache Solr functionalities to index and rank AlgoViz content. Though the Apache Solr module in Drupal has a robust weighting mechanism, by default it does not provide enough flexibility to customize the ranking result. We created a custom module that takes additional factors into account and updates the score for each catalog entry in AlgoViz. One of the custom factors in this function is the 'cluster-view' point.

For any given time span, if an entry in AlgoViz received a certain amount of views within a cluster whose topics were highly related to catalog entries (e.g., cluster 1 in September 2010 DSN), that catalog entry was given additional points as 'cluster-view' point. We used the following function to assign points for AlgoViz specific fields to rank catalog entries (CE):

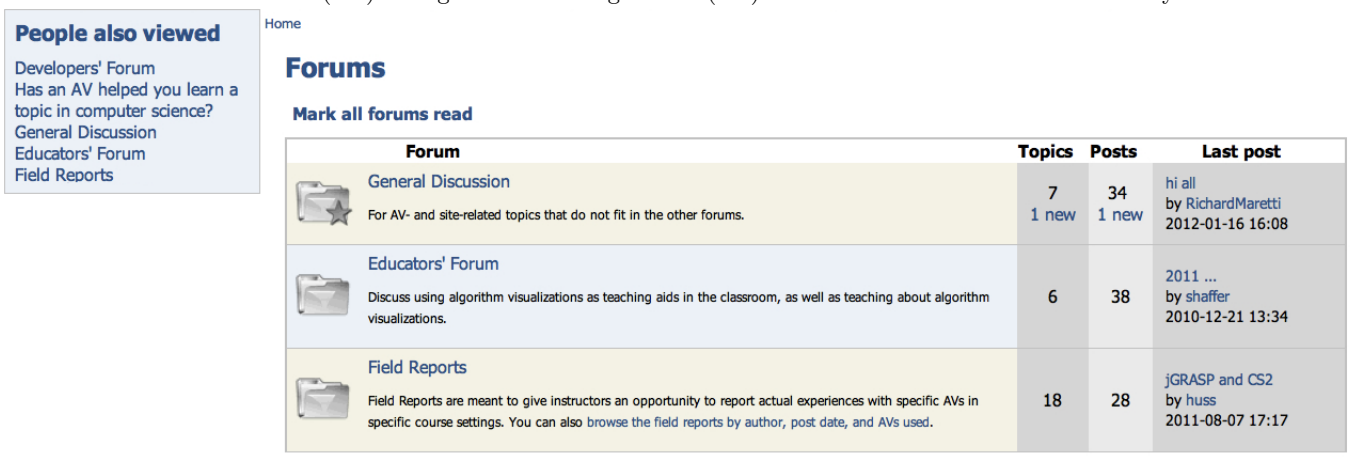
$score(CE_i) = x + y + ... + z$

[1] `http://mallet.cs.umass.edu/`

[2] `http://drupal.org`



The recommendation block (left) in AlgoViz for catalog entries (CE) based on cluster results dominated by CE contents.



The recommendation block (left) for forum posts (FP) in AlgoViz based on cluster results dominated by FP contents.

Figure 4: AlgoViz Content Recommendation Blocks based on Log Data.

where, $x = \begin{cases} 20 & \text{if the CE has 'Yes' in the 'Works' field} \\ 0 & \text{otherwise} \end{cases}$,

$y = \begin{cases} 6 & \text{if the CE is 'Recommended'} \\ 0 & \text{otherwise} \end{cases}$, and

$z = \begin{cases} 5 & \text{if the CE was present at least } m \text{ times, in a} \\ & \text{cluster dominated by Catalog Entry content type.} \\ 0 & \text{otherwise} \end{cases}$

6. CONCLUSIONS AND FUTURE WORK

The focus of this paper was to define DSN, present exploratory analysis, and describe developed prototypes. In the future, we plan to evaluate the prototypes and their variations. One variation on the prototype applications is to customize the search and browsing results for each anonymous user based on their pageviews. We also plan to use the page content, instead of the page titles, to model the topics within each cluster and compare it with the existing result. Creating navigation networks of pages based on common viewers is another area we plan to pursue.

7. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[2] C. L. Borgman. Social Aspects of Digital Libraries (working session). In *Proceedings of the First ACM Internationl Conference on Digital Libraries*, page 170, 1996.

[3] B. S. Butler. Membership Size, Communication Activity, and Sustainability: A Resource-Based Model of Online Social Structures. *Information Systems Research*, 12(4):346–362, 2001.

[4] A. Dong and B. Wang. Domain-Based Recommendation and Retrieval of Relevant Materials in E-learning. In *IWSCA '08*, pages 103–108, 2008.

[5] P. Eades and M. L. Huang. Navigating Clustered Graphs using Force-Directed Methods. *Journal of Graph Algorithms and Applications*, 4:157–181, 2000.

[6] J. Freyne, R. Farzan, and M. Coyle. Toward the Exploitation of Social Access Patterns for Recommendation. In *RecSys*, pages 179–182, 2007.

[7] M. Girvan and M. E. J. Newman. Community Structure in Social and Biological Networks. *National*

Academy of Science, 99(12):7821–7826, 2002.

[8] J. Hadidjojo and S. A. Cheong. Equal Graph Partitioning on Estimated Infection Network as an Effective Epidemic Mitigation Measure. *PLoS ONE*, 6(7):e22124, 2011.

[9] A. Hald. The Compound Hypergeometric Distribution and a System of Single Sampling Inspection Plans Based on Prior Distributions and Costs. *Technometrics*, 2(3):275–340, 1960.

[10] H. Ino, M. Kudo, and A. Nakamura. Partitioning of Web Graphs by Community Topology. In *WWW*, pages 661–669, 2005.

[11] K. Lightle, E. Almasy, L. Barbato, S. Clark, Y. George, S. Hsi, C. Lowe, P. Mackinney, and E. McIlvain. Draft Report: Metrics Recommendations and Resources for NSDL Projects. `https://www.nsdlnetwork.org/sites/default/files/Draft%20Report-Metrics_Recommendations.pdf`, (last accessed on April 2012), 2009.

[12] G. Linden, B. Smith, and J. York. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *Internet Computing*, 7(1):76 – 80, 2003.

[13] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[14] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted Collaborative Filtering for Improved Recommendations. In *National Conference on Artificial Intelligence*, pages 187–192, 2002.

[15] O. Nov, M. Naaman, and C. Ye. Analysis of Participation in an Online Photo-sharing Community: A Multidimensional Perspective. *Journal of the American Society for Information Science and Technology*, 61(3):555–566, 2010.

[16] S. Roy, Y. Wan, and A. Saberi. A Flexible Algorithm for Sensor Network Partitioning and Self-partitioning Problems. In *Algorithmic Aspects of Wireless Sensor Networks*, volume 4240, pages 152–163. 2006.